

STUDENT EDITION

TCP/IP

Karanjit S. Siyan



- > Environnements multi-plates-formes
- > Couches, adressage, routage, sous-réseaux et surréseaux
- > IP, ICMP, BOOTP, UDP, SMTP
- > RFC et documentations
- > IPv6 et ATM



CampusPress

www.pearsoneducation.fr



TIRAGE LIMITÉ

20€



Digitized by the Internet Archive
in 2021 with funding from
Kahle/Austin Foundation

<https://archive.org/details/tcpip0000siya>

TCP/IP



CampusPress

CampusPress a apporté le plus grand soin à la réalisation de ce livre afin de vous fournir une information complète et fiable. Cependant, CampusPress n'assume de responsabilités, ni pour son utilisation, ni pour les contrefaçons de brevets ou atteintes aux droits de tierces personnes qui pourraient résulter de cette utilisation.

Les exemples ou les programmes présents dans cet ouvrage sont fournis pour illustrer les descriptions théoriques. Ils ne sont en aucun cas destinés à une utilisation commerciale ou professionnelle.

CampusPress ne pourra en aucun cas être tenu pour responsable des préjudices ou dommages de quelque nature que ce soit pouvant résulter de l'utilisation de ces exemples ou programmes.

Tous les noms de produits ou marques cités dans ce livre sont des marques déposées par leurs propriétaires respectifs.

47 bis, rue des Vinaigriers
75010 PARIS
Tél. : 01 72 74 90 00

Mise en pages : TyPAO

ISBN : 2-7440-1653-5
Copyright © 2003 CampusPress

Tous droits réservés

CampusPress est une marque de
Pearson Education France

Titre original : *Inside TCP/IP Third edition*
Traduit de l'américain par Pierre Naniche et Yann
Schwartz
Mis à jour par : Jean-Marie Munier

ISBN original : 1-56205-714-6
Copyright © 1997 by New Riders Publishing
Tous droits réservés

New Riders Publishing est une marque de Macmillan
Computer Publishing
201 West 103rd Street
Indianapolis, Indiana 46290, USA

Toute reproduction, même partielle, par quelque procédé que ce soit, est interdite sans autorisation préalable. Une copie par xérogaphie, photographie, film, support magnétique ou autre, constitue une contrefaçon passible des peines prévues par la loi, du 11 mars 1957 et du 3 juillet 1995, sur la protection des droits d'auteur.

TCP/IP

Karanjit S. Siyan



CampusPress

Introduction

I

1 LES COUCHES DE TCP/IP ET L'INFRASTRUCTURE PHYSIQUE

1	<i>Introduction à TCP/IP</i>	7
2	<i>Les couches de TCP/IP</i>	33
3	<i>Support réseau pour TCP/IP</i>	69

2 INFRASTRUCTURE DES RÉSEAUX TCP/IP INTERCONNECTÉS

4	<i>L'adressage IP</i>	153
5	<i>Protocoles de résolution d'adresse</i>	193
6	<i>Internet Protocol</i>	241
7	<i>Le routage IP</i>	295
8	<i>Le protocole ICMP</i>	325
9	<i>Sous-réseaux et surréseaux</i>	357
10	<i>Protocoles de routage IP</i>	403
11	<i>Protocoles de transport</i>	443

3 LES SERVICES TCP/IP

12	<i>Configuration automatique</i>	511
13	<i>Services d'application</i>	545
14	<i>Gestion de réseau TCP/IP</i>	623

4 L'AVENIR

15 *IPv6 et ATM*

643

5 ANNEXES

A *Protocoles standards*

673

Index

689



Introduction	I
Organisation du livre	2
Partie I : Le modèle TCP/IP et l'infrastructure physique	2
Partie II : Infrastructure des réseaux TCP/IP interconnectés	2
Partie III : Les services TCP/IP	3
Partie IV : L'avenir	3
Annexe	3
Conventions typographiques	4
Paragraphes spéciaux utilisés dans ce livre	4

LES COUCHES DE TCP/IP ET L'INFRASTRUCTURE PHYSIQUE

1 Introduction à TCP/IP	7
Vue d'ensemble des réseaux TCP/IP	8
Qu'est-ce que TCP/IP ?	8
TCP/IP hier et aujourd'hui	10
Rôle de la DARPA	10
Les premières expérimentations de la DARPA	10
L'évolution d'ARPAnet	12
Passage des réseaux propriétaires aux réseaux TCP/IP	14
Réseau multiconstructeur	15
Evolution de l'impact de TCP/IP	17
Montée en puissance de TCP/IP	17
Evolution de TCP/IP	18
Vue d'ensemble des applications TCP/IP	19
L'Internet	23
Exemples de réseaux sur l'Internet	23
L'IAB et l'Internet Society	25
RFC, STD et IEN	27
Obtention des RFC	30
Récapitulatif	31
Testez vos connaissances	31
2 Les couches de TCP/IP	33
Principes des couches de protocoles	34
Le modèle OSI	35
Les couches du modèle OSI	36
Echange transversal des données dans le modèle OSI	46

Le modèle DoD	48
La couche accès réseau	49
La couche Internet	49
La couche transport hôte à hôte	49
La couche application	50
Transfert des données sur un réseau TCP/IP d'après le modèle DoD	50
Comparaison des modèles OSI et DoD	51
Hierarchie de l'implémentation de TCP/IP	52
Multiplexage et démultiplexage de protocoles	53
Implémentation de TCP/IP et système d'exploitation	66
Récapitulatif	67
Testez vos connaissances	67
3 Support réseau pour TCP/IP	69
Vue d'ensemble de la partie matérielle concernée par TCP/IP	69
TCP/IP sur des réseaux locaux IEEE	71
Couche <i>Media Access Control</i> (MAC)	72
Couche <i>Logical Link Control</i> (LLC)	73
Règles de câblage des réseaux TCP/IP	77
Disposition du câblage	77
Choix du support physique	79
Fonctionnement des réseaux Ethernet	81
Méthode d'accès sur les réseaux Ethernet	81
Options de câblage Ethernet	83
Règles de câblage du câble standard Ethernet	83
Règles de câblage du câble fin Ethernet	88
Règles de câblage des câbles 10BASE-T	90
Câblage en paires torsadées non blindées (UTP)	94
Combinaison de supports dans les réseaux Ethernet	96
Combinaison de câbles fin et épais dans un segment	100
Liaison de réseaux avec les standards 10BASE-FB et 10BASE-FL ..	101
Comprendre les trames dans les réseaux TCP/IP	102
La trame Ethernet-II	102
La trame IEEE 802.3	103
Différences entre Ethernet-II et IEEE 802.3	104
L'en-tête SNAP	104
Fast Ethernet et TCP/IP	106
La politique du numérotage : IEEE 802.13, IEEE 802.14,	
IEEE 802.30 ou IEEE 803.3 ?	113



IEEE 802.12 pour les réseaux TCP/IP	113
Commutateurs Ethernet	115
TCP/IP sur les réseaux Token Ring	119
Fonctionnement des réseaux Token Ring	119
Options Token Ring	126
Éléments d'un réseau local Token Ring	127
Règles de câblage de l'IEEE 802.5	127
Pour éviter les problèmes dans les réseaux Token Ring TCP/IP	129
La taille d'un Token Ring	129
Diagnostics internes pour les réseaux Token Ring	130
Trame TCP/IP dans les réseaux IEEE 802.5	132
TCP/IP sur les réseaux FDDI	133
Fonctionnement de FDDI	134
Trame TCP/IP dans les réseaux FDDI	137
Extension des réseaux locaux vers les réseaux étendus	137
Rôle des répéteurs, ponts et routeurs	137
Le répéteur	138
Le pont	139
Routeurs et passerelles	141
Connexions locales et connexions distantes	143
Fonctionnement d'un pont	144
Récapitulatif	148
Testez vos connaissances	149

2 INFRASTRUCTURE DES RÉSEAUX TCP/IP INTERCONNECTÉS

4 L'adressage IP

153

Qu'est-ce qu'une adresse de réseau ?	153
Valeurs d'adresses pour les NSAP	155
Indépendance des adresses NSAP vis-à-vis du matériel du réseau	155
L'adresse IP	156
Structure de l'adresse IP	157
Classes d'adresse IP	159
Pourquoi utiliser des classes d'adresses spécifiques	161
Notation décimale à points pour les adresses IP	161
Calculer une classe d'adresses	164

Adresses IP spéciales	166
Adresses pour le réseau lui-même	167
Diffusion dirigée	168
Diffusion limitée	170
Adresse IP à zéro	170
Adresse IP sur "le réseau d'ici"	172
Bouclage (loopback) logiciel	172
Une exception dans l'adressage IP	174
Unicast, diffusion et multidiffusion	174
Affectation des adresses IP	179
Adresses de réseau privé	179
Allocation d'adresses de classe C	182
Configuration des adresses IP	183
Adresses d'IP version 6	185
Récapitulatif	188
Testez vos connaissances	189

5 *Protocoles de résolution d'adresse* **193**

Nécessité de la résolution d'adresse	193
Address Resolution Protocol (ARP)	194
Le format ARP	196
Fonctionnement d'ARP	201
Commentaires sur la conception du protocole ARP	207
Surveillance du réseau avec ARP	207
Temporisations dans la table de cache ARP	208
ARP dans les réseaux à ponts	211
Doublons d'adresses et ARP	212
Trace de protocole pour ARP	220
Diffusion ARP initiale	221
Bourrage des trames Ethernet	222
Paquet de requête ARP	223
Paquet de réponses ARP	224
Résolution inverse d'adresses	225
Fonctionnement de RARP	227
Tempêtes RARP	229
Serveurs RARP principaux et auxiliaires	230
Trace de paquets RARP	230
Requête RARP	232
Réponse RARP	233



Affichage du cache ARP	235
Autres protocoles de résolution d'adresse	235
Récapitulatif	236
Testez vos connaissances	236
6 Internet Protocol	241
L'abstraction de IP	241
Taille du datagramme IP	244
Fragmentation IP	247
Format du datagramme IP	248
Format de l'en-tête IP	249
Options IP	266
Ordre réseau des octets	278
Trace IP	279
Datagramme IP no 1 : Paquet IP avec Flags = 0, MF = 0, DF = 0	281
Datagramme IP no 2 : paquet IP de réponse avec Flags = 0, MF = 0, DF = 0	283
Datagrammes IP n° 3 et 4 : étude du champ Identification	285
Datagramme IP n° 5 : DF = 1	289
Récapitulatif	290
Testez vos connaissances	290
7 Le routage IP	295
Concepts de base du routage	295
Routage statique et routage dynamique	296
Définition des protocoles de routage	296
Configuration des routeurs	297
Remise des datagrammes	298
Types de remises de datagrammes	299
Un exemple de remises directe et indirecte	301
Analyse des remises directe et indirecte	307
Tables de routage de l'hôte et du routeur	308
Un exemple de table de routage d'hôte	311
Un exemple de table de routage de routeur	312
Routes spécifiques des hôtes	314
Routes par défaut	316
L'algorithme de routage IP	318
Traitement des datagrammes entrants	319
Récapitulatif	321
Testez vos connaissances	322

8	<i>Le protocole ICMP</i>	325
	Présentation de ICMP	325
	RFC traitant de ICMP	327
	Détection d'erreur par ICMP	327
	Encapsulation ICMP	329
	Services ICMP	332
	Types ICMP 0 et 8 : Echo/PING	337
	Type ICMP 3 : Destination inaccessible	340
	Type ICMP 4 : Source Quench	347
	Type ICMP 5 : Redirection	349
	Récapitulatif	352
	Testez vos connaissances	353
9	<i>Sous-réseaux et surréseaux</i>	357
	Sous-réseaux	357
	Pourquoi des sous-réseaux ?	358
	Définition et avantages de la mise en sous-réseau	361
	Masques de sous-réseau	362
	Les masques de sous-réseaux dans les RFC	365
	Conversions entre nombres décimaux et binaires	365
	Un exemple d'utilisation de sous-réseaux : utilisation d'une adresse de classe B avec un masque de sous-réseau aligné à l'octet	368
	Un exemple de sous-réseau : utilisation d'une adresse de classe avec un masque de sous-réseau non aligné à l'octet	369
	Masques de sous-réseaux non alignés à l'octet	371
	Espace d'adressage perdu avec un masque de sous-réseau unique	372
	Affectation optimisée des numéros de sous-réseaux	374
	Masques de sous-réseau de longueur variable	376
	Le routage dans les sous-réseaux	379
	Utilisation des masques de sous-réseaux dans une table de routage	379
	L'algorithme de routage IP complet, tenant compte des sous-réseaux	382
	Autres techniques liées aux sous-réseaux	383
	Proxy ARP	383
	Routeurs transparents	386
	Surréseaux	388
	Classless Internet Domain Routing (CIDR)	389
	Diffusion et multidiffusion	392
	Diffusion dans des sous-réseaux	393
	Multidiffusion	395
	Internet Group Management Protocol (IGMP)	396



Récapitulatif	398
Testez vos connaissances	398
10 Protocoles de routage IP	403
Systèmes autonomes	403
Routing Information Protocol	405
Implémentations des protocoles de routage	408
Routeurs RIP actifs et passifs	409
Problèmes posés par le calcul en nombre de pas	410
Convergence lente avec RIP	412
Résoudre le problème de la convergence lente de RIP	414
Format de message RIP	417
Trace de message RIP	421
Open Shortest Path First	424
Protocoles d'état de liaison	424
Caractéristiques d'OSPF	427
Format de message OSPF	428
Modèle de table de routeur	436
Récapitulatif	439
Testez vos connaissances	439
11 Protocoles de transport	443
Transmission Control Protocol	444
Fonctionnalités TCP	445
L'environnement de l'hôte TCP	453
Ouverture et fermeture d'une connexion TCP	455
Le format des messages TCP	457
Les ACK cumulés	468
Les temporisateurs adaptatifs	470
Réduire l'impact d'une congestion avec TCP	472
Le syndrome SWS	474
Gérer une connexion TCP rompue	477
TCP comme automate à états finis	478
Les traces TCP	479
User Datagram Protocol	496
Le format de l'en-tête UDP	497
Position et encapsulation des couches UDP	499
La trace UDP	500
Récapitulatif	503
Testez vos connaissances	503



LES SERVICES TCP/IP

12	Configuration automatique	511
	Configuration dynamique avec le protocole BOOTP	511
	Les adresses IP de requêtes et de réponses dans le protocole BOOTP	512
	Gérer la perte des messages BOOTP	514
	Format des messages BOOTP	515
	Phases de la procédure BOOTP	517
	Le champ Vendor Specific Area	519
	Configuration dynamique avec le protocole DHCP	521
	Gestion des adresses IP avec le protocole DHCP	522
	Processus d'acquisition des adresses IP dans le protocole DHCP	523
	Format des paquets DHCP	526
	La trace du protocole DHCP	530
	Récapitulatif	540
	Testez vos connaissances	541
13	Services d'application	545
	DNS	545
	Protocoles de courrier	551
	SMTP	551
	POP3 (Post Office Protocol Version 3)	558
	IMAP4 (Internet Message Access Protocol Rev 4)	562
	Protocoles d'accès distant	565
	Telnet	566
	Utilitaires Berkeley r*	575
	Protocoles de transfert de fichiers	578
	File Transfer Protocol	579
	Trivial File Transfer Protocol	590
	Network File System	594
	Protocoles d'accès Internet	600
	WWW (World Wide Web)	600
	Récapitulatif	617
	Testez vos connaissances	618
14	Gestion de réseau TCP/IP	623
	Introduction à la gestion de réseau TCP/IP	623
	Le modèle de gestion de réseau	623
	SNMP et nœuds gérés	625



Management Information Base	630
Le paradigme de gestion de SNMP	630
Les commandes et protocoles SNMP	632
Traps SNMP	633
Messages et identifiants d'objets SNMP	635
Identifiants d'objets MIB	635
Les messages SNMP	636
Récapitulatif	639
Testez vos connaissances	640

4 L'AVENIR

15 IPv6 et ATM 643

IPv6	643
Format de l'en-tête IPv6	643
En-tête de routage (Routing)	649
En-tête de fragmentation (Fragment)	651
En-tête d'options de destination (Destination Options)	652
En-tête de fin d'en-tête (No Next)	654
En-tête d'options pas à pas (Hop-by-Hop Options)	655
En-tête d'identification (Authentication)	656
En-tête de sécurité ou Encrypted Security Payload	657
Ordre des en-têtes d'extension IPv6	659
IP sur ATM	660
Les circuits ATM	661
La couche d'adaptation ATM (AAL)	662
Le codage des types de données pour ATM	665
Récapitulatif	667
Testez vos connaissances	668

5 ANNEXES

A Protocoles standards 673

Index 689



Introduction

Cet ouvrage introduit les bases de la suite de protocoles TCP/IP. Cette suite permet à des machines connectées à un réseau de communiquer entre elles. A l'origine, TCP/IP était utilisé sur des réseaux étendus comme l'Internet. Désormais, il est adopté dans des réseaux d'entreprise. La démarche suivie dans ce livre vise à vous permettre de maîtriser les techniques et pratiques propres aux réseaux TCP/IP.

Ce livre s'adresse aux utilisateurs avertis et aux administrateurs système désireux de déployer la suite TCP/IP dans leurs réseaux. Si vous découvrez les réseaux TCP/IP, ce livre vous fera comprendre le fonctionnement de TCP/IP dans un réseau, en dehors même des systèmes d'exploitation ou des implémentations de ces protocoles. Les présentations en profondeur de TCP/IP s'appliquent à toutes les implémentations. De nombreux manuels se cantonnent aux problèmes de configuration d'une implémentation particulière de TCP/IP comme dans UNIX ou Windows NT. Cependant, bien que les auteurs s'étendent sur les commandes, les clics de souris ou les séquences de touches nécessaires pour configurer TCP/IP, ils n'approfondissent ni les principes ni les théories qui sous-tendent le fonctionnement de TCP/IP. Bien que ce livre ne se rapporte pas à un système d'exploitation spécifique, il indique, quand cela est nécessaire, les configurations et les différentes implémentations. Les discussions autour de TCP/IP s'appliquent à toutes les plates-formes et à n'importe quelle implémentation de ce protocole.

L'une des caractéristiques de ce livre concerne l'analyse des protocoles TCP/IP, menée à partir de réseaux bien réels. L'organisation des protocoles est expliqué à partir des paquets de données et des valeurs des champs d'en-têtes. Ces paquets, appelés aussi protocole trace, sont présentés pour beaucoup comme la base des protocoles TCP/IP.

Chaque chapitre de ce livre se termine par une série de questions se rapportant aux sujets traités dans le chapitre. Alors que certaines questions relèvent de concepts qui nécessitent une réponse très simple et très rapide, d'autres exigent un peu plus de réflexion pour mieux maîtriser les explications contenues dans le chapitre. C'est en répondant à ces questions que vous comprendrez mieux toutes les finesses de TCP/IP.

Le contenu de ce livre et les questions situées en fin de chapitre peuvent aussi servir de base à des formations dispensées dans les instituts ou les universités. Les

» » »



LE MACMILLAN

chapitres suivent une démarche logique. Ils doivent être lus suivant l'ordre dans lequel ils ont été écrits.

Organisation du livre

Les paragraphes suivants donnent un bref aperçu des différents chapitres du livre.

• **Partie I : Le modèle TCP/IP et l'infrastructure physique**

Le Chapitre 1 aborde la suite de protocoles TCP/IP et décrit les principaux protocoles comme IP, TCP, UDP, Telnet, FTP, DNS ou HTTP. Il indique comment ces protocoles s'imbriquent dans la suite TCP/IP.

Le Chapitre 2 présente la hiérarchie du protocole TCP/IP. Le concept de couche permet de simplifier l'écriture de logiciels TCP/IP et de comprendre la description de la suite TCP/IP. Les modèles OSI et DoD sont comparés.

Le Chapitre 3 décrit les techniques des réseaux locaux et étendus basés sur TCP/IP, comme les réseaux Ethernet, Token Ring, Fast Ethernet et IEEE. Ce chapitre explique aussi le rôle des répéteurs, ponts, routeurs et passerelles et décrit l'encapsulation des paquets TCP/IP dans les trames de la couche de liaison des données.

• **Partie II : Infrastructure des réseaux TCP/IP interconnectés**

Le Chapitre 4 présente la structure, le format et la représentation des adresses IPv4 et IPv6. Ce chapitre explique les différentes classes d'adresses IPv4 et leurs utilisations ainsi que les adresses de diffusion.

Le Chapitre 5 aborde les protocoles utilisés pour relier les adresses IP aux adresses physiques. Le format des paquets ARP et RARP est présenté en détail.

Le Chapitre 6 explique le protocole IPv4 qui fournit une vue d'un réseau dans lequel tous les nœuds sont traités comme des nœuds IP. Ce chapitre décrit aussi les services en mode non connectés.

Le Chapitre 7 explique comment le protocole IP permet de bâtir des réseaux complexes à partir de routeurs. Ce chapitre décrit le rôle d'un routeur IP et explique les différences entre les routages statique et dynamique. Des concepts importants concernant la structure des tables de routage, des routes par défaut et la logique du routage IP sont présentés en détail.

Le Chapitre 8 est consacré au protocole ICMP, qui présente un grand intérêt dans les dépannages et les diagnostics des réseaux IP.

Le Chapitre 9 explique les possibilités d'adressage résultant de l'utilisation des masques de sous-réseaux dans les adresses de classe A, B ou C. Ce chapitre décrit aussi l'utilisation de VLSM (*Variable Length Subnet Masking*) et de CIDR (*Classless Internet Domain Routing*).

Le Chapitre 10 définit les systèmes autonomes et l'utilisation des protocoles IGP. De plus, vous y trouverez des explications sur les protocoles de routage Vector Distance et Link State.

Le Chapitre 11 décrit les caractéristiques des protocoles TCP et UDP. Les protocoles de transfert permettent de relier des ordinateurs entre eux. Le protocole TCP permet un flux de données aussi sûr, les techniques qui augmentent cette sûreté de transmission et d'efficacité sont, elles aussi, présentées en détail.

• Partie III : Les services TCP/IP

Le Chapitre 12 précise les caractéristiques des protocoles BOOTP et DHCP utilisés pour fournir les paramètres TCP/IP à une station quelconque à partir d'un serveur central BOOTP ou DHCP.

Le Chapitre 13 présente quelques services disponibles sur des réseaux TCP/IP comme, par exemple, le système de nom de domaine (DNS), les protocoles de courrier électronique (SMTP, POP3 et IMAP4), les protocoles d'accès distant (Telnet et Berkeley), les protocoles de transfert de fichiers (FTP et TFTP), les protocoles d'accès aux fichiers (NFS et Web NFS) et les protocoles d'accès au Web (HTTP/HTML et Gopher).

Le Chapitre 14 décrit les systèmes de gestion des réseaux, principalement le protocole SNMP (*Simple Network Management Protocol*), largement utilisé dans le monde des réseaux.

• Partie IV : L'avenir

Le Chapitre 15 présente deux technologies très prometteuses : la nouvelle génération IP, appelée IPv6, et ATM. Le protocole IPv6 est analysé en détail, et les bases de l'ATM permettront de comprendre comment les paquets TCP/IP sont envoyés dans des réseaux ATM.

• Annexe

L'annexe contient la liste des protocoles standards et des RFC.



Conventions typographiques

Tout au long de ce livre, plusieurs conventions vous faciliteront la lecture.

- Les informations que vous devez taper sont écrites en caractères **gras**. Cette convention ne s'applique pas aux touches spéciales : Entrée, Echap ou Ctrl.
- Les termes nouveaux sont écrits en *italique*.
- Les textes apparaissant à l'écran sont écrits en police à espacement fixe.

Paragraphes spéciaux utilisés dans ce livre

Vous rencontrerez des paragraphes spéciaux reconnaissables aux icônes se trouvant dans la marge.

INFO



Ces encadrés dispensent un surcroît d'informations sur des situations ou des circonstances particulières.

ASTUCE



Optimiser une procédure... Gagner du temps... Voilà ce que vous enseignent les encadrés "Astuce" en vous dévoilant des techniques particulières. Vous verrez aussi comment éviter un problème potentiel avec votre système.

ATTENTION



Ces encadrés vous mettent en garde contre une procédure qui peut être dangereuse si elle n'est pas lancée avec précaution.

LES COUCHES DE TCP/IP ET L'INFRASTRUCTURE PHYSIQUE

1	<i>Introduction à TCP/IP</i>	7
2	<i>Les couches de TCP/IP</i>	33
3	<i>Support réseau pour TCP/IP</i>	69



I

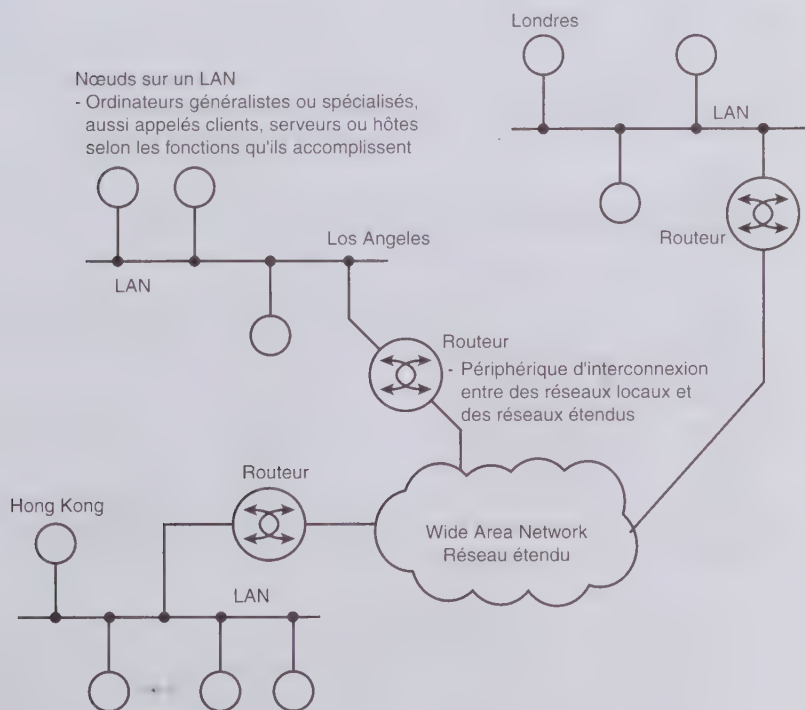
Introduction à TCP/IP

Les réseaux d'ordinateurs ont pris peu à peu une importance considérable dans la vie de tous les jours. Cet engouement provient essentiellement d'un besoin fondamental de communication. Dans une entreprise, il est important de communiquer et de partager l'information. Comme de plus en plus d'informations se prêtent à une représentation électronique, les employés d'une société peuvent se connecter à des mainframes, à des ordinateurs personnels ou à de simples stations. Pour pouvoir partager des données, ces ordinateurs doivent être connectés entre eux. Une fois reliés, ils forment un réseau.

Dans une petite société, ce réseau est dit "local" (LAN). Dès que l'entreprise doit communiquer avec d'autres entités, le réseau devient "étendu" (WAN) [voir Figure 1.1].

Figure 1.1

Réseaux local et étendu.



Ces exemples se rapportent à des entreprises. Cependant, aujourd'hui, les réseaux augmentent sans cesse afin de relier des personnes de différentes organisations comme nous pouvons le voir dans le développement phénoménal de l'Internet, le réseau mondial qui relie des individus, des entreprises, des organisations à but non lucratif, les gouvernements, les organisations internationales et les instituts d'éducation. Mais il n'est pas nécessaire que les utilisateurs des réseaux soient des personnes ; ce peut être des périphériques effectuant certaines fonctions de contrôle. Par exemple, pour l'aviation, les bateaux ou l'automobile, plusieurs ordinateurs peuvent gérer eux-mêmes les systèmes qui récupèrent des informations de contrôle.

Ce livre décrit les réseaux d'ordinateurs qui sont bâtis autour des technologies connues sous le nom de TCP/IP (*Transmission Control Protocol/Internet Protocol*). Les réseaux basés sur ces technologies portent le nom de réseaux TCP/IP.

Ces technologies jouent un grand rôle dans le monde de l'Internet et de plusieurs réseaux d'entreprises. Le fait que les réseaux TCP/IP sont basés sur des spécifications ouvertes et non propriétaires peut expliquer leur succès.

Ce chapitre examine les origines des réseaux TCP/IP et les utilisations commerciales de ce protocole. Il fait aussi un rapide tour d'horizon des principaux services utilisant le protocole TCP/IP. Ce chapitre constitue une introduction conceptuelle de cette technologie. Les chapitres suivants expliquent plus concrètement comment fonctionne TCP/IP.

Vue d'ensemble des réseaux TCP/IP

Avant de décrire en détail les réseaux TCP/IP dans les chapitres suivants, voici un rapide tour d'horizon des protocoles TCP/IP, qui permettra de mieux comprendre leur utilisation dans un réseau.

• Qu'est-ce que TCP/IP ?

Quand on parle de TCP/IP, on se réfère souvent à différents concepts. En première analyse, le terme TCP/IP peut signifier *protocole de communication pour la transmission de données*. TCP signifie *Transmission Control Protocol* ; et IP, *Internet Protocol*.

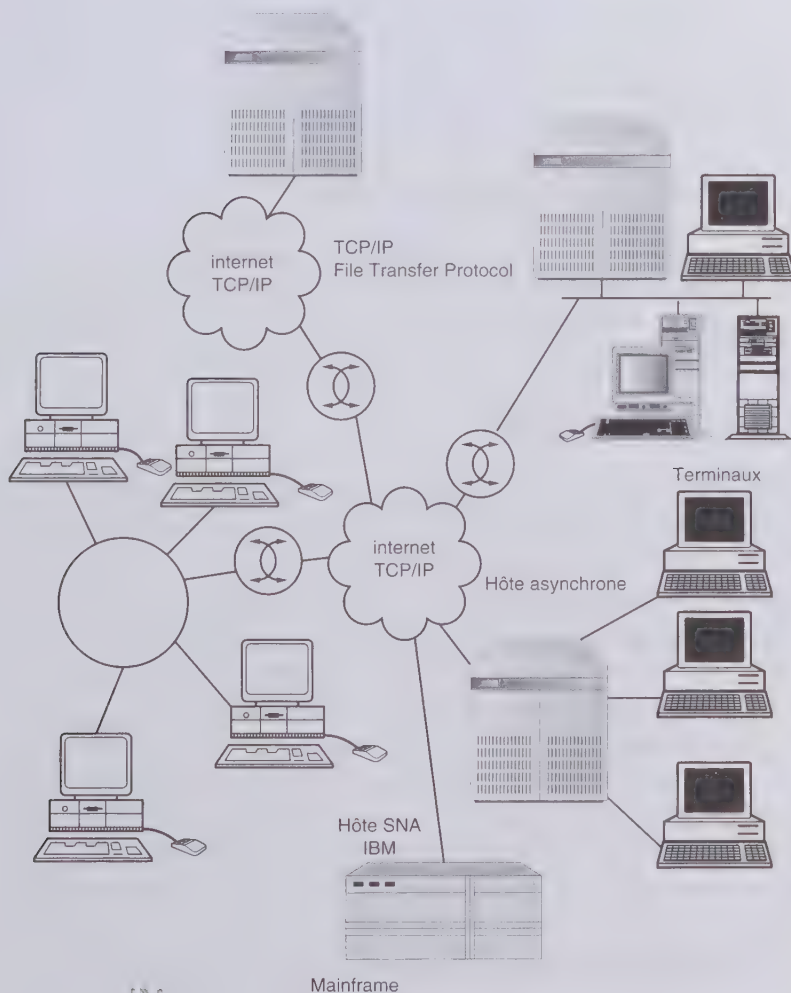
Un protocole de communication est un ensemble de règles permettant à plusieurs ordinateurs de dialoguer entre eux. Les peuples ont besoin d'un langage pour se comprendre, il en est de même pour les ordinateurs. A l'ONU, les diplomates des différentes nations se comprennent grâce aux interprètes qui traduisent sur-le-champ les discours des uns et des autres. Les ordinateurs ne font que copier les méthodes de communication des humains. TCP/IP est l'un des langages utilisés dans les réseaux.

Le terme TCP/IP n'est pas limité à l'expression *Transmission Control Protocol/Internet Protocol*. TCP/IP recouvre, en fait, toute une famille de protocoles comme UDP (*User Datagram Protocol*), FTP (*File Transfer Protocol*), Telnet (*Terminal Emulation Protocol*), HTTP (*Hypertext Transfer Protocol*), etc.

La Figure 1.2 montre un exemple d'un réseau TCP/IP utilisant plusieurs de ces protocoles.

Figure 1.2

Exemple de réseau TCP/IP.



Les réseaux utilisant TCP/IP sont souvent appelés des réseaux *internets* TCP/IP. La Figure 1.2 en est un exemple. Ne confondez pas un *internet* TCP/IP et l'*Internet*. Un internet TCP/IP est un réseau utilisant les protocoles TCP/IP, qu'il soit ou non relié à d'autres réseaux, tandis que l'*Internet* (avec un *I* majuscule) représente le plus grand réseau mondial reliant des milliers d'ordinateurs entre eux sur tous les continents. Il est basé sur TCP/IP.

Quand les entreprises utilisent les services de l'Internet (en particulier le World Wide Web basé sur le protocole HTTP) sur leur propre réseau privé, on dit que leur réseau forme un *intranet*.

Les technologies TCP/IP décrites dans ce livre couvrent aussi bien les réseaux internets, intranets ou l'Internet.

• TCP/IP hier et aujourd'hui

Dans les années soixante, la DARPA (*Defense Advanced Research Projects Agency*) remarqua que les ordinateurs utilisés dans le domaine militaire, de marques différentes, ne pouvaient communiquer qu'avec des ordinateurs de la même marque. Aucun protocole commun ne permettait de faire dialoguer ces ordinateurs.

• Rôle de la DARPA

Pour résoudre ces problèmes de communication, le ministère de la Défense des Etats-Unis demanda à la DARPA de définir une famille de protocoles pour :

- **Simplifier les communications.** Grâce à un jeu de protocoles communs, tous les appareils pourront communiquer entre eux.
- **Développer la compétition entre les différentes sociétés d'informatique.** Les constructeurs d'ordinateurs peuvent entrer en compétition pour améliorer encore leurs implémentations des protocoles standards.
- **Interopérabilité.** En proposant aux constructeurs un ensemble de protocoles communs, l'interopérabilité entre différents équipements devient possible.
- **Efficacité et productivité.** Avec un seul ensemble de protocoles, les constructeurs peuvent consacrer toute leur attention à l'implémentation des protocoles standards sur leurs machines et augmenter ainsi leur productivité.

• Les premières expérimentations de la DARPA

En 1969, une première expérimentation a permis de relier les quatre sites suivants :

- l'université de Californie Los Angeles (UCLA) ;
- l'université de Californie Santa Barbara (UCSB) ;

- l'université d'Utah ;
- SRI International.

La Figure 1.3 montre les différents sites connectés. Cette expérience marqua le début du fameux projet ARPAnet (*Advanced Research Projects Agency Network*). L'expérience a été un succès, et d'autres sites se sont intégrés à ce réseau.

Figure 1.3

Les quatre sites de l'expérience ARPAnet.



En 1972, une démonstration reliait cinquante nœuds et vingt hôtes. Ce fut encore un succès et le point de départ d'un déploiement plus important d'hôtes sur le réseau ARPAnet.

Le terme *nœud* est un nom générique pour tout périphérique connecté à un réseau. Ce périphérique peut être un périphérique de routage ou un ordinateur (ordinateur personnel, station de travail, mini-ordinateur ou mainframe).

Un autre terme, *hôte*, est aussi très largement employé dans le monde des réseaux. Historiquement, ce terme se rapporte à un ordinateur puissant sur lequel sont connectés plusieurs terminaux. Aujourd'hui, ce terme est utilisé pour toute machine qui offre un service à des utilisateurs. Ce type de machine peut être aussi bien un ordinateur personnel, une station de travail, un mini-ordinateur ou un mainframe.

Vous devez encore connaître deux autres termes : *serveur* et *client*. Le premier se rapporte à tout type de machine sur laquelle tourne un logiciel *serveur* offrant des services à des logiciels utilisateurs appelés *clients*.



- L'évolution d'ARPAnet

L'ARPAnet continua à se développer et subit plusieurs transformations. Avant 1984, l'ARPAnet consistait en réseaux militaires connectés entre eux (voir Figure 1.4). Après 1984, les réseaux militaires formèrent un réseau unique indépendant des autres réseaux. Le réseau de la Défense nord-américaine fut créé avec des liens vers ARPAnet (voir Figure 1.5).

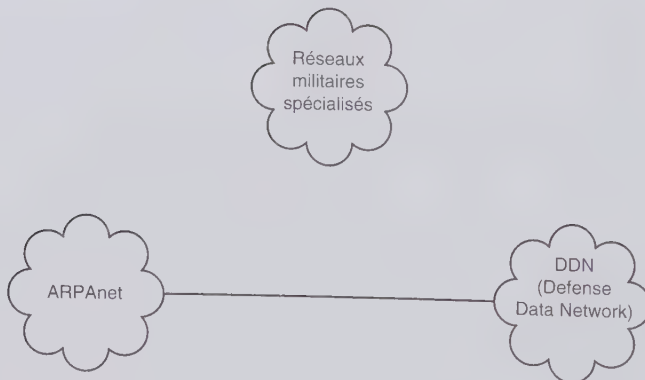
Figure 1.4

ARPAnet avant 1984.



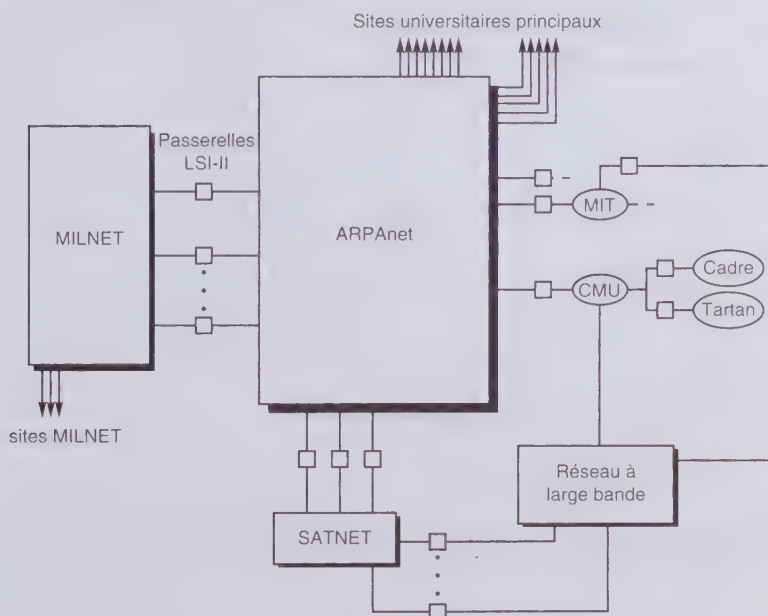
Figure 1.5

ARPAnet en 1984.



A partir de 1986, le réseau ARPAnet englobait la plupart des grandes universités nord-américaines, le réseau militaire MILNET et d'autres centres de recherche internationaux (voir Figure 1.6).

Figure 1.6
ARPAnet en 1986.



MILNET = Réseau militaire (Réseau non classé)

Peu à peu, le réseau ARPAnet fut remplacé par l'Internet. Celui-ci dépassa le domaine exclusif des universités et passa très vite dans le domaine commercial, à tel point que le trafic commercial l'emporta sur tous les autres.

Le premier réseau dorsal (*backbone*) aux Etats-Unis fut NSFNET, puis ANSNET pour le trafic commercial.

Actuellement, la communauté Internet regroupe à la fois des organisations commerciales et de simples utilisateurs. On y trouve les universités, les organismes de recherche, les fournisseurs d'accès, les ministères et les utilisateurs.



Le Tableau 1.1 recense les principaux fournisseurs d'accès à l'Internet. Cette liste ne peut évidemment pas être exhaustive étant donné l'espérance de vie des fournisseurs d'accès, qui apparaissent ou disparaissent très vite du paysage de l'Internet.

Tableau 1.1 : Quelques fournisseurs d'accès à l'Internet

Fournisseur d'accès	Contact
MSN	01 43 62 34 50
Wanadoo	0 801 63 34 34
Club Internet	01 55 45 46 47
AOL	01 43 16 44 44
CompuServe	0 803 00 60 00
HOL	0 836 69 13 00
Worldnet	01 40 37 90 90
Imaginet	01 43 38 10 24
France Pratique	01 44 69 15 00

• Passage des réseaux propriétaires aux réseaux TCP/IP

Les premiers systèmes informatiques étaient fondés sur des solutions propriétaires. Deux exemples classiques illustrent cela : le réseau SNA (*Systems Network Architecture*) d'IBM et le réseau DECnet de Digital (voir Figure 1.7).

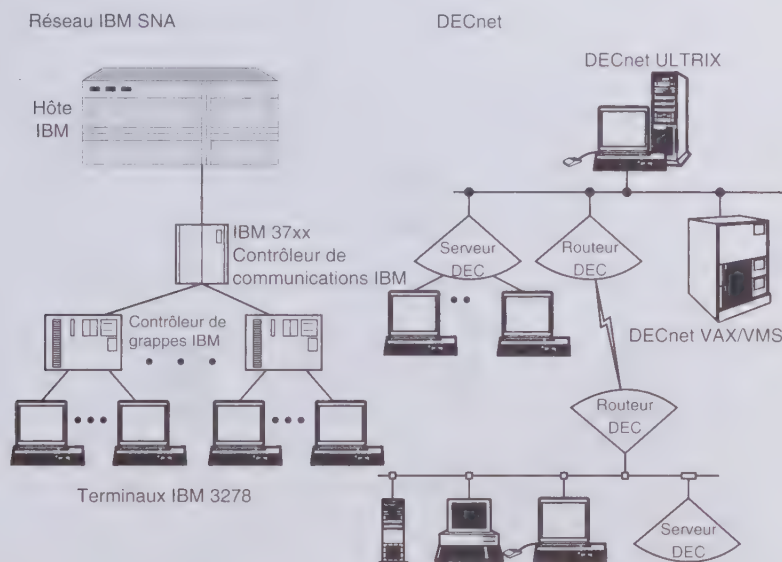
À l'origine, le réseau SNA d'IBM est hiérarchique. L'hôte IBM est relié à des contrôleurs IBM 37xx eux-mêmes reliés à des contrôleurs de grappes (*cluster controllers*), qui jouent le rôle de serveurs de terminaux pour les terminaux 3278/3279 IBM.

Le réseau DECnet Phase IV de Digital est bâti autour des protocoles DECnet, tandis que le réseau DECnet Phase V utilise à la fois les protocoles DECnet et OSI (*Open Systems Interconnection*). Comparé au réseau SNA, DECnet constitue surtout un réseau poste à poste aussi bien au niveau physique qu'au niveau des protocoles.

SNA a évolué pour se rapprocher de la solution poste à poste grâce à l'introduction d'APPC (*Advanced Peer-to-Peer Communications*) et d'APPN (*Advanced Peer-to-Peer Networking*).

Ces deux réseaux propriétaires ont évolué vers des solutions ouvertes avec l'adoption des standards TCP/IP. On peut maintenant accéder à ces réseaux grâce aux protocoles et aux services TCP/IP.

Figure 1.7
Exemples de réseaux
propriétaires.



IBM et DEC ont intensifié leurs efforts dans le développement de TCP/IP sur leur système. Par exemple, IBM propose des solutions TCP/IP intégrant des contrôleurs d'interconnexion basés sur des plates-formes MCA tournant sous TCP/IP ou OS/2.

• Réseau multiconstructeur

Les constructeurs ont peu à peu développé des solutions basées sur TCP/IP. Les utilisateurs ne désiraient plus être prisonniers d'une solution propriétaire certes avantageuse pour le constructeur, mais trop contraignante et trop coûteuse pour l'utilisateur.

La Figure 1.8 montre l'exemple d'un réseau reliant, grâce à TCP/IP, plusieurs plates-formes différentes. Sur ce réseau coexistent un HP-9000 tournant sous HP-UX, des stations SUN tournant sous SunOS ou Solaris, un VAX VMS tournant sous TCP/IP pour VMS, un serveur NetWare, des stations UNIX, des IBM PC, un Mac TCP et Windows NT. Sur cette figure, la place manque pour représenter toutes les plates-formes qui pourraient être reliées sur ce réseau !

Originellement, TCP/IP a été implémenté dans le noyau du système d'exploitation Unix BSD 4.2. Ce système a constitué une version de base d'Unix, ce qui explique la popularité de TCP/IP. Plusieurs organismes de recherche utilisent le système UNIX BSD et, aujourd'hui, plusieurs machines de l'Internet sont basées sur des Unix dérivés de BSD.



Figure 1.8

Exemple de réseau
multiconstructeur.

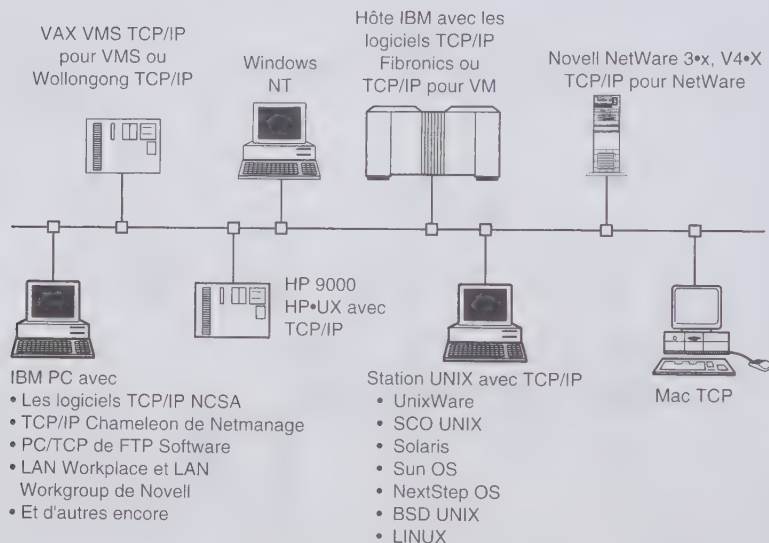
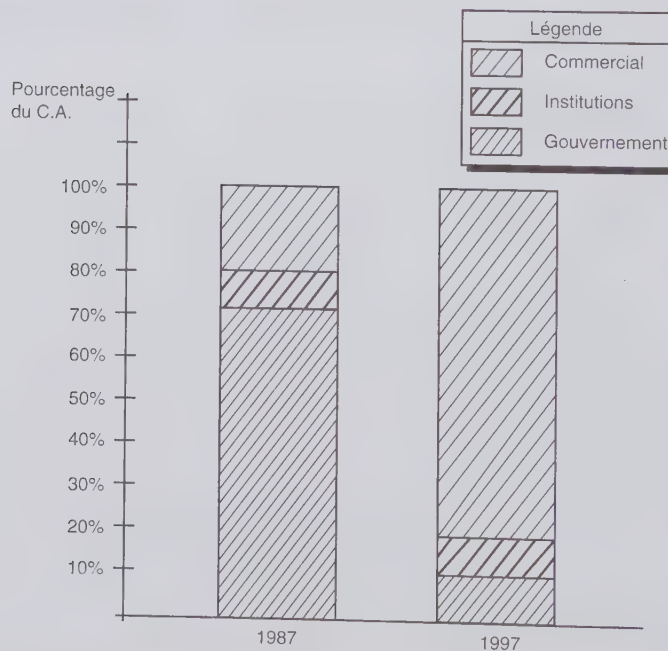


Figure 1.9

Evolution de la demande
en TCP/IP.

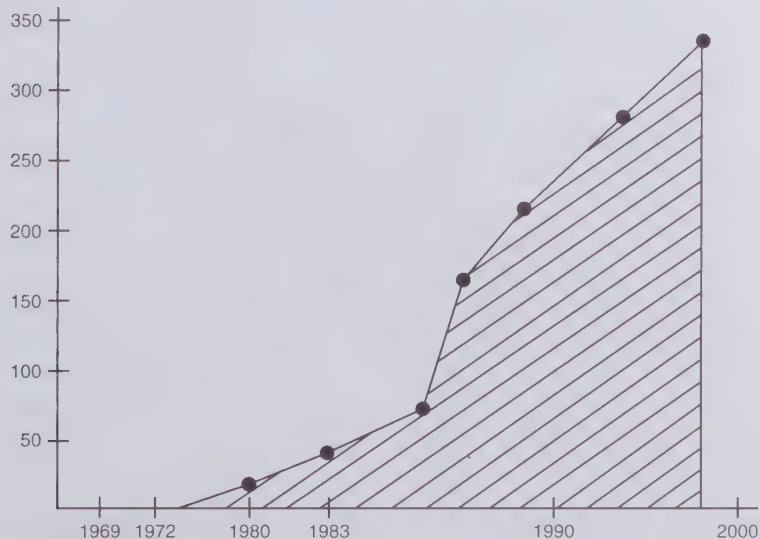


Source : Infonetics, Network World, Newton-Evans Research Co.

• Evolution de l'impact de TCP/IP

Les Figures 1.9 et 1.10 précisent l'évolution de TCP/IP dans le monde des réseaux et l'augmentation du nombre de sociétés proposant TCP/IP. Ces deux graphiques montrent l'intérêt de TCP/IP. La Figure 1.9 fait apparaître l'importance grandissante de TCP/IP dans le domaine commercial.

Figure 1.10
*Nombre de sociétés
proposant TCP/IP.*



Source : Infonetics

La Figure 1.10 met en évidence l'augmentation rapide du nombre de sociétés proposant des solutions fondées sur TCP/IP. Parmi les raisons qui expliquent cette augmentation, les sociétés proposent maintenant des applications basées sur les protocoles TCP/IP.

• Montée en puissance de TCP/IP

TCP/IP propose aujourd'hui ce que les protocoles OSI avaient promis il y a plusieurs années. La faiblesse majeure de TCP/IP par rapport au modèle OSI, c'est le domaine des services. OSI proposait une plus grande variété de services



que TCP/IP. Dans les dernières années, grâce aux efforts de la communauté Internet, TCP/IP a comblé le fossé qui existait entre les services proposés par l'OSI et les services proposés par TCP/IP. Certaines applications OSI, comme X.500, sont maintenant implémentées dans les réseaux TCP/IP.

Trois facteurs expliquent la montée en puissance de TCP/IP :

- **L'interopérabilité.** Un protocole commun permet l'interopérabilité de produits provenant de différents constructeurs. Pour une interopérabilité véritable, les constructeurs doivent utiliser non seulement TCP/IP, mais aussi les services TCP/IP.
- **L'intérêt commercial sur l'Internet.** L'Internet est basé sur les protocoles TCP/IP et sur ses services. Comme le grand public commence à s'intéresser à l'Internet, on peut imaginer un intérêt accru pour les protocoles TCP/IP et pour ses services.
- **L'augmentation du nombre d'outils de gestion de réseau.** Le plus important protocole de gestion des réseaux est actuellement SNMP (*Simple Network Management Protocol*). Ce protocole utilise les protocoles TCP/IP. Les principaux constructeurs de produits réseau tels que les hubs, les ponts ou les routeurs proposent des agents SNMP. Ces agents utilisent la pile TCP/IP.

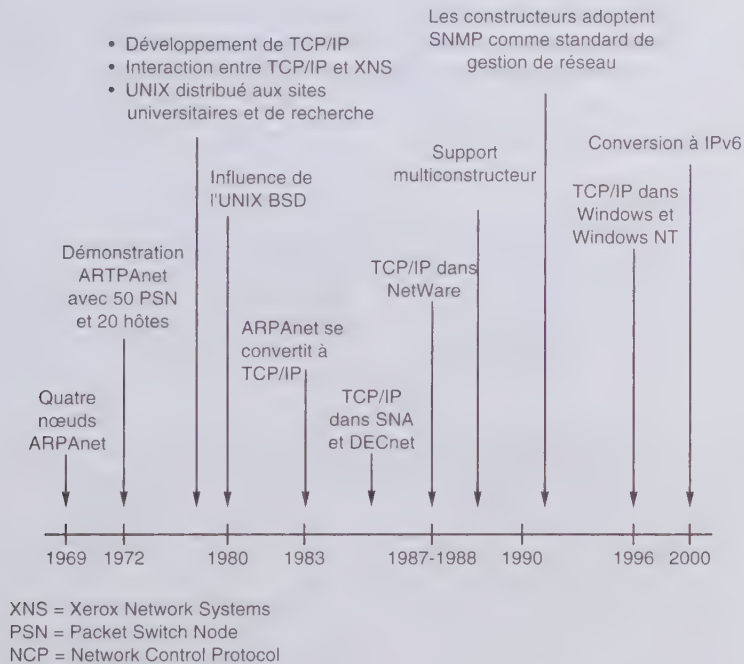
• Evolution de TCP/IP

La Figure 1.11 montre l'évolution de TCP/IP. On y trouve les événements importants qui ont marqué cette évolution. Ce graphique commence au moment des quatre nœuds de l'ARPAnet. La période 1978-1980 se caractérise par les fructueux échanges entre les développeurs de TCP/IP et les chercheurs du centre Xerox de Palo Alto qui travaillaient sur XNS (*Xerox Network System*). A la suite de ce travail commun, RIP (*Routing Information Protocol*) de XNS fut adopté dans l'Unix BSD. Comme Unix était largement exploité, RIP devint un protocole très utilisé dans les réseaux TCP/IP. L'influence d'Unix BSD devint importante en 1980, quand ce système d'exploitation se généralisa dans plusieurs sites.

En 1983, l'ARPAnet – qui utilisait NCP (*Network Control Protocol*) – se convertit à TCP/IP.

Vers la fin des années quatre-vingt, le mouvement s'accéléra. Au début des années quatre-vingt-dix, SNMP devint le standard des gestionnaires de réseaux. SNMP n'est pas limité aux réseaux TCP/IP. Il peut aussi tourner sur des réseaux Novell IPX (*Internetwork Packet Exchange*) et CLNP OSI.

Figure 1.11
Evolution de TCP/IP.



Vue d'ensemble des applications TCP/IP

Plusieurs applications TCP/IP comme FTP ou Telnet ont déjà été mentionnées dans ce chapitre. Ce paragraphe donne une vue d'ensemble de ces applications. Les chapitres suivants décriront en détail les services liés à ces applications.

La Figure 1.12 montre quelques-unes des principales applications TCP/IP. On y voit un réseau internet avec plusieurs utilisateurs accédant aux applications TCP/IP sur des hôtes distants.

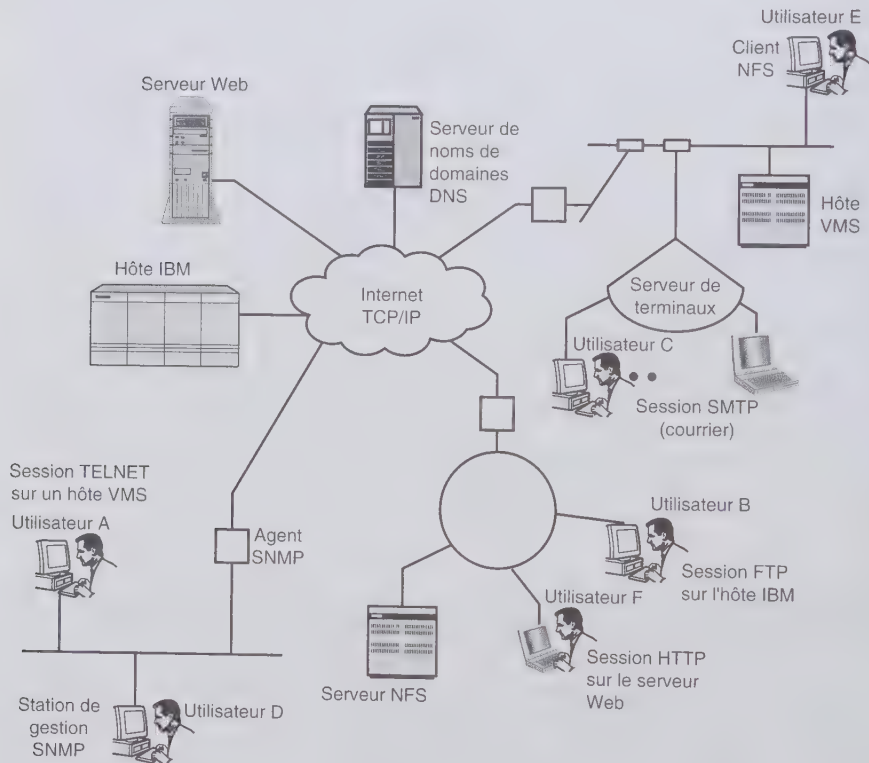
Supposons que l'utilisateur A désire ouvrir une session sur un hôte distant VMS. Telnet va lui permettre d'accéder à cet hôte distant. Pour démarrer une session Telnet, il faut une application Telnet *cliente* tournant sur la station de l'utilisateur. Cette application envoie les séquences de touches tapées par l'utilisateur à l'hôte distant. Sur l'hôte distant doit se trouver une application Telnet *serveur* qui reçoit



les séquences de touches et les envois au système d'exploitation, comme si elles étaient directement tapées à partir d'un terminal relié à l'hôte. Le serveur Telnet reçoit les réponses de l'hôte et les envois au client Telnet.

Figure 1.12

*Vue d'ensemble
des applications TCP/IP.*



Pour transférer des fichiers entre sa station et un hôte IBM, l'utilisateur B exécute une session FTP. L'application FTP permet à un utilisateur d'accéder à un système de fichiers hôte de façon interactive. Dès que la session FTP est établie, l'utilisateur tape des commandes spéciales FTP. Elles lui permettent de parcourir les répertoires ou les fichiers de l'hôte distant. L'utilisateur peut lancer des commandes de *téléchargement* de fichiers entre l'hôte et sa station. Pour lancer une session FTP, il faut, là encore, une application FTP *cliente* tournant sur la station et une application FTP *serveur* tournant sur l'hôte. L'application cliente peut alors envoyer des commandes au serveur FTP situé sur l'hôte.

L'utilisateur C utilise, lui, le protocole SMTP (*Simple Mail Transfer Protocol*) pour envoyer du courrier à un autre utilisateur sur le réseau. Le logiciel de courrier installé sur la station de l'utilisateur C prend le message tapé par l'utilisateur et l'envoie dans un serveur de messagerie, puis un programme tournant en tâche de fond sur ce serveur se saisit du courrier et le dépose dans la boîte aux lettres du destinataire.

L'utilisateur D utilise SNMP (*Simple Network Management Protocol*) pour recueillir des informations sur les différents périphériques installés sur le réseau. Un logiciel de gestion SNMP tourne sur la station de l'utilisateur et envoie une requête dans un format SNMP vers le périphérique concerné. Un agent SNMP installé sur le périphérique répond à la requête lancée par l'utilisateur. Celui-ci peut alors analyser cette réponse et gérer le périphérique.

Autre application : l'accès direct aux services de fichiers sur un hôte. Pour cela, l'utilisateur E utilise NFS (*Network File System*) pour accéder directement aux services de fichiers sur un hôte distant. Le système de fichiers de l'hôte distant apparaît comme une extension du système de fichiers de la station. L'utilisateur accède aux fichiers de l'hôte distant en utilisant les commandes du système d'exploitation de sa station. Là encore, pour accéder aux fichiers du site distant, vous devez installer un *client* NFS sur la station et un *serveur* NFS sur l'hôte. Ce serveur interagit avec le système de fichiers de l'hôte afin que celui-ci puisse comprendre les commandes de la station.

Pour accéder à une page Web d'un serveur Web distant, l'utilisateur F utilise un navigateur Web. Grâce à ce navigateur et au protocole HTTP, l'utilisateur peut parcourir la page Web et cliquer sur les liens de ces pages pour atteindre d'autres pages situées sur le même serveur Web ou sur un autre serveur. La station comporte un client HTTP, qui interprète les documents envoyés par un serveur HTTP installé sur l'hôte distant. Les pages Web sont codées à partir d'un langage spécial appelé HTML (*HyperText Markup Language*).

Le serveur DNS (*Domain Name System*), représenté sur la Figure 1.12, permet de retrouver l'adresse IP d'un hôte sur le réseau, uniquement à partir de son nom symbolique. Le serveur DNS comporte une table faisant correspondre l'adresse IP et le nom de l'hôte. Quand une application TCP/IP, comme Telnet, doit se connecter sur l'hôte VMS, elle peut utiliser le nom symbolique de l'hôte, et DNS transforme ce nom en adresse IP.

Le Tableau 1.2 résume les applications TCP/IP décrites dans ce paragraphe.



Tableau 1.2 : Résumé des applications TCP/IP

Service	Description
Telnet	Telnet permet à un utilisateur de se connecter à distance à un hôte par l'intermédiaire des protocoles TCP/IP. L'utilisateur travaille sur l'hôte, comme s'il travaillait sur un terminal directement connecté sur l'hôte. Les séquences de touches tapées sur l'ordinateur sont envoyées à l'hôte, et celui-ci renvoie les réponses à l'ordinateur. C'est le protocole Telnet qui est responsable de ces transferts. Telnet utilise le protocole TCP/IP comme protocole de communication.
FTP	FTP permet de transférer des fichiers entre deux machines sur un réseau TCP/IP. Peu importe la taille des fichiers. L'utilisateur se connecte sur un serveur FTP distant grâce à un client FTP et entre dans une session interactive avec ce serveur. Il peut alors visualiser les fichiers ou les répertoires de l'hôte et lancer des commandes de transfert de fichiers.
SMTP	SMTP permet à un utilisateur d'envoyer un message à un destinataire connecté à un réseau TCP/IP. Le message est composé sous forme de texte, et l'utilisateur tape l'adresse électronique du destinataire, l'objet du message et le texte du message lui-même. SMTP utilise TCP/IP comme protocole pour envoyer ce message à un serveur de messagerie. Les serveurs de messagerie agissent comme des relais et délivrent les messages à leur destinataire.
SNMP	SNMP permet la gestion à distance de périphériques tels que des ponts, des routeurs, des hubs ou des passerelles. Un agent SNMP doit tourner sur les périphériques gérés par ce système. Une station SNMP envoie des requêtes pour lire ou modifier les paramètres d'un périphérique. SNMP utilise les protocoles UDP et IP comme protocoles de communication.
DNS	DNS constitue un annuaire électronique permettant de nommer les différentes ressources d'un réseau. DNS associe les noms symboliques des périphériques à leur adresse IP. Les noms DNS jouent un rôle très important dans le courrier électronique, car l'adresse électronique d'un utilisateur contient le nom symbolique du serveur de messagerie gérant sa boîte aux lettres. Par exemple, dans l'adresse électronique Pierre@wanadoo.fr , le nom situé après le signe @ correspond au nom DNS du serveur de messagerie gérant la boîte aux lettres de Pierre. DNS utilise les protocoles UDP et IP comme protocoles de communication.
HTTP	HTTP est le protocole permettant d'envoyer des pages Web à un ordinateur équipé d'un navigateur. Ce navigateur peut lire des documents graphiques, audio et vidéo. HTTP utilise TCP/IP comme protocole de communication.

Ce paragraphe vous a donné un avant-goût des différentes applications TCP/IP. Les chapitres suivants détailleront davantage chacune de ces applications.

L'Internet

L'ARPAnet a été peu à peu remplacé par l'Internet. Plus qu'un simple réseau, l'Internet regroupe des centaines de réseaux. L'Internet est basé principalement sur les protocoles TCP/IP, mais tous les réseaux de l'Internet n'utilisent pas forcément TCP/IP. Par exemple, les réseaux BITNET et le CREN se servent des protocoles SNA d'IBM. Pour communiquer avec les réseaux de l'Internet, ces deux réseaux doivent passer par des passerelles pour communiquer entre eux.

Bien que l'Internet ne cesse d'évoluer vers la commercialisation, il offre un incomparable laboratoire de tests pour le développement de nouveaux protocoles et services. Par exemple, plusieurs expérimentations sur les services OSI sont conduites sur l'Internet.

Le paragraphe suivant donne quelques exemples des réseaux qui composent l'Internet.

• Exemples de réseaux sur l'Internet

Le Tableau 1.3 recense quelques réseaux qui forment l'Internet. Cette liste n'est évidemment pas exhaustive, et certains réseaux présentent un intérêt historique. Cette liste vous donne une idée de la participation de la communauté internationale à l'Internet.

Tableau 1.3 : Exemples de réseaux sur l'Internet

Réseau	Description
NSFNET	(<i>National Science Foundation Network</i> .) C'est le réseau dorsal (<i>backbone</i>) des réserves aux Etats-Unis, finalement remplacé par ANSNET et MCI.
CSNET	(<i>Computer Science Network</i> .) C'est un service Internet utilisant X.25, conçu pour les écoles et les petites organisations.
MILNET	Ministère de la Défense des Etats-Unis. Il a été à l'origine de l'ARPAnet.
BITNET	(<i>Because It's Time Network</i> .) Il utilise des mainframes IBM.



Tableau 1.3 : Exemples de réseaux sur l'Internet (*suite*)

Réseau	Description
EARN	(<i>European Academic Research Network.</i>) C'est un réseau qui couvre l'Europe et l'Afrique.
JANET	(<i>Joint Academic Network.</i>) C'est le réseau des universités au Royaume-Uni.
RENATER	Acronyme signifiant "Réseau national de télécommunications pour la technologie, l'enseignement et la recherche" en France.

NSFNET fournit l'accès aux ressources scientifiques. Ce réseau comprend trois niveaux :

- **L'épine dorsale.** Réseau transcontinental connectant des réseaux administrés séparément aux centres NSF.
- **Réseaux régionaux.** Ce sont des réseaux qui regroupent des centres de recherche.
- **Réseaux de campus.** Réseaux connectés aux réseaux régionaux.

La gestion de NFSNET relève de la responsabilité de ANSNET. Le support de l'utilisateur final de NFSNET est fourni par un centre d'information réseau (NIC). NFSNET possède une architecture intéressante, utilisant des nœuds appelés NSS (*Nodal Switching Subsystems*). Chaque NSS consiste en neuf IBM PC/RT tournant sous AIX et connectés sur un réseau Token Ring. Dans chaque NSS, un processeur de contrôle de routage gère les accès entre plusieurs NSS. Le logiciel de routage dans chaque NSS utilise IS-IS (un protocole de routage appelé *Intermediate System to Intermediate System*). Les réseaux régionaux sont connectés par le protocole *Exterior Gateway Protocol*.

CSNET, construit en janvier 1981, reliait uniquement les départements de sciences des écoles et des universités. Actuellement, ce service s'est ouvert à l'industrie, au gouvernement et à des institutions à but non lucratif engagées dans la recherche. Le réseau se situe aux Etats-Unis et au Canada. Mais il entretient des liens étroits avec l'Australie, la Finlande, la France, l'Allemagne, Israël, le Japon, la Corée, la Nouvelle-Zélande, la Suède, la Suisse, la Chine et le Royaume-Uni.

BITNET est un réseau coopératif qui dessert plus de deux mille trois cents hôtes (IBM), sur des centaines de sites dans trente-deux régions. Le protocole de base est NJE, avec quelques sites tournant sous NJE sur TCP/IP. En Europe, le réseau doit migrer vers les protocoles ISO. Les principaux acteurs de BITNET se situent aux Etats-Unis, à Mexico, au Japon, à Singapour, à Taiwan et en Corée. Les réseaux de l'Asie sont connus sous le nom d'Asianet ; les réseaux canadiens, sous le nom de NetNorth ; les européens, sous le nom d'EARN.

En octobre 1988, les deux réseaux CSNET et BITNET décident de se regrouper en un seul réseau : CREN. Le regroupement est définitif en 1989.

EARN a été créé en 1983, et sa charte indiquait que le réseau serait réservé à l'Europe, au Moyen-Orient et à l'Afrique. Une récente décision intègre aussi le Maroc, la Tunisie, l'Égypte et l'Inde. Administrativement, EARN est enregistré en France, et son équipe de direction est formée de membres de chaque région. EARN utilise les technologies de BITNET. Plusieurs hôtes EARN sont des IBM VM et des DEC VAX VMS. Des passerelles existent vers d'autres réseaux comme JANET au Royaume-Uni et en Irlande.

JANET a été créé pour relier plusieurs universités et centres de recherche au Royaume-Uni. JANET est limité aux membres suivants :

- universités ;
- laboratoires et instituts ;
- membres individuels d'instituts polytechniques.

Les services de JANET comportent la messagerie électronique, le transfert de fichiers, la connexion à distance. Les réseaux connectés à JANET sont des réseaux X.25, des anneaux de type Cambridge Ring (standard CR82), des réseaux Ethernet et IEEE 802.3. Les liens sont de type X.25. JANET utilise un système de noms de domaines similaire au DNS d'Internet, mais l'ordre des noms de domaines se présente inversement de celui de l'Internet, avec le plus haut niveau à gauche du nom. Par exemple, pour envoyer un message de JANET vers l'Internet, la passerelle est à `uk.ac.ucl.cs.nss` à l'université du London Computer Center.

CDNet est administré par l'université UBC (*University of British Columbia*). La plupart des machines sont des DEC VAX ou des SUN. Les liens sont de type X.25 à 2 400 bits/s. Des passerelles existent entre CSNET et BITNET.

• L'IAB et l'Internet Society

La Figure 1.13 montre les relations entre les différents acteurs de l'Internet.

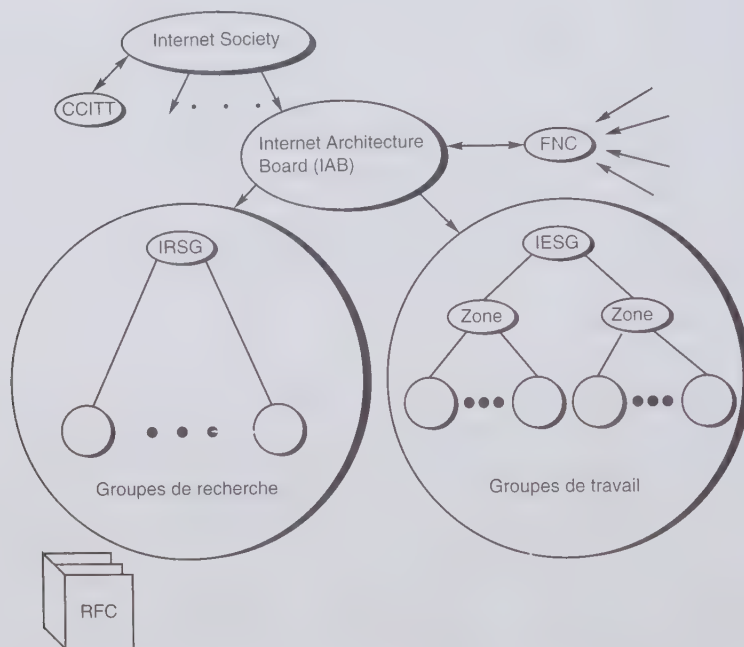
L'architecture de l'Internet fournit une direction et une coordination des protocoles en relation avec TCP/IP. Ce cadre guide les évolutions de l'Internet. Il comprend deux groupes : le groupe de recherche, IRTF (*Internet Research Task Force*) et le groupe de développement, IETF (*Internet Engineering Task Force*).

L'IETF est géré par l'IESG (*Internet Steering Group*). Il est divisé en zones, elles-mêmes segmentées en groupes de travail. L'IETF étudie les problèmes à court et moyen terme.



Figure 1.13

Structure de l'IAB.



CCITT = Comité Consultatif Internationale Telegraphique et Telephonique

IESG = Internet Engineering Steering Group

IRSG = Internet Research Steering Group

FNC = Federal Networking Council

L'Internet Research Task Force est géré par l'IRSG (*Internet Research Steering Group*) et étudie les problèmes à long terme.

L'IETF est divisé en huit zones principales qui peuvent changer en fonction des besoins de l'Internet :

- applications ;
- protocoles hôtes ;
- protocoles Internet ;
- routage ;
- gestion du réseau ;
- interopérabilité OSI ;
- opérations ;
- sécurité.

L'IETF dispose de plusieurs groupes de travail. Parmi eux : l'authentification, les noms de domaines, la configuration dynamique des hôtes, les caractéristiques des

hôtes, l'interconnexion, Telnet, etc. L'IETF comprend l'IESG, son président et les groupes de travail.

En 1992, l'Internet Society a été créée pour promouvoir l'utilisation de l'Internet. La société fournit un forum pour l'industrie, les enseignants, le gouvernement et d'autres utilisateurs. Il propose des recommandations de procédure et des standards techniques pour l'Internet global et les internets privés.

• RFC, STD et IEN

Afin de faire circuler les nouvelles idées sur les protocoles, la recherche et les standards, on commença d'abord à utiliser les "appels à commentaires" : RFC (*Request for Comments*). Quand un chercheur définit un nouveau protocole, il propose ses recherches dans un document RFC. Ainsi, les RFC consistent en définitions de standards de l'Internet, propositions pour de nouveaux protocoles, stratégies d'implémentation, etc.

Les protocoles qui peuvent devenir un standard de l'Internet passent par une série d'états qui décrivent leur niveau de maturité. Ces niveaux sont la proposition, le brouillon puis le standard. Chaque niveau permet un développement supplémentaire de tests. Quand un protocole passe ces trois niveaux, un numéro STD y est attribué.

Au fur et à mesure des avancées techniques, quelques protocoles sont dépassés par d'autres, plus performants. Ces protocoles sont classés *historiques*. Les résultats des recherches et des développements précédents sont archivés dans quelques RFC. Ces protocoles sont désignés comme *expérimentaux*.

Les protocoles développés par d'autres organisations ou par des constructeurs particuliers peuvent être recommandés pour être utilisés sur l'Internet. Les spécifications de ces protocoles sont publiées dans des RFC. Ces protocoles sont appelés *informationnels*.

Quelquefois, des protocoles se développent sur l'Internet sans l'approbation de l'IESG. Cela est dû à la rapidité de l'évolution des protocoles ou encore pour des raisons politiques. Quelques constructeurs essaient de monopoliser le monde de l'Internet. Aussi, l'IAB recommande fortement de suivre les procédures standards pour maintenir l'interopérabilité et pour prévenir toute incompatibilité entre les protocoles.

Tous les protocoles ne sont pas forcément nécessaires. En effet, il existe plusieurs systèmes pour lesquels l'implémentation de certains protocoles n'aurait pas de sens. Par exemple, certains périphériques (comme des passerelles, des routeurs, des serveurs de terminaux, des stations) ont des besoins spécifiques.



Tous les RFC ne décrivent pas les protocoles standards de l'Internet, bien que tous les protocoles standards de l'Internet possèdent un numéro de RFC. La liste officielle des protocoles standards de l'Internet est publiée régulièrement. Cette liste constitue à elle seule une RFC. Quand une nouvelle RFC est publiée sur un même sujet, elle contient un titre sur la RFC qu'elle remplace. Ainsi, la RFC 2200 contient les renseignements suivants :

2200 SJ.Postel, "INTERNET OFFICIAL PROTOCOL STANDARDS,"
(Obsoletes RFC 2000) (STD 1)

Notez que cette RFC indique que la RFC 2200 est un standard officiel des protocoles Internet, et qu'elle rend obsolète la RFC 2000 sur le même sujet. Notez la lettre S après le numéro de la RFC. Cette lettre signifie que la RFC définit un standard Internet. D'autres codes sont utilisés. Par exemple, PS signifie *proposed standard* ; DS, *draft standard* ; I, *informational purpose* ; et E, *documenting results of experiments* (voir Tableau 1.4). La RFC 2200 est aussi décrite comme étant STD 1. Les standards Internet sont numérotés en partant de STD 1, STD 2, etc. Ces standards sont décrits par les RFC dont les numéros se trouvent dans le document STD 1.

Tableau 1.4 : Définitions des états des protocoles de la RFC 2000

Etat	Définition
Standard protocol	L'IESG a établi un standard officiel des protocoles de l'Internet. Ces protocoles ont des numéros STD qui forment deux groupes : un protocole IP qui comprend les protocoles s'appliquant à tout l'Internet, et des protocoles spécifiques réseau comportant des indications sur le mode d'emploi pour les adresses IP s'appliquant à des types particuliers de réseaux.
Draft standard protocol	L'IESG considère ce protocole comme un protocole standard potentiel. Des tests et des commentaires sont désirés par l'IESG. Des modifications peuvent être apportées à ces protocoles avant qu'ils ne deviennent standards.
Proposed standard protocol	Ce sont les protocoles considérés par l'IESG comme pouvant devenir standards dans l'avenir. Des tests et des implémentations sont demandés à plusieurs groupes.
Experimental protocol	Un système ne doit pas utiliser un protocole en expérimentation, à moins que ce système ne participe à l'expérimentation en collaboration avec le développeur de ce protocole. Normalement, ces protocoles sont développés à partir d'un projet de recherche.
Informational protocol	Les protocoles développés par d'autres organisations ou des constructeurs peuvent être soumis à RFC pour la communauté de l'Internet.

Tableau 1.4 : Définitions des états des protocoles de la RFC 2000 (suite)

Etat	Définition
Historic protocol	Ces protocoles ne sont pas devenus des standards. Ils ont été dépassés par des développements plus récents ou parce qu'ils ne présentent pas d'intérêt.

Dans le processus de standardisation, une période de temps suffisante est prévue par la communauté de l'Internet afin d'attendre des réactions aux propositions de standardisation. Cette période est de six mois pour une proposition draft et de quatre mois avant qu'un draft ne puisse être promu à l'état de standard. Ainsi, le temps minimal au bout duquel une proposition peut être adoptée est de dix mois.

Pour qu'un standard puisse être implémenté, une règle générale fait qu'aucun standard proposé ne peut devenir un standard draft sans, au moins, deux implémentations indépendantes et les recommandations de l'IESG.

Un comité spécial formé d'experts de l'IETF, de l'IRTF et de l'IAB est créé en cas de doute au sujet d'un standard proposé.

Le passage d'un protocole dans un standard proposé constitue une étape importante, car elle place le protocole sur la piste d'un protocole standard. En d'autres termes, le protocole peut devenir un standard. A moins d'objections majeures, il deviendra un standard pour la communauté de l'Internet.

La Figure 1.14 montre une grille qui décrit les différents stades d'évolution d'un protocole. A un moment donné, le protocole occupe une cellule de la grille.

Figure 1.14

Etat et statut des RFC.

		STATUT				
		Req	Rec	Ele	Lim	Not
E T A T	Std	X	XXX	XXX		
	Draft	X	X	XXX		
	Prop		X	XXX		
	Info					
	Expr				XXX	
	Hist					XXX

Req : Required
 Rec : Recommended
 Ele : Elective
 Lim : Limited Use
 Not : Not Recommended

Il convient de noter que les documents et standards relatifs à ARPAnet portaient le nom de IEN (*Internet Engineering Notes*).



Tout le monde peut soumettre un document pour publication en tant que RFC. Les soumissions peuvent être envoyées par e-mail. Vous devez lire la RFC 1543 avant de soumettre une RFC. Vous pouvez obtenir des informations à l'adresse suivante :

1543 I.J.Postel, "Instructions to RFC authors," 10/28/1993

(Pages = 16)(Format = .txt)(Obsoletes RFC1111)

Les nouvelles RFC doivent être envoyées à l'éditeur, dont l'adresse est :

RFC-Editor@ISLEDU

• Obtention des RFC

Vous pouvez obtenir les RFC à partir de différentes sources. Si vous ne pouvez accéder à l'Internet, contactez le SRI à l'adresse suivante :

SRI International, Room EJ291

333 Ravenswood Avenue

Menlo Park, CA 94025

415-859-3695

NIC@SRI-NIC.ARP

Le Tableau 1.5 indique quelques sites permettant de récupérer des RFC. Les sites primaires disposent des RFC dès qu'elles sont disponibles. Il en est de même de plusieurs sites secondaires. Certains sites secondaires demandent cependant quelques jours avant de disposer des RFC les plus récentes.

Tableau 1.5 : Sites RFC

Nom de l'hôte	Primaire/secondaire
DS.INTERNIC.NET	Primaire
NIS.NSF.NET	Primaire
NISC.JVNC.NET	Primaire
VENERA.ISI.EDU	Primaire
WUARCHIVE.WUSTL.EDU	Primaire
SRC.DOC.IC.AC.UK	Primaire
FTP.CONCERT.NET	Primaire
FTP.SESQUI.NET	Primaire
SUNIC.SUNET.SE	Secondaire
CHALMERS.SE	Secondaire

Tableau 1.5 : Sites RFC (suite)

Nom de l'hôte	Primaire/secondaire
WALHALLA.INFORMATIK.UNI-DORTMUND.DE	Secondaire
MCSUN.EU.NET	Secondaire
FUNET.FI	Secondaire
UGLE.UNIT.NO	Secondaire
FTP.DENET.DK	Secondaire
MUNNARI.OZ.AU	Secondaire
NIC.CERF.NET	Secondaire
FTP.UU.NET.	Secondaire

Si vous possédez un accès à l'Internet, vous pouvez obtenir des copies des RFC à l'adresse <http://ds.internic.net>.

Récapitulatif

Les réseaux TCP/IP jouent un rôle de plus en plus important dans le monde informatique. Les réseaux TCP/IP sont basés sur des spécifications de protocoles ouverts non propriétaires. Ce chapitre a défini complètement TCP/IP. Il a décrit les origines de ce protocole et ses usages commerciaux. Il a expliqué certaines utilisations de TCP/IP. Ce chapitre a aussi présenté la relation entre l'Internet, ARPAnet et les différents organismes qui influencent la recherche et le développement des protocoles de l'Internet. Vous avez pu découvrir aussi le parcours d'un protocole proposé jusqu'à son adoption comme standard de l'Internet.

Testez vos connaissances

1. Qu'est-ce qu'un protocole de communication ?
2. Donnez un exemple montrant que la communication humaine s'apparente à celle des ordinateurs.
3. Que signifie l'expression TCP/IP ?



4. Donnez une explication des différences, si elles existent, entre internets, intranets et l'Internet.
5. Indiquez les principales raisons pour lesquelles le ministère de la Défense des Etats-Unis a commandité une famille de protocoles.
6. Nommez certaines sociétés qui offrent des solutions TCP/IP pour des réseaux multipropriétaires.
7. Citez quelques lignes de force qui ont conduit au développement de TCP/IP.
8. Donnez une description des services suivants : Telnet, FTP, SMTP, SNMP, DNS et HTTP.
9. Quel est le rôle de l'IAB et de l'Internet Society ?
10. Que sont les RFC, STD et IEN ?
11. Toutes les RFC sont-elles standards ? Donnez des exemples de RFC non standards.
12. Lisez la RFC 1160 et décrivez la charte de l'IAB.
13. Comment pouvez-vous obtenir une copie d'une RFC ?
14. Les RFC ont des états associés. Décrivez ces différents états.

2

Les couches de TCP/IP

Les applications TCP/IP utilisées par les intranets et l'Internet dépendent de plusieurs couches de protocoles. Le terme *couche de protocoles* suggère qu'il y a interaction entre les différents protocoles. Chaque couche de protocoles interagit avec la couche suivante grâce à une interface spécifique.

Construire des systèmes, comme des protocoles, en les disposant en couches présente plusieurs avantages. Chaque couche peut être définie précisément pour effectuer une fonction bien particulière. La couche représente, en quelque sorte, une "boîte noire". Cela signifie que les détails de l'implémentation peuvent être traités indépendamment de la fonction réalisée par la couche. D'un point de vue fonctionnel, vous transmettez des données à la couche de protocoles, et celle-ci applique la fonction sur ces données. Ainsi, avec ces traitements, vous pouvez représenter des données dans un format particulier, acheminer des données de façon fiable à travers un réseau, choisir une route optimale pour acheminer les données, etc.

En séparant les fonctions des couches de l'implémentation, vous pouvez modifier certaines caractéristiques d'une couche sans toucher aux autres couches. Par exemple, pour augmenter les performances d'un réseau, vous pouvez changer un élément du réseau sans modifier les protocoles utilisés.

Pour comprendre l'importance des couches, imaginez un réseau sans système de couches de protocoles. Dans ce cas, les applications réseau dépendent du protocole de communication et des éléments matériels. Et tout changement apporté à un élément affectera l'ensemble. Enfin, vous serez totalement tributaire du constructeur du système.

L'industrie informatique est encore très jeune. Il y a une dizaine d'années, beaucoup de professionnels pensaient qu'une seule solution résoudrait tous les problèmes. Mais l'on s'est vite aperçu qu'une telle solution coûtait trop cher et interdisait toute évolution des produits.

Aujourd'hui, les technologies changent très rapidement, et les professionnels doivent constamment se tenir au courant et savoir choisir très rapidement entre plusieurs solutions.

En utilisant des standards ouverts basés sur des couches de protocoles et sur la liberté de choix des applications, on évite d'être prisonnier d'une solution propriétaire. On peut alors rechercher les meilleures solutions aussi bien matérielles que logicielles.



Les deux modèles décrivant les couches de protocoles sont les modèles DoD et OSI. Le modèle DoD a été spécialement développé pour TCP/IP, tandis que le modèle OSI a été développé plus tard et prend en compte d'autres protocoles. Vous devez bien maîtriser les notions de couches de protocoles, car elles permettent de mieux comprendre l'interopérabilité entre les applications TCP/IP. D'autre part, les modèles font intervenir un vocabulaire largement utilisé dans les présentations des protocoles TCP/IP.

Principes des couches de protocoles

Un réseau TCP/IP comme celui de la Figure 2.1 peut être décomposé en plusieurs éléments :

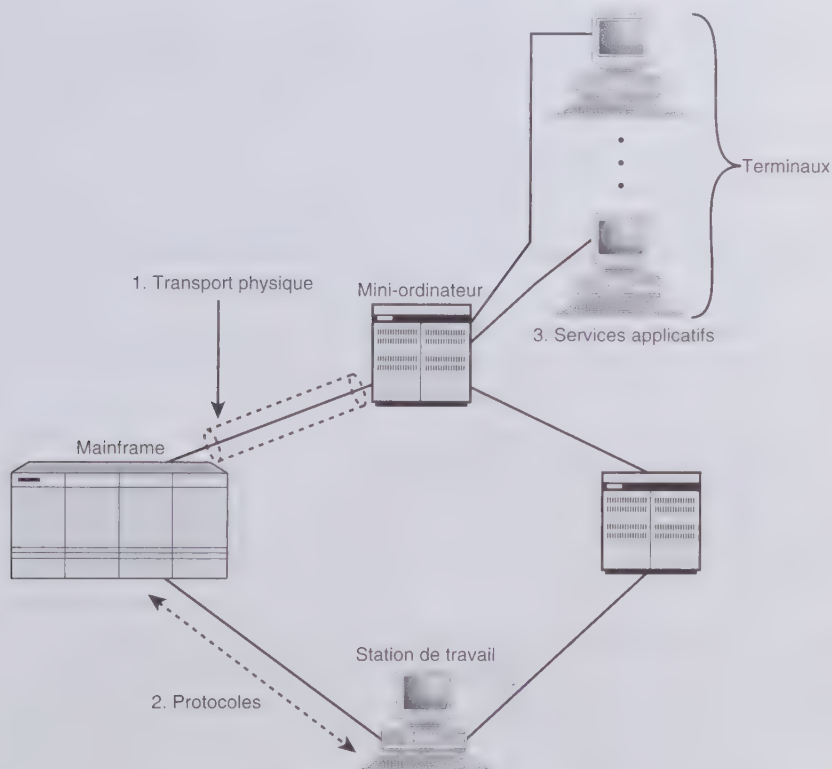
- les connexions physiques ;
- les protocoles ;
- les applications.

Les connexions physiques fournissent le support avec lequel sont transmises les données à travers le réseau. Ces connexions peuvent être des câbles coaxiaux, des paires torsadées (blindées ou non), de la fibre optique, des lignes téléphoniques, des lignes spécialisées, des micro-ondes, de l'infrarouge, des ondes radio ou des liaisons satellite. Vous devez choisir ces connexions en fonction de facteurs spécifiques, par exemple, la bande passante, la facilité d'installation, la maintenance, le coût de l'installation et des équipements.

Les connexions physiques représentent le plus bas niveau des fonctions logiques nécessaires au fonctionnement d'un réseau. Mais, pour que le réseau transmette des données, il faut choisir des règles et des conventions pour que tous les périphériques puissent dialoguer entre eux. Ces règles et ces conventions s'appellent des *protocoles*. Un grand nombre de protocoles remplissent les différentes fonctions nécessaires à la bonne marche d'un réseau.

Au-dessus de ces protocoles se trouvent les applications qui créent les données acheminées sur le réseau par les protocoles. Enfin, les protocoles utilisent les connexions pour transmettre les données.

En analysant ces différentes opérations, on s'aperçoit que les connexions, les protocoles et les applications sont disposés hiérarchiquement. Dans cette hiérarchie, les applications se trouvent au niveau le plus haut et les connexions au niveau le plus bas. Les protocoles agissent comme un "pont" entre les applications et les connexions physiques.

Figure 2.1*Eléments d'un réseau TCP/IP.*

Afin de comprendre cette hiérarchie, il faut disposer d'un modèle qui définit chacune des fonctions nécessaires à la bonne marche du réseau. Deux modèles ont été développés : le modèle OSI, qui est une référence, et le modèle DoD, qui décrit plus particulièrement les protocoles TCP/IP.

Le modèle OSI

Le modèle OSI a été développé en 1978 par l'ISO (*International Organization of Standards*) afin que soit défini un standard utilisé dans le développement de systèmes ouverts. Les réseaux s'appuyant sur les spécifications de l'OSI "parlent le même langage", c'est-à-dire qu'ils utilisent des méthodes de communication semblables pour échanger des données.



Avant les définitions du modèle OSI, les systèmes propriétaires régnaient en maîtres. Une entreprise qui désirait installer un réseau devait choisir entre IBM, DEC, HP, Honeywell et Sperry. Chacune de ces sociétés proposait des architectures propriétaires, et il était pratiquement impossible de faire communiquer les réseaux de ces différents constructeurs.

Dès qu'une entreprise avait choisi son système, elle devenait prisonnière du constructeur. Les mises à jour ou les modifications du système ne pouvaient être effectuées que par le constructeur choisi, ce qui éliminait toute concurrence.

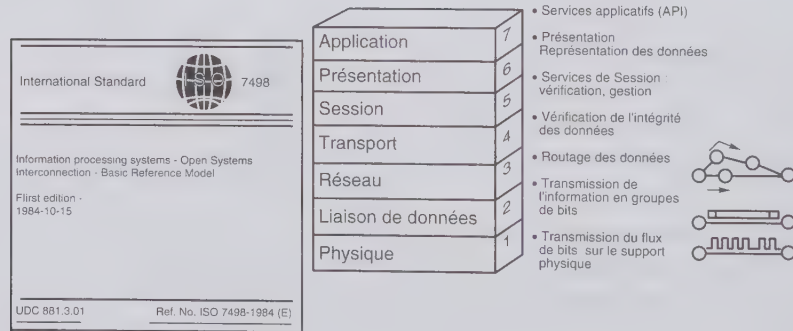
Les utilisateurs d'aujourd'hui peuvent se dire que rien n'a changé dans de nombreux secteurs de l'industrie informatique. Les architectures propriétaires existent toujours. Cependant, le modèle OSI peut vous éclairer sur la façon dont les différents éléments d'un réseau sont reliés entre eux.

• Les couches du modèle OSI

Le modèle OSI est formé de sept couches (voir Figure 2.2).

Figure 2.2

Les couches du modèle OSI.



API = Application Programming Interface

Voici les différentes couches en partant du bas :

- physique ;
- liaison de données ;
- réseau ;
- transport ;
- session ;
- présentation ;
- application.

Cinq principes de base s'appliquent aux différentes couches :

1. Une couche ne peut être créée que quand un niveau différent d'abstraction est nécessaire.
2. Chaque couche doit fournir une fonction bien définie.
3. La fonction de chaque couche doit être choisie de façon à définir internationalement les protocoles standards.
4. Les caractéristiques d'une couche doivent être choisies pour qu'elles réduisent les informations transmises entre les couches.
5. Des fonctions différentes doivent être définies dans des couches différentes, mais il faut éviter d'augmenter le nombre de couches pour que l'architecture ne devienne trop compliquée.

L'application de ces cinq principes crée un modèle idéal, où chaque couche effectue une seule fonction et dépend des services de la couche immédiatement inférieure. De même, chaque couche fournit ses services à la couche immédiatement supérieure. La couche réseau, par exemple, utilise les services de la couche immédiatement inférieure, liaison des données, et fournit ses services à la couche transport, immédiatement supérieure.

Les couches physique et application se situent aux extrémités du modèle OSI. La couche physique n'utilise aucun service d'une autre couche, mais fournit la connectivité physique à la couche supérieure, liaison des données. La couche application utilise les services de la couche présentation et propose ses services à l'utilisateur final.

Les fonctions de chacune des couches sont décrites dans les paragraphes qui suivent.

Couche physique

La *couche physique* transmet les bits à travers un canal de communication. Les bits représentent des enregistrements de base de données ou des fichiers à transférer, mais la couche physique ignore ce que ces bits représentent. Ces bits peuvent être encodés sous forme de 0 et 1 ou sous forme analogique (voir Figure 2.3). La couche physique fait intervenir les interfaces mécaniques et électriques sur le média physique.

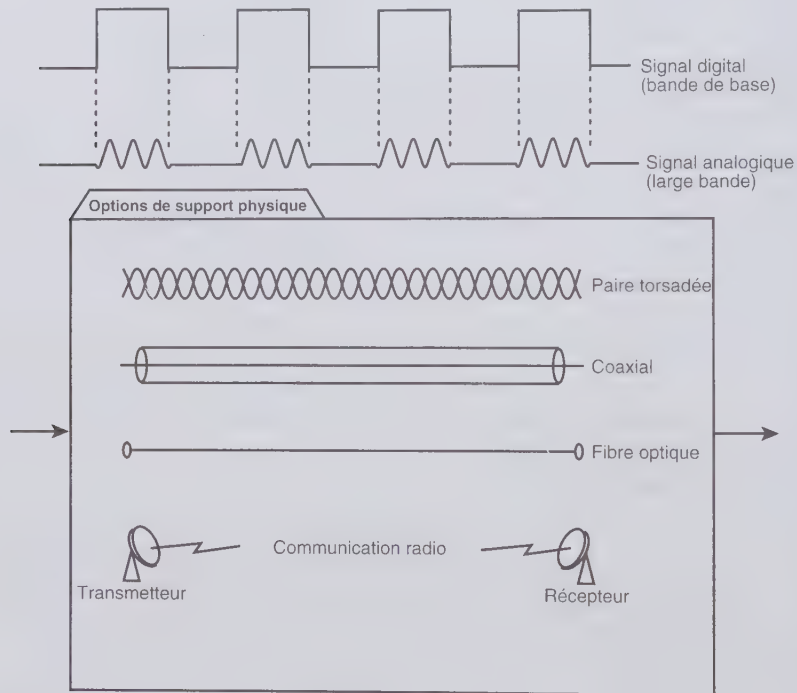
La couche physique n'a aucune connaissance de la structure des données nécessaires pour émettre ou recevoir. La couche physique est responsable de la transmission des bits de données sur le média physique, en utilisant le signal approprié compatible avec les périphériques de communication. De même, la couche physique



reçoit des signaux et les convertit en bits de données qu'elle délivre à la couche liaison des données.

Figure 2.3

*Couche physique
et principaux médias.*



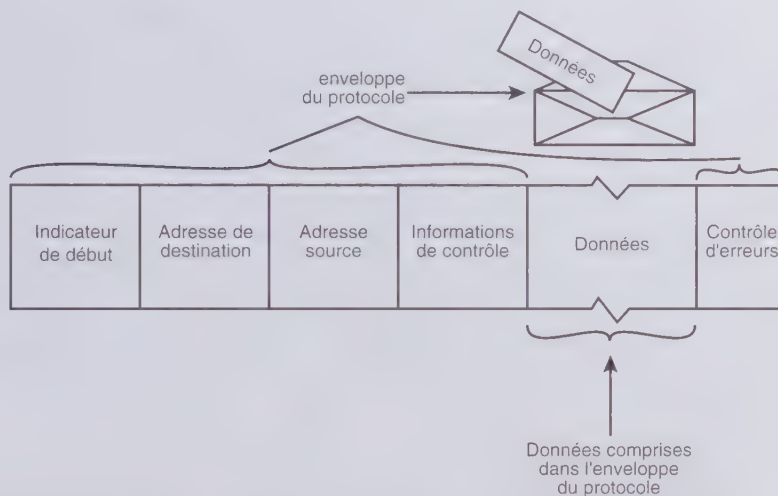
Couche liaison des données

La *couche liaison des données* prend les données de la couche physique et fournit ses services à la couche réseau. Les bits reçus sont groupés en unités logiques appelées *trame*. Dans le contexte d'un réseau, une trame peut être une trame Token Ring ou Ethernet, FDDI, ou une autre type de trame réseau. Pour les liens des réseaux étendus, ces trames peuvent être des trames SLIP, PPP, X.25 ou ATM.

Les bits d'une trame ont une signification spéciale. Le début et la fin d'une trame peuvent être marqués par des bits spéciaux. De plus, les bits de la trame sont répartis en champ adresses, champ de contrôle, champ de données et champ de contrôle d'erreurs. La Figure 2.4 montre une trame de liaison de données. Nous verrons des exemples précis de ce type de trame, plus loin dans ce chapitre.

Les *champs d'adresse* contiennent les adresses source et destination. Le *champ de contrôle* indique les différents types de trames de liaison de données. Le *champ de données* contient les données proprement dites, transmises par la trame. Le *champ de contrôle d'erreurs* détecte les erreurs dans la trame de liaison de données.

Figure 2.4
Trame d'une couche liaison
de données.



La couche liaison de données est la première couche qui gère les erreurs de transmission. En général, le champ de contrôle d'erreurs consiste en un générateur de total de contrôle (*checksum*), utilisé pour détecter les erreurs dans la trame de liaison de données. Dans la plupart des cas, les réseaux modernes utilisent un contrôle de redondance cyclique (CRC). Pour les réseaux locaux, c'est un CRC 32 bits. Pour les réseaux étendus où les liens sont plus lents, on utilise un CRC à 16 bits pour éviter la surcharge des liaisons.

Détailler les calculs du CRC sort du sujet de ce livre, il suffit de savoir que les CRC peuvent détecter :

- toutes les erreurs affectant un seul bit ou deux bits isolés ;
- toutes les erreurs affectant un nombre impair de bits ;
- toutes les erreurs groupées (*burst errors*) de 16 bits ou moins pour les CRC à 16 bits, et de 32 bits ou moins pour les CRC 32 bits.

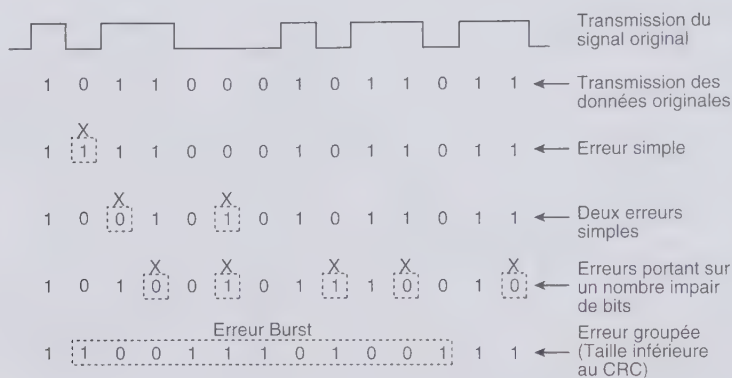
Une erreur groupée est une erreur dans laquelle des séries de bits contigus dans la trame sont en erreur (voir Figure 2.5). La probabilité d'une erreur groupée non détectée parce que plus grande que la taille du CRC est de $1/2^S$, où S est la taille du CRC.

Dans les réseaux TCP/IP, les implémentations de la couche liaison de données comprennent les technologies suivantes : Token Ring, Ethernet, FDDI, relais de trames (*frame relay*), X.25, SLIP, PPP et ATM.



Figure 2.5

Propriétés de détection d'erreurs par CRC.

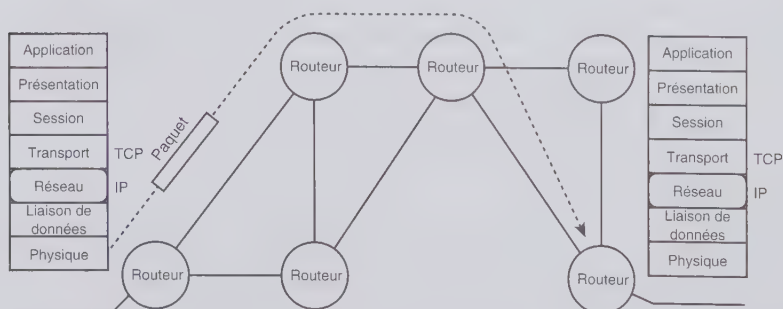


Couche réseau

La couche réseau gère les connexions entre les nœuds du réseau. Un service supplémentaire, fourni par la couche réseau, concerne la façon de router les *paquets* (unité d'information de la couche réseau) entre les nœuds d'un réseau (voir Figure 2.6).

Figure 2.6

Les fonctions de la couche réseau.



- Un routeur est un périphérique qui détermine la prochaine destination du paquet
- Le routeur examine l'en-tête de la couche réseau qui contient un mécanisme d'adressage

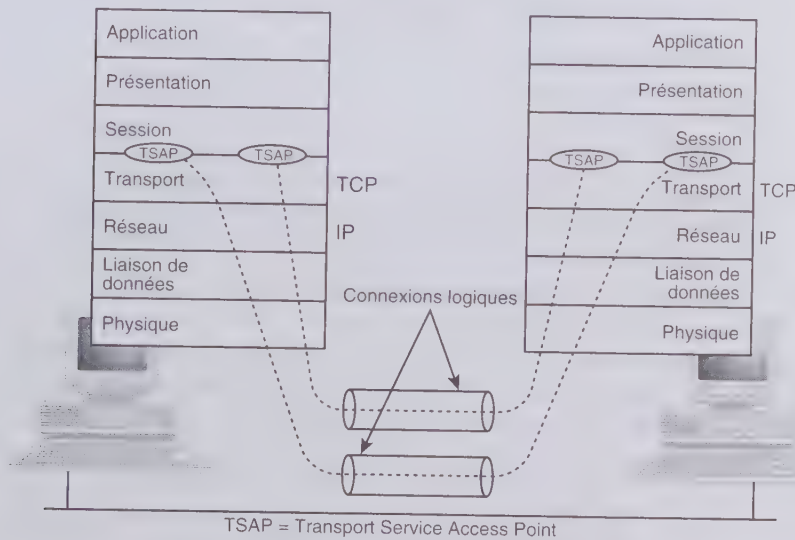
La couche réseau sert aussi à éliminer les congestions et à réguler le flot des données. Cette couche permet aussi à deux réseaux différents d'être interconnectés en implémentant un mécanisme d'adressage uniforme. Token Ring et Ethernet possèdent, par exemple, différents types d'adresses. Pour interconnecter ces réseaux, vous avez besoin d'un mécanisme d'adressage compréhensible par les deux réseaux. Le Chapitre 3 décrit les adresses des réseaux Token Ring et Ethernet. Pour les réseaux TCP/IP, la couche réseau est implémentée en utilisant le protocole IP.

Couche transport

La *couche transport* offre des services supplémentaires par rapport à la couche réseau. Cette couche garantit que les données reçues sont telles qu'elles ont été envoyées. Pour vérifier l'intégrité des données, cette couche se sert des mécanismes de contrôle des couches inférieures.

La couche transport est aussi responsable de la création de plusieurs connexions logiques par *multiplexage* sur la même connexion réseau. Le multiplexage se produit quand plusieurs connexions logiques partagent la même connexion physique. La Figure 2.7 illustre les propriétés de la couche transport.

Figure 2.7
Les fonctions de la couche transport.



La couche transport se trouve au milieu du modèle OSI. Les trois couches inférieures forment le sous-réseau, et les trois couches supérieures sont implémentées par les logiciels réseau. La couche transport est aussi implémentée sur le nœud. Son travail consiste à relier un sous-réseau non fiable à un réseau plus fiable.

Dans les réseaux TCP/IP, la fonction de la couche transport est assurée par le protocole TCP (*Transmission Control Protocol*) et par le protocole UDP (*User Datagram Protocol*).

La couche transport implémente le multiplexage dans lequel plusieurs éléments logiciels partagent la même adresse de la couche réseau. Pour identifier sans



erreur l'élément logiciel dans la couche transport, une forme plus spécifique d'adresse est nécessaire. Ces adresses, appelées *adresses de transport*, sont fournies par une combinaison de l'adresse de la couche réseau et d'un numéro TSAP (*Transport Service Access Point*). Dans les réseaux TCP/IP, l'adresse de transport porte le nom de *numéro de port*.

Couche session

La *couche session* gère les connexions entre les applications coopérantes. Avec cette couche, un utilisateur peut se connecter à un hôte, à travers un réseau où une session est établie pour transférer des fichiers.

La couche session offre les fonctions suivantes :

- contrôle du dialogue ;
- gestion des jetons ;
- gestion de l'activité.

En général, une session permet des communications full duplex, bien que certaines applications se contentent d'une communication half duplex. La couche session peut fournir une ou deux voies de communication (contrôle du dialogue).

Pour certains protocoles, il est essentiel qu'un seul côté lance une opération critique. Pour éviter que les deux côtés lancent la même opération, un mécanisme de contrôle, comme l'utilisation de *jetons*, doit être implémenté. Avec la méthode du jeton, seul le côté qui possède le jeton peut lancer une opération. La détermination du côté qui doit posséder le jeton et son mode de transfert s'appellent la gestion du jeton.

INFO



Le jeton dont il est question ici n'a rien à voir avec le jeton des réseaux Token Ring. La gestion du jeton dans les réseaux Token Ring relève des couches 1 et 2 du modèle OSI, alors que celui dont il est question ici relève du niveau 5.

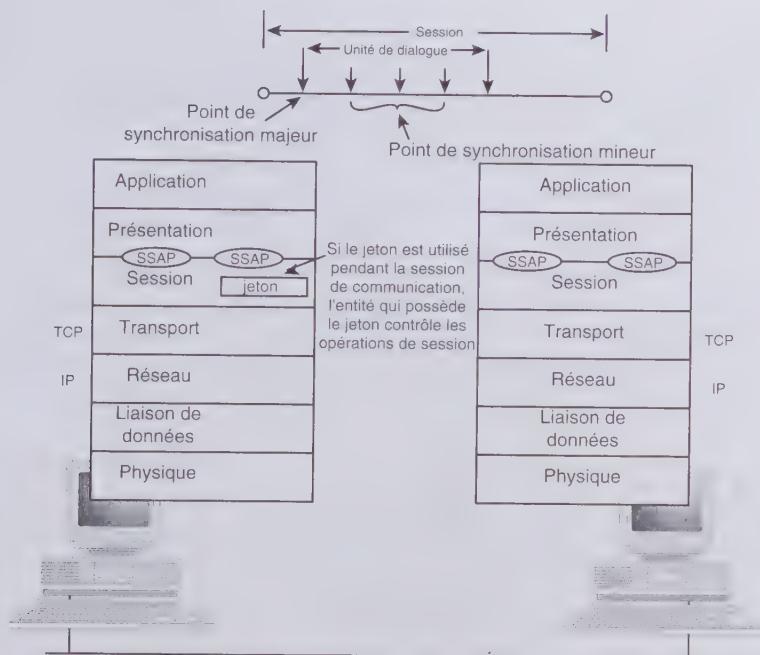
La Figure 2.8 montre quelques-unes des fonctions de la couche session.

Si vous transférez un fichier pendant une heure entre deux machines, et qu'une panne réseau intervienne au bout de trente minutes, vous ne pourrez reprendre le transfert là où il s'était arrêté. Il vous faudra toujours reprendre le transfert à son début. Pour éviter cela, vous pouvez traiter tout le fichier comme une seule activité et insérer des points de vérification dans le flot de données. Ainsi, si une coupure survient, la couche session synchronisera à nouveau le transfert, à partir du dernier point de vérification transmis. Ces points de vérification s'appellent "points de synchronisation". Il en existe deux types :

- majeurs ;
- mineurs.

Figure 2.8

Fonctions de la couche session avec gestion du jeton.



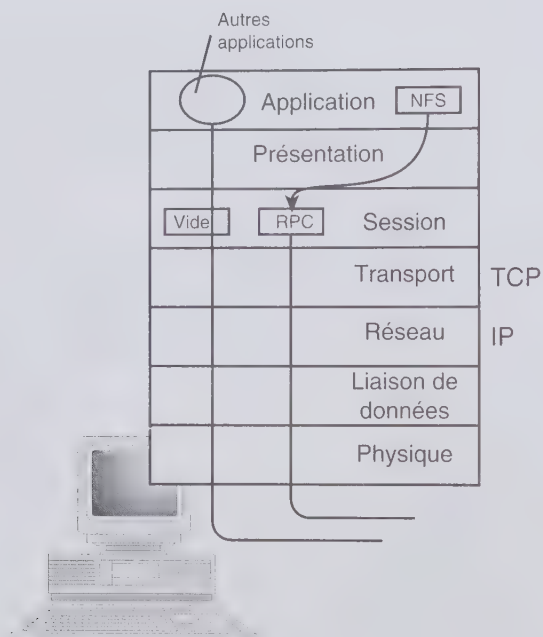
Un point de synchronisation majeur inséré par un des côtés doit recevoir un accusé de réception de la part de l'autre côté, alors qu'un point de synchronisation mineur n'a pas besoin d'être vérifié par un accusé de réception. La session comprise entre deux points majeurs s'appelle une *unité de dialogue*. La gestion de toute l'activité s'appelle une *gestion d'activité*. Une activité consiste en une ou plusieurs unités de dialogue.

Les réseaux TCP/IP ne possèdent pas de couche session, car certaines caractéristiques de cette couche sont fournies par le protocole TCP. Les applications TCP/IP fournissent elles-mêmes certains services. Par exemple, le service NFS (*Network File System*) comporte son propre service de la couche session : le protocole RPC (*Remote Procedure Call*). La Figure 2.9 montre les fonctions de la couche session des réseaux TCP/IP. Certaines applications TCP/IP n'utilisent aucun service de la couche session.

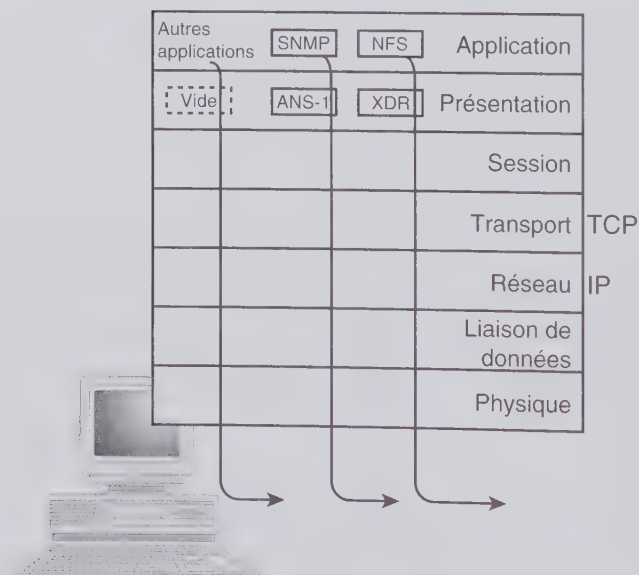


Figure 2.9

Les protocoles
de la couche session
pour les réseaux TCP/IP.

**Figure 2.10**

Les fonctions
de la couche présentation
pour les réseaux TCP/IP.



Couche présentation

Pour que deux systèmes puissent se comprendre, ils doivent utiliser le même système de représentation des données. La *couche présentation* gère cette représentation des données. Il existe plusieurs façons de représenter des données, comme par exemple, le codage ASCII ou EBCDIC pour les fichiers texte. La couche présentation utilise un langage commun compréhensible par tous les nœuds du réseau.

Un exemple de ce langage commun est ASN.1 (*Abstract Syntax Representation Rev 1*). Ce langage est utilisé par SNMP pour coder ses données. Le protocole NFS utilise son propre service : le protocole XDR (*External Data Representation*). La Figure 2.10 montre les fonctions de la couche présentation pour les réseaux TCP/IP. Plusieurs applications TCP/IP n'utilisent pas les services de cette couche.

Couche application

La *couche application* fournit les protocoles et les fonctions nécessaires aux applications utilisatrices qui doivent accomplir des tâches de communication. Voici des exemples de fonctions fournies par la couche application :

- les protocoles pour les services de fichiers distants tels que l'ouverture, la fermeture, la lecture, l'écriture et le partage des fichiers ;
- les services de transfert de fichiers et d'accès aux bases de données distantes ;
- les services de gestion des messages des applications de messagerie ;
- les services des répertoires pour localiser les ressources d'un réseau ;
- la gestion des périphériques ;
- l'exécution de travaux distants.

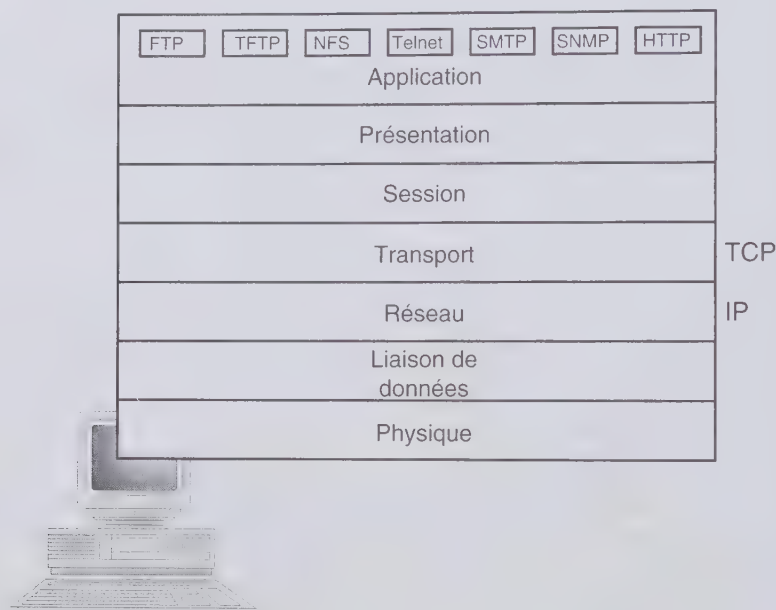
Plusieurs de ces services sont appelés des API (*Application Programming Interface*). Les API consistent en des bibliothèques de programmation qu'un développeur peut utiliser pour écrire des logiciels réseau.

La Figure 2.11 donne quelques exemples de services TCP/IP : FTP (*File Transfer Protocol*), TFTP (*Trivial File Transfer Protocol*), NFS (*Network File System*), TELNET, SMTP (*Simple Mail Transfer Protocol*), SNMP (*Simple Network Management Protocol*), HTTP (*HyperText Transfer Protocol*).



Figure 2.11

Les fonctions
de la couche application
des réseaux TCP/IP.



• Echange transversal des données dans le modèle OSI

La Figure 2.12 montre un exemple d'échange de données dans un réseau. Pour simplifier, seules les communications entre deux ordinateurs A et B sont représentées. Le processus d'envoi à partir de l'ordinateur A transmet des données vers le processus de réception de l'ordinateur B.

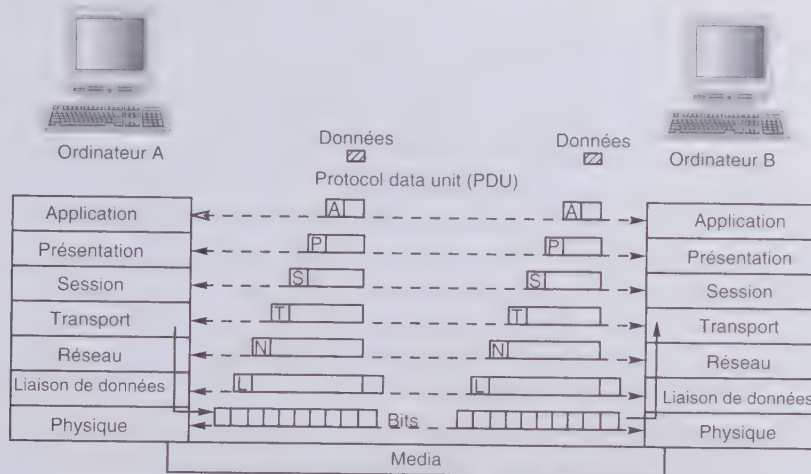
Les deux ordinateurs intervenant dans l'échange de données sont nommés ES (*End Systems*) dans la terminologie OSI. La couche application de chaque ES crée le message et ajoute des bits en en-tête du message. Ces bits sont appelés PCI (*Protocol Control Information*) et représentent des informations sur le processus. En plus, le PCI peut contenir des informations identifiant les entités application (adresses des couches sources et destination de l'application). Le PCI et le message original sont appelés APDU (*Application Protocol Data Unit*).

L'APDU est envoyé à la couche immédiatement inférieure (la couche présentation). Cette couche ajoute son PCI aux données reçues de la couche application, et le message résultant devient le PPDU (*Presentation Protocol Data Unit*). Le message original est encapsulé dans l'APDU et l'APDU est encapsulé à son tour dans le PPDU.

Chaque couche ajoute ainsi son information d'en-tête au message reçu de la couche supérieure. C'est comme si chaque message était placé dans une enveloppe. L'enveloppe avec son contenu forme le PDU de la couche.

Figure 2.12

Echange transversal de données
dans le modèle OSI.



- Les données transitent de haut en bas au niveau de l'extrémité locale
- Informations de contrôle de protocole (en-tête/en queue) utilisées comme enveloppe sur chaque couche
- Les données transitent de bas en haut au niveau de l'extrémité distante
- Informations de contrôle de protocole (en-tête/en queue) supprimées à mesure que les données remontent les couches

Le processus continue ainsi jusqu'à la couche physique. Au niveau de la couche liaison des données, un champ de vérification (le *trailer field*) est ajouté aux données. Ce champ est en général un CRC. Cette vérification est générée par un mécanisme matériel de la couche liaison des données (cartes réseau). Il est ajouté à la fin, car le mécanisme matériel crée le CRC au fur et à mesure que les données sont envoyées en série sur la ligne.

Au niveau de la couche physique, l'information d'en-tête prend la forme d'une indication informant le destinataire de l'arrivée d'un paquet. Sur les réseaux Ethernet, cette information est représentée par un préambule de 56 bits, utilisé par le destinataire pour se synchroniser.

A l'autre extrémité (ordinateur B), la couche physique reçoit les bits, supprime les bits de synchronisation et envoie le reste à la couche liaison des données. Cette couche regroupe les bits dans une trame et vérifie si le CRC est valide. Ensuite, elle supprime le PCI et envoie le reste NPDU (*Network PDU*) à la couche réseau.

A chaque extrémité, d'une couche, le paquet est traité d'après les informations PCI de la couche. Le PCI est supprimé, puis les données sont envoyées à la couche



supérieure. Ce processus continue jusqu'à la couche application qui récupère les données originales.

Si l'on examine les couches au niveau local et au niveau distant, on s'aperçoit que chaque couche communique avec la couche homologue à l'autre extrémité. Par exemple, les couches transport aux deux extrémités communiquent entre elles. Pour que cette communication ait lieu, les couches utilisent l'infrastructure fournie par les couches inférieures (réseau, liaison de données et physique).

Les éléments actifs dans chaque couche OSI sont appelés des entités protocoles. Ces entités protocoles implémentent les services dans chaque couche OSI et communiquent avec les entités protocoles de la couche correspondante de l'autre système. Une entité protocole est constituée d'un ensemble de logiciels ou de matériels (circuit électronique, composant, etc.).

Normalement, les couches physique et liaison de données sont implémentées dans le matériel, et les entités protocoles des couches réseau et supérieures sont implémentées sous forme logicielle. Dans les périphériques comme les routeurs, un hybride de logiciel et de matériel appelé *firmware* est utilisé pour implémenter les entités protocoles.

Le modèle OSI fournit une vue conceptuelle des communications, facile à appréhender. En pratique, les couches ne sont pas définies exactement comme dans le modèle OSI. Une implémentation particulière peut ne pas avoir de séparation entre les couches présentation et application. Plusieurs fonctions de la couche présentation, comme l'encodage et le décodage des données, peuvent être effectuées par la couche application, par appel de fonctions spécifiques d'encodage et de décodage.

Le modèle DoD

Le modèle OSI a été créé en 1979, bien que le concept de couches existât bien avant d'être normalisé par l'ISO. Avec les protocoles TCP/IP, vous avez un exemple de protocoles utilisés bien avant le modèle OSI. Comme ces protocoles ont été historiquement créés à la demande du ministère de la Défense des Etats-Unis, les couches TCP/IP portent le nom de modèle DoD.

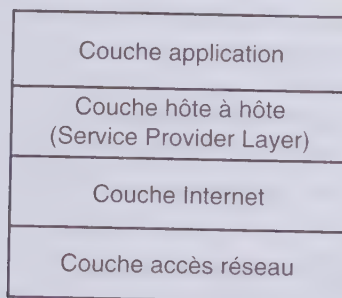
La Figure 2.13 décrit le modèle DoD, formé de quatre couches.

Voici les différentes couches, en partant du bas vers le haut :

- la couche accès réseau ;
- la couche Internet ;
- la couche transport hôte à hôte ;
- la couche application.

Figure 2.13

Le modèle DoD.



• La couche accès réseau

La couche la plus basse représente la connexion physique avec les câbles, les circuits d'interface électrique (*transceivers*), les cartes réseau, les protocoles d'accès au réseau (CSMA/CD pour les réseaux Ethernet et le jeton pour les réseaux Token Ring). La couche accès réseau est utilisée par la couche Internet.

• La couche Internet

La couche Internet doit fournir une adresse logique pour l'interface physique. L'implémentation du modèle DoD de la couche Internet est IP (*Internet Protocol*). Cette couche fournit un mappage entre l'adresse logique et l'adresse physique fournie par la couche accès réseau grâce aux protocoles ARP (*Address Resolution Protocol*) et RARP (*Reverse Address Resolution Protocol*). Les incidents, les diagnostics et les conditions particulières associées au protocole IP relèvent du protocole ICMP (*Internet Control Message Protocol*), qui opère aussi au niveau de la couche Internet.

La couche Internet est aussi responsable du routage des paquets entre les hôtes. Cette couche est utilisée par les couches plus hautes du modèle DoD.

• La couche transport hôte à hôte

La couche hôte à hôte définit les connexions entre deux hôtes sur le réseau. Le modèle DoD comprend deux protocoles hôte à hôte : TCP (*Transmission Control Protocol*) et UDP (*User Datagram Protocol*). Le protocole TCP est responsable du service de transmission fiable de données avec détection et correction d'erreurs.



TCP permet aussi les connexions simultanées. Plusieurs connexions TCP peuvent être établies sur un hôte, et les données sont envoyées simultanément. TCP permet des connexions full duplex, ce qui signifie que les données peuvent être envoyées et reçues sur une seule connexion.

Le protocole UDP est un protocole peu fiable et peut être utilisé par des applications qui n'exigent pas la fiabilité de TCP.

• La couche application

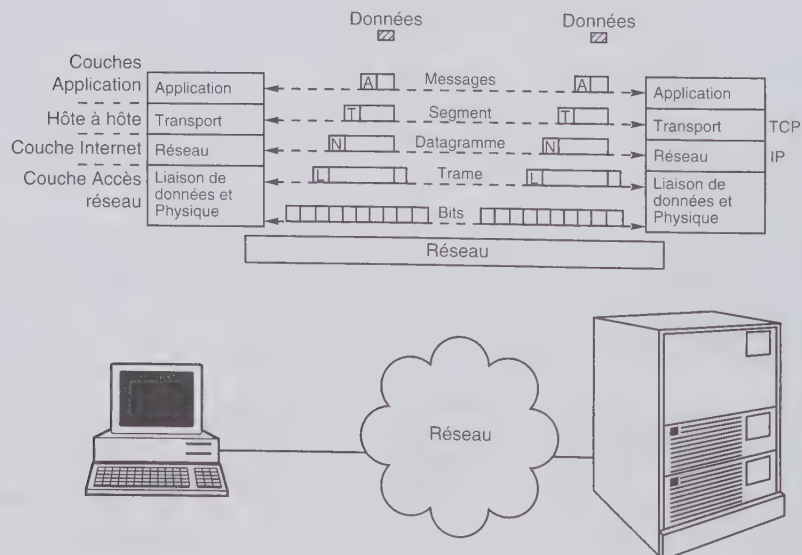
La couche application permet aux applications d'utiliser les protocoles de la couche hôte à hôte (TCP et UDP). Parmi ces applications, on trouve : FTP (*File Transfer Protocol*), TELNET (*Terminal Emulation*), SMTP (*Simple Mail Transfer Protocol*) et SNMP (*Simple Network Management Protocol*). La couche application assure l'interface des applications utilisatrices avec la pile de protocoles TCP/IP.

• Transfert des données sur un réseau TCP/IP d'après le modèle DoD.

La Figure 2.14 montre la communication entre deux hôtes d'après le modèle DoD.

Figure 2.14

Communication à travers les couches de protocoles du modèle DoD.



Les données transmises par un hôte sont encapsulées par le protocole de la couche application. Les données de cette couche sont ensuite encapsulées par la couche hôte à hôte (TCP ou UDP). Les données de la couche hôte à hôte sont à nouveau encapsulées par la couche Internet. Enfin, les données de la couche Internet sont encapsulées par le protocole de la couche accès réseau.

Le processus inverse a lieu dès que les données sont reçues par l'hôte distant. Chaque couche transmet à la couche supérieure les données débarrassées des entêtes des couches précédentes.

• Comparaison des modèles OSI et DoD

La Figure 2.15 établit un parallèle entre les modèles OSI et DoD.

Figure 2.15

Comparaison des modèles OSI et DoD.

Modèle de référence OSI			
7	Couche Application		
6	Couche Présentation		
5	Couche Session		
4	Couche Transport		
3	Couche Réseau		
2	Couche Liaison de données		
1	Couche Physique		
		Couche Application	4
		Couche Hôte à hôte (Service Provider Layer)	3
		Couche Internet	2
		Couche Accès réseau	1

Aux deux couches physique et liaison de données du modèle OSI correspond la couche accès réseau du modèle DoD.

La couche réseau du modèle OSI correspond à la couche Internet du modèle DoD.

La couche transport du modèle OSI correspond à la couche hôte à hôte du modèle DoD.

Les trois couches les plus hautes du modèle OSI (session, présentation et application) correspondent à la seule couche application du modèle DoD.



Le Tableau 2.1 résume ces comparaisons.

Tableau 2.1 : Comparaison des modèles DoD et OSI

Couches DoD	Numéro des couches OSI	Couches OSI
Accès réseau	1	Physique
	2	Liaison de données
Internet	3	Réseau
Hôte à hôte	4	Transport
Application	5	Session
	6	Présentation
	7	Application

La terminologie utilisée pour décrire les données au niveau de chaque couche diffère dans les deux modèles. Dans le modèle OSI, l'expression PDU (*Protocol Data Unit*) est employée pour décrire les données d'une couche. Dans le modèle DoD, le terme *message* est utilisé au niveau de la couche application ; le terme *segment*, au niveau de la couche hôte à hôte ; le terme *datagramme*, au niveau de la couche Internet ; et le terme *trame*, au niveau de la couche accès réseau.

Hiérarchie de l'implémentation de TCP/IP

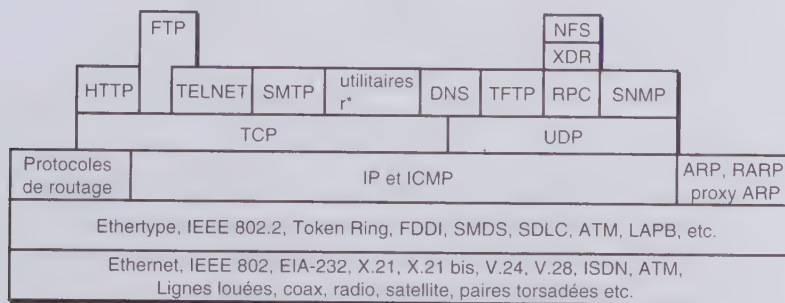
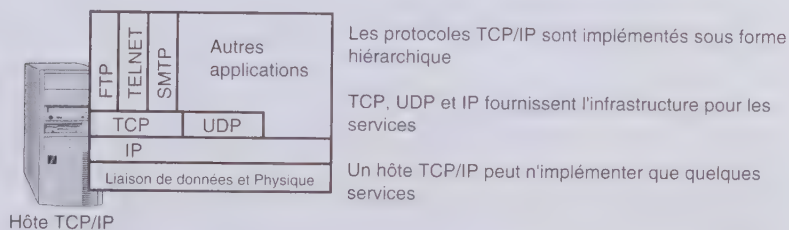
La suite de protocoles TCP/IP comprend un ensemble performant de services qui peuvent utiliser une grande variété de technologies réseau comme les réseaux étendus, les réseaux locaux, les ondes radio, les satellites ou les lignes Numéris. Les modules du protocole TCP/IP et les relations entre ces modules forment la *hiérarchie de l'implémentation TCP/IP*.

La Figure 2.16 montre une liste de services TCP/IP ainsi que la plupart des réseaux qui supportent TCP/IP. Les services sont présentés en relation avec le modèle OSI. Une implémentation supporte seulement quelques services, ou d'autres protocoles comme les protocoles ARP, RARP, proxy ARP et routage. L'hôte TCP/IP dans la Figure 2.16, par exemple, ne supporte que les services FTP, TELNET et SMTP.

Les services décrits dans la Figure 2.16 sont appelés "services TCP/IP" car, historiquement, ils sont rattachés au protocole TCP/IP, mais cela ne signifie pas qu'ils ne peuvent fonctionner avec d'autres piles de protocoles. Par exemple, le protocole SNMP est historiquement associé au protocole TCP/IP, mais peut aussi tourner sur

une pile de protocoles IPX (RFC 1420), une pile AppleTalk (RFC 1419) ou une pile OSI (RFC 1418). De même, les services non fondés sur TCP/IP peuvent aussi tourner sous TCP/IP. C'est le cas, par exemple, des services SMB (*Server Message Block*) et NetBIOS, qui tournent normalement sous le protocole NetBEUI (RFC 1001 et 1002), et de X.500, qui utilise les protocoles OSI. L'ISODE (*ISO Development Environment*) fournit les services X.500 sur TCP/IP.

Figure 2.16
Hiérarchie
de l'implémentation TCP/IP.



Le multiplexage/démultiplexage, concept important de TCP/IP, permet à plusieurs protocoles des couches hautes d'utiliser un protocole commun d'une couche basse. Ce concept est présenté ci-dessous.

• Multiplexage et démultiplexage de protocoles

La Figure 2.17 montre la liaison entre deux hôtes. Les services TCP/IP comme FTP, TELNET, TFTP et SMTP tournent sur chaque hôte.

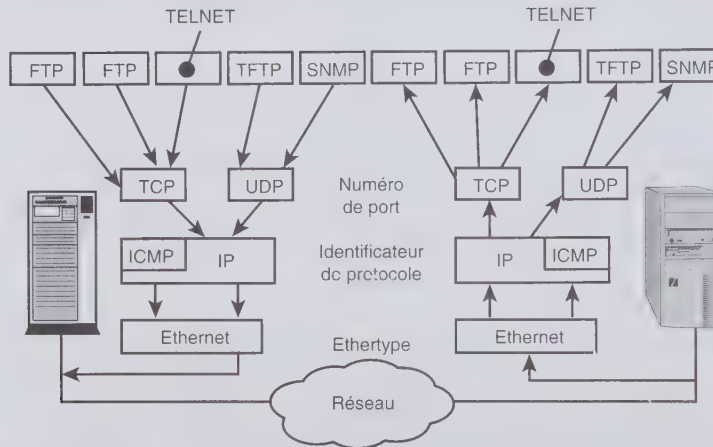


Figure 2.17

Multiplexage et
démultiplexage
de protocoles.

Comment les logiciels distinguent-ils les applications des protocoles dans une couche ?

- Multiplexage /démultiplexage de protocoles
 - - Champ Ethertype
 - - Identificateur de protocole
 - - Numéros de port

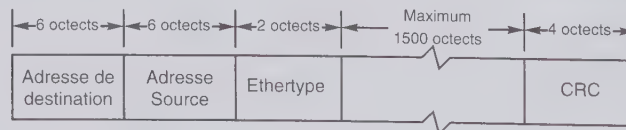


Plusieurs sessions existent entre les hôtes qui utilisent FTP. Une question vient à l'esprit : comment les logiciels sur chaque hôte arrivent-ils à différencier les divers protocoles ou applications au niveau d'une couche donnée ?

Le réseau Ethernet de la Figure 2.17 montre un environnement supportant IP et ICMP et, potentiellement, d'autres protocoles comme IPX et AppleTalk. Comment savoir qu'un paquet arrivant du réseau est destiné à IP ou à ICMP ? Pour résoudre ce problème, la trame Ethernet contient un champ de deux octets : EtherType (voir Figure 2.18). Le Tableau 2.2 répertorie les valeurs du champ EtherType utilisées par les protocoles de la couche réseau. Vous pouvez constater que le protocole IPX possède les deux valeurs 8137 et 8138 pour le champ EtherType.

Figure 2.18

Champ EtherType
d'une trame Ethernet.



Le champ EtherType de la couche liaison des données autorise le multiplexage de plusieurs protocoles réseau au niveau de la source et le démultiplexage au niveau de la destination (voir Figure 2.19).

Figure 2.19

Multiplexage et démultiplexage par le champ EtherType.

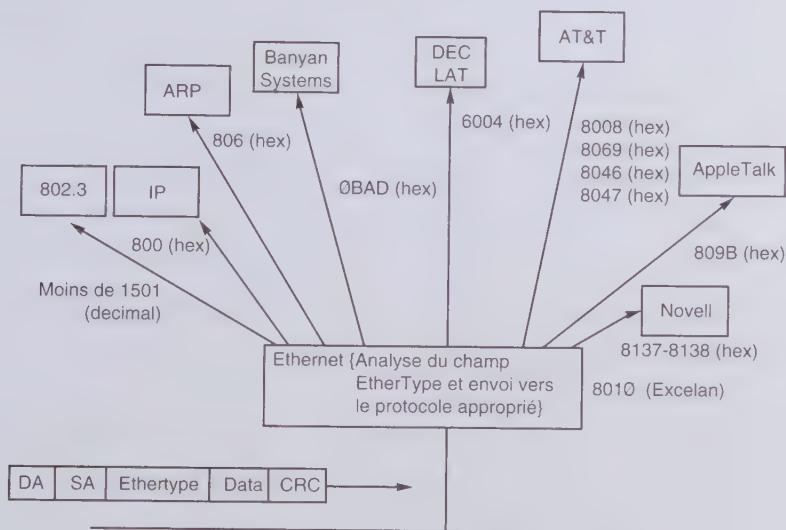


Tableau 2.2 : Valeurs du champ EtherType pour la couche liaison des données

Valeur EtherType (décimale)	Valeur EtherType (hexadécimale)	Affectation du champ EtherType
0-1500	0-05DC	IEEE 802.3
1536	0600	XEROX NS IDP
2048	0800	DoD IP
2049	0801	X.75 Internet
2050	0802	NBS Internet
2051	0803	ECMA Internet
2052	0804	Chaosnet
2053	0805	X.25 niveau 3



Tableau 2.2 : Valeurs du champ EtherType pour la couche liaison des données (suite)

Valeur EtherType (décimale)	Valeur EtherType (hexadécimale)	Affectation du champ EtherType
2054	0806	ARP (<i>Address Resolution Protocol</i>)
2055	0807	Compatibilité XNS
2076	081C	Privé
2184-2186	0888-088A	Xyplex
2304	0900	Ungermann-Bass Net Debugger
2560	0A00	Xerox IEEE 802.3 PUP
2561	0A01	PUP Address Translation
2989	0BAD	Banyan Systems
4096	1000	Berkeley Trailer Negotiation
4097-4101	1001-100F	Berkeley Trailer Encapsulation/IP
5632	1600	Valid Systems
16962	4242	PCS Basic Block Protocol
21000	5208	BBN Simnet
24576	6000	DEC non assigné (expérimental)
24577	6001	DEC MOP Dump/Load
24578	6002	DEC MOP Remote Console
24579	6003	DEC DECNET Phase IV Route
24580	6004	DEC LAT
24581	6005	DEC Diagnostic Protocol
24582	6006	DEC Customer Protocol
24583	6007	DEC LAVC, SCA
24584-24585	6008-6009	DEC non assigné
24586-24590	6010-6014	3COM Corporation
28672	7000	Ungermann-Bass download
28674	7002	Ungermann-Bass dia/loop
28704-28713	7020-7029	LRT
28720	7030	Proteon

Tableau 2.2 : Valeurs du champ EtherType pour la couche liaison des données (suite)

Valeur EtherType (décimale)	Valeur EtherType (hexadécimale)	Affectation du champ EtherType
28724	7034	Cabletron
32771	8003	Cronus VLN
32772	8004	Cronus Direct
32773	8005	HP Probe
32774	8006	Nestar
32776	8008	AT&T
32784	8010	Excelan (Novell)
32787	8013	SGI diagnostics
32788	8014	SGI network games
32789	8015	SGI réservé
32790	8016	SGI bounce server
32793	8019	Apollo Computers
32815	802E	Tymshare
32816	802F	Tigan, Inc.
32821	8035	Reverse ARP
32822	8036	Aeonic Systems
32824	8038	DEC LANBridge
32825-32828	8039-803C	DEC non assigné
32829	803D	DEC Ethernet Encryption
32831	803F	DEC LAN Traffic Monitor
32832-32834	8040-8042	DEC non assigné
32836	8044	Planning Research Corp.
32838-32839	8046-8047	AT&T
32841	8049	ExperData
32859	805B	Stanford V Kernel exp.



Tableau 2.2 : Valeurs du champ EtherType pour la couche liaison des données (suite)

Valeur EtherType (décimale)	Valeur EtherType (hexadécimale)	Affectation du champ EtherType
32860	805C	Stanford V Kernel prod.
32864	805D	Evans & Sutherland
32866	8062	Counterpoint Computers
32869-32870	8065-8066	University of Massachusetts at Amherst
32871	8067	Veeco Integrated Auto
32872	8068	General Dynamics
32873	8069	AT&T
32874	806A	Autophon
32876	806C	ComDesign
32877	806D	Computgraphic Corp.
32878-32887	806E-8077	Landmark Graphics Corp.
32890	807A	Matra
32891	807B	Dansk Data Elektronik
32892	807C	Merit Internodal
32893-32895	807D-807F	Vitalink Communications
32896	8080	Vitalink TransLAN III
32897-32899	8081-8083	Counterpoint Computers
32923	809B	AppleTalk
32924-32926	809C-809E	Datability
32927	809F	Spider Systems Limited
32931	80A3	Nixdorf Computers
32932-32947	80A4-80B3	Siemens Gammasonics, Inc.
32960-32963	80C0-80C3	DCA Data Exchange Cluster
32966	80C6	Pacer Software
32967	80C7	Applitek Corporation
32968-32972	80C8-80CC	Intergraph Corporation
32973-32975	80CD-80CE	Harris Corporation

Tableau 2.2 : Valeurs du champ EtherType pour la couche liaison des données (suite)

Valeur EtherType (décimale)	Valeur EtherType (hexadécimale)	Affectation du champ EtherType
32975-32978	80CF-80D2	Taylor Instruments
32979-32980	8CD3-80D4	Rosemount Corporation
32981	80D5	IBM SNA Service on Ethernet
32989	80DD	Varian Associates
32990-32991	80DE-80DF	Integrated Solution TRFS
32992-32995	80E0-80E3	Allen-Bradley
32996-3308	80E4-80F0	Datability
33010	80F2	Retix
33011	80F3	AppleTalk AARP (Kinetics)
33012-33013	80F4-80F5	Kinetics
33015	80F7	Apollo Computer
33023	80FF	Wellfleet Communications
33031-33033	8107-8109	Privé
33072	8130	Waterloo Microsystems
33073	8131	VG Laboratory Systems
33079-33080	8137-8138	Novell, Inc.
33081	8139	KTI
33100	814C	SNMP Research
36864	9000	Loopback
36865	9001	3COM (Bridge) XNS System Management
36866	9002	3COM (Bridge) TCP Sys
36867	9003	3COM (Bridge) loopdetect
65280	FF00	BBN VITAL-LanBridge cache



Quand la couche IP reçoit un paquet Ethernet, elle doit distinguer les paquets pour TCP ou UDP grâce à un champ d'identification de protocole de huit bits situé dans le paquet. Le Tableau 2.3 répertorie quelques valeurs de ce champ. Plusieurs implémentations de TCP/IP stockent ces valeurs dans un fichier spécial, appelé fichier *protocoles*.

Tableau 2.3 : Valeurs d'identification de protocoles de la couche réseau

Identificateur du protocole	Protocole de la couche suivante dans le paquet IP
0	Réservé
1	Internet Control Message Protocol (ICMP)
2	Internet Group Management Protocol (IGMP)
4	IP dans l'encapsulation IP
5	Flux IP
6	Transmission Control Protocol (TCP)
8	Exterior Gateway Protocol (EGP)
9	Interior gateway protocol privé (par exemple, IGP de CISCO)
11	Network Voice Protocol (NVP-II)
12	Parc Universal Protocol (PUP)
16	Chaos Protocol
17	User Datagram Protocol (UDP)
21	Packet Radio Measurement (PRM)
22	XEROX NS IDP (XNS-IDP)
29	ISO Transport Protocol Class 4 (ISO-TP4)
30	Bulk Transfer Protocol (NETBLT)
36	Express Transport Protocol (XTP)
37	Datagram Delivery Protocol (DDP)
75	Packet Video Protocol (PVP)
80	ISO Internet Protocol (ISO-IP)
83	VINES

Quand un module TCP ou UDP reçoit un paquet de la couche IP, il doit distinguer l'application qui traitera ce paquet (FTP, TELNET, etc.). Chaque paquet contient un champ de seize bits indiquant le numéro du port correspondant à l'application.

Les Tableaux 2.4 et 2.5 recensent les principales valeurs des numéros de ports pour TCP et UDP.

Notez que certaines applications répertoriées dans les deux tableaux possèdent le même numéro de port. Ces applications sont disponibles aussi bien en TCP qu'en UDP. Plusieurs implémentations de TCP/IP stockent ces valeurs dans le fichier services situé dans le répertoire /etc des hôtes Unix.

Tableau 2.4 : Numéros des ports utilisés par TCP

Numéro du port TCP	Service
0	Réservé
1	TCP Port Service Multiplexor
2	Utilitaire de gestion
3	Processus de compression
5	Travail distant
7	Echo
9	Discard
11	Utilisateurs actifs (systat)
13	Daytime
17	QUOTD (Quote of the Day)
20	Port données FTP
21	Port contrôle FTP
23	Telnet
25	SMTP
35	Serveur d'impression privé
37	Time
39	Resource Location Protocol
42	Nameserver
43	Nickname
49	Login Host Protocol
52	XNS Time Protocol



Tableau 2.4 : Numéros des ports utilisés par TCP (*suite*)

Numéro du port TCP	Service
53	Domain Name Server
54	XNS clearing house
66	Oracle SQL*NET
67	Bootstrap Protocol Server
68	Bootstrap Protocol Client
70	Gopher protocol
79	Finger protocol
80	World Wide Web HTTP
88	Kerberos
94	Trivoli Object Dispatcher
95	SUPDUP
102	ISO-TSAP
107	Remote Telnet Service
108	SNA Gateway Access Server
110	Post Office Protocol – Version 3 (POP3)
111	Sun Remote Procedure Call
119	Network News Transfer Protocol (NNTP)
123	Network Time Protocol (NTP)
134	INGRES-NET Service
137	NETBIOS Naming Service
138	NETBIOS Datagram Service
139	NETBIOS Session Service
142	Britton-Lee IDM
191	Prospero
194	Internet Relay Chat Protocol (IRC)
201	AppleTalk Routing Maintenance
202	AppleTalk Name Binding
213	IPX

Tableau 2.4 : Numéros des ports utilisés par TCP (suite)

Numéro du port TCP	Service
215	Insignia (Soft PC)
217	dBASE UNIX
372	UNIX Listserv
519	Unixtime
525	Time Server
533	Netwall
556	RFS server
565	Who Am I
749	Kerberos Administration
767	Phone
1025	Network Blackjack
1352	Lotus Notes
7000-7009	Utilisé par Andrew File System
17007	ISODE Directory User Agent

Tableau 2.5 : Numéros des ports utilisés par UDP

Numéro du port UDP	Service
0	Réservé
2	Utilitaires de gestion
3	Processus de compression
5	Entrée de travaux distants
7	Echo
9	Discard
11	Utilisateurs actifs (sysstat)
13	Daytime
17	QUOTD (Quote of the Day)



Tableau 2.5 : Numéros des ports utilisés par UDP (suite)

Numéro du port UDP	Service
35	Serveur d'impression privé
37	Time
39	Resource Location Protocol
42	Nameserver
43	Nickname
49	Login Host Protocol
52	XNS Time Protocol
53	Domain Name System Server
54	XNS Clearing House
66	Oracle SQL*NET
67	Bootstrap Protocol Server
68	Bootstrap Protocol Client
69	Trivial File Transfer Protocol (TFTP)
70	Gopher Protocol
79	Finger Protocol
80	World Wide Web HTTP
88	Kerberos
94	Trivoli Object Dispatcher
95	SUPDUP
102	ISO-TSAP
107	Remote Telnet Service
108	SNA Gateway Access Server
110	Post Office Protocol – Version 3 (POP3)
111	Sun Remote Procedure Call
119	Network News Transfer Protocol (NNTP)
123	Network Time Protocol (NTP)
134	INGRES-NET Service
137	NETBIOS Naming Service

Tableau 2.5 : Numéros des ports utilisés par UDP (suite)

Numéro du port UDP	Service
138	NETBIOS Datagram Service
139	NETBIOS Session Service
142	Britton-Lee IDM
161	SNMP
162	SNMP Traps
191	Prospero
194	Internet Relay Chat Protocol (IRC)
201	AppleTalk Routing Maintenance
202	AppleTalk Name Binding
213	IPX (utilisé pour le tunneling IP)
215	Insignia (Soft PC)
217	dBASE UNIX
372	UNIX Listserv
513	Renseigne une base de données sur les logins d'un réseau local et la charge moyenne de chaque machine
519	Unixtime
525	Time Server
533	Netwall
556	RFS server
565	Who Am I
749	Kerberos Administration
767	Phone
1025	Network Blackjack
1352	Lotus Notes
7000-7009	Utilisé par Andrew File System
17007	ISODE Directory User Agent



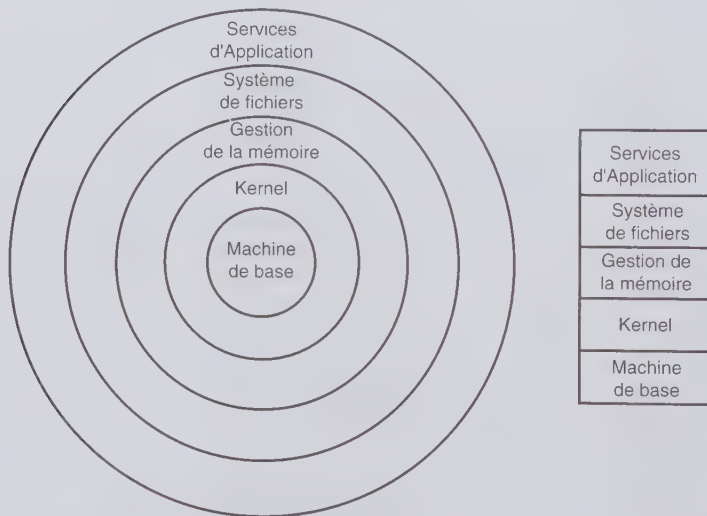
• Implémentation de TCP/IP et système d'exploitation

Les performances de TCP/IP, sa configuration et sa facilité de maintenance dépendent du système d'exploitation sur lequel il est implémenté. Afin de souligner l'interaction entre TCP/IP et les applications, les constructeurs essaient de montrer que le système d'exploitation fait partie du modèle OSI. Malheureusement, le système d'exploitation n'en fait pas exactement partie, car le modèle OSI est défini pour des fonctions de communication. Le système d'exploitation possède son propre modèle en couches. Un tel modèle est représenté dans la Figure 2.20.

En comparant cette figure à la Figure 2.2, on s'aperçoit que le système d'exploitation et le modèle OSI sont distincts. Un système d'exploitation est un programme qui gère des ressources. Un système de communication réseau est une des ressources que doit gérer le système d'exploitation.

Figure 2.20

Les couches d'un système d'exploitation.



En théorie, le système d'exploitation n'est pas nécessaire pour faire tourner TCP/IP : le protocole TCP/IP peut être implémenté en ROM. Toutefois, la plupart des implémentations commerciales de TCP/IP interagissent avec le système d'exploitation.

Ces interactions entre TCP/IP et le système d'exploitation peuvent être classées de la façon suivante :

- comme faisant partie du noyau (*kernel*) du système d'exploitation ;
- comme un pilote de périphérique ;
- comme un processus d'application.

Dans un système d'exploitation comme Unix ou NetWare, TCP/IP est implémenté dans le kernel du système. Ce type d'implémentation TCP/IP permet d'obtenir des fonctions de communication rapides.

Les systèmes d'exploitation dont l'implémentation de TCP/IP est faite sous forme de pilote de périphérique comprennent Windows NT et Windows 95, VMS, OS/2, Windows et MS-DOS. Dans Windows, qui tourne au-dessus de MS-DOS, le pilote de périphérique VxD virtuel implémente la pile TCP/IP sous forme d'un pilote 32 bits qui utilise la mémoire étendue et évite ainsi l'utilisation de la mémoire de base des processeurs Intel.

Les systèmes d'exploitation qui implémentent TCP/IP sous forme de processus comprennent les systèmes d'exploitation des mainframes IBM MVS et VM, Windows et MS-DOS. Certaines implémentations de TCP/IP dans Windows et MS-DOS utilisent un programme TSR et des bibliothèques DLL.

Récapitulatif

Les applications TCP/IP sont construites de façon hiérarchique, à partir d'une série de protocoles disposés en couches. Ces composants sont appelés couches de protocoles. Il existe deux modèles décrivant les couches de protocoles : le modèle OSI et le modèle DoD. Ces deux modèles permettent de mieux comprendre l'interaction des différents protocoles. Ces modèles en couches permettent aussi de définir le vocabulaire utilisé dans la présentation des protocoles TCP/IP.

Testez vos connaissances

1. Nommez les modèles utilisés pour décrire les protocoles TCP/IP.
2. Quels sont les trois principaux éléments d'un réseau d'ordinateur ?
3. Nommez les couches des modèles OSI et DoD.
4. Quels sont les cinq principes utilisés pour déterminer les couches du modèle OSI ?
5. Quelle couche du modèle OSI correspond à la détection d'erreurs CRC ?
6. A quelle couche OSI correspond le protocole IP ?
7. A quelle couche OSI correspond le protocole TCP ?



8. Quelle couche OSI est responsable de la remise et de l'intégrité des données ?
9. Quelle est la couche OSI qui définit la représentation des données ?
10. Quelles sont les différences entre les modèles OSI et DoD ?
11. Donnez des exemples de deux implémentations des protocoles de la couche présentation, utilisées par les applications TCP/IP. Nommez les applications TCP/IP qui se servent des protocoles de la couche présentation.
12. Donnez une description rapide des protocoles de multiplexage/démultiplexage dans les réseaux TCP/IP.
13. Quelles sont les informations contenues dans la trame Ethernet et utilisées pour les protocoles de multiplexage/démultiplexage ?
14. A partir du Tableau 2.2, déterminez la valeur EtherType utilisée pour le protocole IP.
15. Dans les réseaux TCP/IP, quel champ utilise-t-on dans un paquet IP pour le protocole de multiplexage/démultiplexage ?
16. Pourquoi un protocole de multiplexage/démultiplexage est-il nécessaire dans les réseaux ?
17. A partir du Tableau 2.3, déterminez les valeurs d'identification du protocole pour les protocoles TCP et UDP.
18. Quel champ, dans un paquet TCP ou UDP, emploie-t-on pour déterminer le service TCP/IP à qui est destiné le paquet ?
19. A partir des Tableaux 2.4 et 2.5, déterminez le numéro de port et le protocole de la couche de transport utilisée par TELNET, SMTP, SNMP, HTTP, TFTP et DNS.
20. Nommez les trois méthodes utilisées pour implémenter les protocoles TCP/IP dans la plupart des systèmes d'exploitation. Donnez des exemples de systèmes d'exploitation utilisant ces méthodes.

3 Support réseau pour TCP/IP

Une des raisons de la popularité de TCP/IP provient de l'adaptabilité de ces protocoles à la plupart des réseaux et des périphériques. La suite de protocoles TCP/IP peut tourner sur des mainframes, des mini-ordinateurs, des stations de travail, des palmtops et même sur des téléphones GSM !

Le matériel et les logiciels utilisés sont totalement transparents à l'utilisateur. En fait, la plupart des utilisateurs ignorent qu'ils utilisent les protocoles TCP/IP. Mais comme ce livre est écrit pour vous, professionnels réseau qui voulez tout apprendre sur TCP/IP, nous pensons qu'il est intéressant de comprendre comment TCP/IP tourne sur la partie matérielle d'un réseau.

Vue d'ensemble de la partie matérielle concernée par TCP/IP

Par matériel réseau, il faut comprendre le matériel associé aux couches 1 et 2 du modèle OSI. La couche 1 représente la couche physique et la 2, la couche liaison des données. Cette couche reçoit et émet des données sous forme de trames et interagit avec la couche physique qui est responsable de la transmission des trames sur le support physique. Les cartes d'interface réseau sont des accessoires insérés dans l'ordinateur et reliant celui-ci au câblage réseau. Les cartes réseau implémentent les fonctionnalités de la couche liaison et de la couche physique.

Avant de décrire les différents matériels réseau disponibles pour TCP/IP, vous devez vous poser une question fondamentale : pourquoi ai-je besoin de tant de types différents de matériels pour faire tourner TCP/IP ?

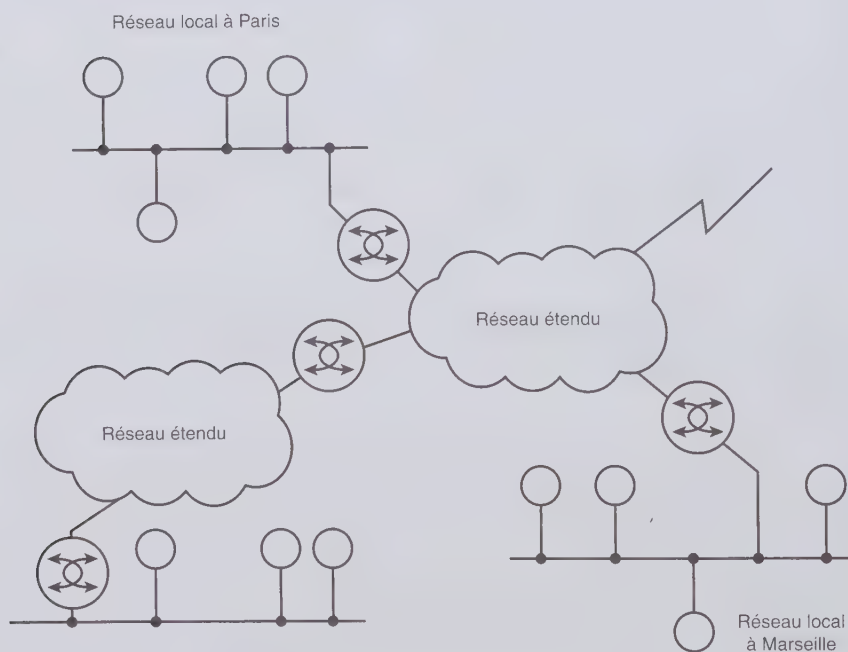
Disposer de différents types de matériels répond à la difficulté, pour un type, de satisfaire à toutes les caractéristiques d'un réseau en fonctionnement normal. Beaucoup d'entreprises utilisent un réseau local à un endroit donné (parce qu'un réseau local fournit des débits élevés et n'est pas très coûteux). Cependant, les réseaux locaux sont limités en distance et ne peuvent s'étendre sur plusieurs sites distants.



Pour relier des sites distants, il est nécessaire d'utiliser des réseaux étendus (WAN), qui connectent les sites entre eux. Les réseaux étendus relèvent de technologies différentes de celles des réseaux locaux. Comme leur nom l'indique, ils doivent connecter des réseaux sur de grandes distances. De ce fait, les vitesses seront plus lentes que dans les réseaux locaux. Ainsi, un seul réseau local ou un seul réseau étendu ne peut, à lui seul, couvrir les besoins d'une entreprise. Dans ce cas, une combinaison des deux types de réseaux est nécessaire (voir Figure 3.1).

Figure 3.1

Combinaison de réseaux locaux et de réseaux étendus.



Ainsi, un réseau est souvent formé d'un ou de plusieurs réseaux locaux et réseaux étendus. Pour relier des réseaux locaux et des réseaux étendus, il est nécessaire d'utiliser des périphériques spéciaux appelés *ponts*, *routeurs* ou *passerelles* (ce chapitre définit le rôle des ponts et des routeurs dans la liaison entre différents réseaux. Les chapitres suivants décriront plus en détail les routeurs, les passerelles et le routage sur l'Internet).

L'un des objectifs de l'interconnexion réseau consiste à fournir aux utilisateurs les ressources informatiques dont ils ont besoin, sans se soucier de l'endroit où elles se trouvent. De plus, ces accès doivent être transparents aux utilisateurs. En d'autres termes, la complexité de l'interconnexion, les périphériques, le câblage, etc., doivent être cachés à l'utilisateur. En revanche, toutes les "coulisses"

de cet environnement doivent être connues du responsable réseau pour la gestion et la maintenance de l'ensemble.

La Figure 3.1 montre un réseau formé de réseaux locaux et de réseaux étendus. Dans chaque type de réseau se trouvent différents types de matériels. Les choix doivent porter sur les besoins de l'entreprise. Voici quelques technologies de réseaux locaux :

- IEEE LAN ;
- Ethernet ;
- Token Ring ;
- réseaux commutés ;
- FDDI.

Voici quelques techniques utilisées dans ces réseaux :

- X.25 ;
- Relais de trame (*frame relay*) ;
- SMDS ;
- MAN ;
- SLIP/PPP/CSLIP ;
- ATM.

TCP/IP sur des réseaux locaux IEEE

Dans le monde des réseaux, on rencontre souvent des références aux standards 802 de l'*Institute of Electrical and Electronic Engineers* (IEEE). Ce sont des standards réseau portant sur les deux premières couches du modèle OSI. L'IEEE définit les réseaux Ethernet par la norme nommée 802.3, et les réseaux Token Ring par la norme 802.5. Ces standards se retrouvent dans un grand nombre de réseaux sur lesquels tourne TCP/IP. Voici une brève présentation de l'évolution des standards IEEE.

Le projet 802 de l'IEEE a commencé en février 1980 afin de formaliser les standards réseau pour des débits n'excédant pas 20 Mbit/s. Le numéro 802 fut choisi par rapport à l'année (80 pour 1980) et au mois (2 pour février).



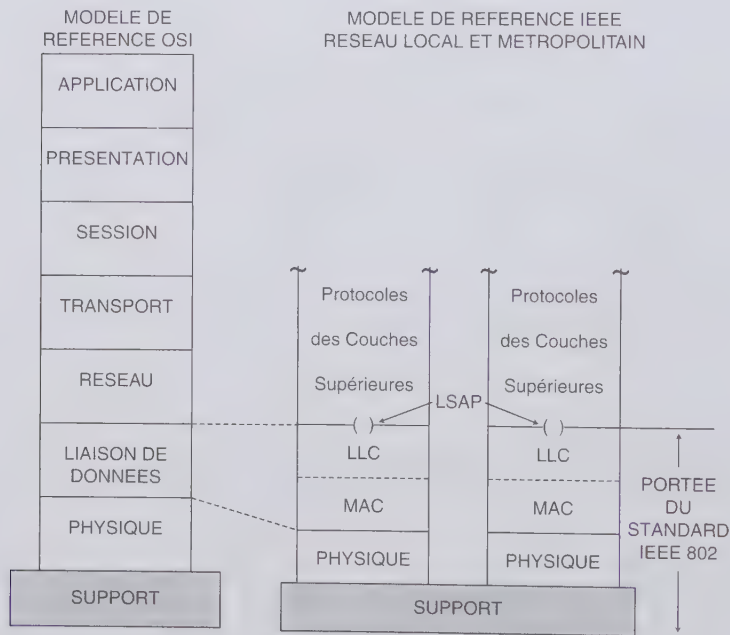
La Figure 3.2 montre le standard des réseaux IEEE comparé au modèle OSI. Sur cette figure, vous pouvez voir que le premier souci de l'IEEE a été de standardiser les techniques matérielles des deux premières couches physique et liaison des données, car les matériels réseau tels que les cartes ou le câblage sont totalement définis par ces deux premières couches.

L'IEEE divise la couche liaison des données en deux sous-couches :

- *Media Access Control (MAC)* ;
- *Logical Link Control (LLC)*.

Figure 3.2

Relations entre le standard de l'IEEE et le modèle OSI.



• Couche *Media Access Control* (MAC)

La couche MAC se charge des techniques d'accès utilisées pour contrôler l'accès au support physique du réseau. Token Ring et Ethernet possèdent des implémentations différentes de la couche MAC, car les méthodes d'accès au support sont différentes pour les deux réseaux. Cela signifie que, pour relier des réseaux Ethernet et Token Ring, il faut un périphérique de type pont ou routeur entre les deux réseaux.

• Couche *Logical Link Control* (LLC)

Tous les réseaux définis par l'IEEE possèdent la même couche LLC définie par le standard 802.2. L'avantage d'une sous-couche commune réside dans le fait que les mécanismes de la couche supérieure sont les mêmes, quel que soit le type de réseau utilisé.

La Figure 3.2 montre l'interface entre les protocoles de la couche supérieure et de la couche LLC, lesquels sont définis par LSAP (*Link Service Access Points*). LSAP représente des adresses logiques de la couche liaison de données. Une seule adresse MAC, telle qu'une adresse Ethernet, peut posséder plusieurs adresses LSAP. Ces multiples adresses LSAP permettent plusieurs connexions entre deux nœuds du réseau.

La couche LLC constitue la sous-couche supérieure de la couche liaison des données. Cette couche est responsable de la transmission des données entre deux nœuds sur le réseau. Ces nœuds possèdent des adresses LSAP appelées DSAP (*Destination Service Access Point*) et SSAP (*Source Service Access Point*). La couche LLC fournit aussi les services de datagramme (mode non connecté), ou de circuit virtuel (mode connecté) ou une combinaison des deux.

Services de datagramme

Dans un datagramme, chaque paquet contient les adresses source et destination. Aucune vérification n'est faite pour savoir si les paquets arrivent intacts et dans le bon ordre. Les services de datagramme se font en mode non connecté et peuvent ou non utiliser les accusés de réception. Ces accusés de réception sont des trames spéciales, envoyées par le destinataire qui annonce à la source que les données ont été correctement reçues. En cas de problème, les données sont retransmises. Cela arrive lors d'un problème matériel ou d'interférence sur la ligne. Le Chapitre 2 explique comment le CRC est utilisé pour détecter les erreurs de transmission.

Des services de datagramme sans accusé de réception cela signifie que les datagrammes sont envoyés, mais que rien n'est fait au niveau de la couche liaison des données pour retransmettre les trames perdues. Pour les applications qui nécessitent une garantie de transmission, c'est la couche transport du modèle OSI qui se charge de cette vérification. Dans les réseaux TCP/IP, la couche transport est implémentée par TCP.

Le service de datagramme sans accusé de réception est appelé service de type 1 dans les réseaux locaux IEEE. Le protocole Internet, implémenté au niveau de la couche 3 du modèle OSI, n'utilise pas les accusés de réception. C'est un service en mode non connecté. Cela simplifie le mappage des datagrammes IP dans les

* 30 *



trames utilisées par les réseaux locaux. Normalement, un seul datagramme IP est encapsulé dans une trame d'un réseau local (voir Figure 3.3). Quand la taille du datagramme est trop importante pour tenir dans une seule trame, le datagramme est coupé en plusieurs fragments pour tenir dans une trame, et chaque fragment du datagramme est envoyé dans une trame (voir Figure 3.4). Vous verrez dans les chapitres suivants les mécanismes qui permettent de garder la trace des fragments des datagrammes.

Figure 3.3

Un datagramme IP dans une trame d'un réseau local.

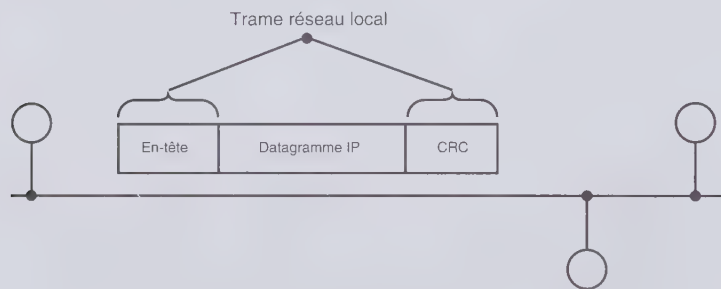
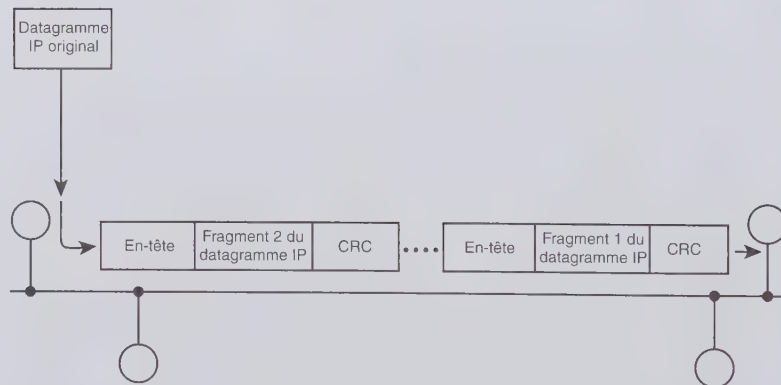


Figure 3.4

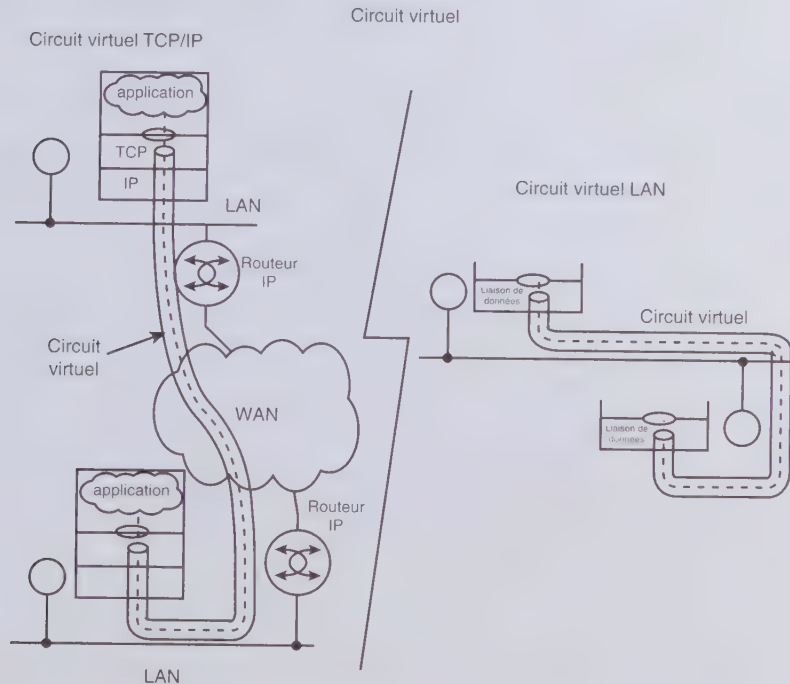
Fragments de datagramme IP dans plusieurs trames.



Circuit virtuel

Dans un circuit virtuel, une connexion est établie entre deux nœuds avant que les données ne soient échangées entre eux. Dès que toutes les données ont été transmises, le circuit virtuel est coupé. Les services des circuits virtuels sont appelés services de type 2. Dans les réseaux TCP/IP, les circuits virtuels sont rarement utilisés au niveau de la couche MAC d'un réseau local, car le protocole TCP est utilisé pour créer un circuit virtuel dans un réseau étendu (voir Figure 3.5). Le service de type 2 est créé entre deux nœuds d'un même réseau ou de deux réseaux identiques reliés par un pont.

Figure 3.5
Circuits virtuels TCP
et circuits virtuels LAN.



Services de datagramme avec accusé de réception

Les services de datagramme avec accusé de réception sont une combinaison de datagrammes et de circuits virtuels. Ce sont des services de type 3 dans lesquels les erreurs de transmission sont corrigées par retransmission des paquets en erreurs. Les services de type 3 sont utilisés dans les réseaux IBM pour transmettre des données SNA sur un réseau local.

En résumé, les types de services fournis par la couche LLC sont les suivants :

- **Type 1.** Service de datagramme sans accusé de réception. Supporte le point à point, le multipoint et les diffusions.
- **Type 2.** Service des circuits virtuels. Fournit les services en séquence, contrôle de flux et correction d'erreurs entre LSAP.
- **Type 3.** Service de datagramme avec accusé de réception. Fournit des datagrammes point à point.



La Figure 3.6 montre comment l'IEEE a identifié les choix au niveau des différentes couches.

Figure 3.6

*Services définis
par les standards IEEE 802.*

Logical Link Control (LLC)	IEEE 802.2		
	Type 1	Service de datagramme sans accusé de réception	
	Type 2	Service de circuit virtuel	
	Type 3	Service de datagramme avec accusé de réception	
Medium Access Control (MAC)	IEEE 802.3	IEEE 802.4	IEEE 802.5
	Medium Access Control (MAC) CSMA/CD	Medium Access Control (MAC) Token-Bus	Medium Access Control (MAC) Token-Ring
Support physique	Coaxial bande de base 1, 10 Mbits/s Paires torsadées bande de base 10 Mbits/s Coaxial large bande 10 Mbits/s	Coaxial bande de base 1, 5, 10 Mbits/s Canal large bande 1, 5, 10 Mbits/s Fibre optique 5, 10, 20 Mbits/s	Paires torsadées blindées 1, 4 Mbits/s Paires torsadées non blindées

Chacun de ces choix représente un protocole standard ou une spécification. Les références IEEE et leur signification sont représentées dans le Tableau 3.1.

Tableau 3.1 : Standards IEEE

Standard	Signification
IEEE 802.1	Réseau ponté
IEEE 802.2	LLC (Logical Link Control)
IEEE 802.3	Standard de la technologie Ethernet
IEEE 802.4	Standard Token Bus
IEEE 802.5	Standard Token Ring
IEEE 802.6	Réseau métropolitain (MAN)
IEEE 802.7	Technique bande de base

Tableau 3.1 : Standards IEEE (suite)

Standard	Signification
IEEE 802.8	Technique fibre optique
IEEE 802.9	Intégration voix/données (IVD)
IEEE 802.10	Sécurité réseau
IEEE 802.11	Réseaux locaux sans fil
IEEE 802.12	100 VG-AnyLAN

Règles de câblage des réseaux TCP/IP

Chaque standard de réseau local possède ses propres règles de câblage. Ces règles définissent le support physique, les caractéristiques matérielles et la disposition des différents éléments. Deux règles concernent le support :

- la disposition du câblage ;
- le type de support.

• Disposition du câblage

La disposition du câblage est appelée *topologie physique*. Voici les principales topologies :

- l'étoile ;
- le bus ;
- l'anneau.

Ces topologies physiques sont représentées sur la Figure 3.7.

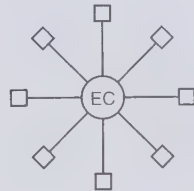
Topologie en étoile

Dans une topologie en étoile, la communication entre deux nœuds du réseau se fait par l'intermédiaire d'un élément central. Les systèmes qui se connectent sur l'élément central sont simples, et toute la complexité réside dans l'élément central. Cet élément doit être fiable et fournir une séparation des signaux entre les ports reliés aux différents systèmes afin qu'un problème sur l'un des ports ne se transmette pas aux autres ports.



Figure 3.7

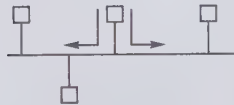
Les différentes topologies physiques des réseaux locaux.



Topologie en Etoile

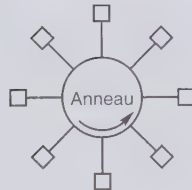
EC = Élément central

- Le média n'est pas partagé
- EC effectue les fonctions de connexion centralisées



Topologie en bus

- Média partagé
- Accès au média par un contrôle distribué
- Emission bidirectionnelle



Topologie en anneau

- Média partagé
- Accès au média par contrôle distribué
- Emission unidirectionnelle

L'architecture des mainframes et des mini-ordinateurs sont des exemples de topologie en étoile. Le centre de l'étoile est représenté par l'hôte. Mais si un problème survient sur l'hôte, les communications entre les nœuds sur le réseau sont coupées. Dans ce cas, la vulnérabilité de la topologie en étoile constitue un point faible de ces systèmes.

En revanche, si l'élément central présente à la fois une bonne fiabilité et une isolation entre les ports, la topologie en étoile représente l'un des meilleurs choix. Cela explique son utilisation dans les réseaux Token Ring, FDDI et 10BASE-T. Un autre avantage provient du fait qu'il est facile de déconnecter une station de l'élément central. Dans beaucoup de réseaux locaux, l'élément central (un hub) possède une gestion de réseau comme par exemple SNMP (*Simple Network Management Protocol*).

Topologie en bus

La topologie en *bus* consiste en un câble linéaire sur lequel sont connectées toutes les stations du réseau. Les signaux envoyés par une station se propagent sur tout le câble dans les deux sens. Chaque transmission est reçue par chaque station plus ou moins simultanément. Ethernet est un exemple classique de topologie en bus.

Topologie en anneau

La topologie en *anneau* (*ring*) consiste en un câble disposé en anneau fermé sur lequel sont connectées les différentes stations du réseau. Les signaux ne sont envoyés que dans un seul sens. Les stations ne voient les données qu'au moment où elles passent devant elles. Le réseau Token Ring est un exemple de topologie en anneau.

Il faut faire une distinction importante entre la topologie physique et la topologie logique. La topologie *physique* décrit la façon dont le câblage est construit. La topologie *logique* décrit la façon dont les données circulent dans les câbles. Vous pouvez "voir" ou "toucher" la topologie physique, vous ne pouvez ni voir ni toucher la topologie logique.

Token Ring est décrit comme une topologie en anneau, car les données passent d'une station à une autre station jusqu'à ce qu'elles aient fait le tour de l'anneau. L'anneau d'un réseau Token Ring ressemble cependant à une étoile, car les câbles qui partent d'une station pour aller à la station suivante passent par un élément central. De même, le réseau Ethernet possède une topologie en bus bien que, dans la câblage 10BASE-T, la disposition soit en étoile.

• Choix du support physique

Ce paragraphe décrit rapidement les différents supports possibles présentés dans la Figure 3.8.

Câble coaxial

Le câble coaxial est formé d'une *âme* en cuivre utilisée pour transmettre les signaux. Le retour du signal se fait par une gaine conductrice qui entoure l'âme, les deux étant séparées par un isolant. La gaine conductrice joue le rôle d'écran et isole l'âme des interférences extérieures et donne ainsi aux câbles coaxiaux une bonne largeur de bande. Ce type de câble a été utilisé dans les premiers réseaux locaux, spécialement dans la topologie en bus.

Câble en paires torsadées

Le câble en paires torsadées est formé de paires de fil conducteur, contenues dans une gaine isolante. Les paires sont torsadées pour éviter les interférences électriques. Les câbles en paires torsadées peuvent être entourés d'un blindage métallique pour diminuer l'effet des interférences électromagnétiques créées par des



machines extérieures au réseau. Dans un câble en paires torsadées, l'une des paires sert à l'émission du signal, l'autre à la réception. Les câbles en paires torsadées sont peu coûteux et faciles à installer.

Câble en fibre optique

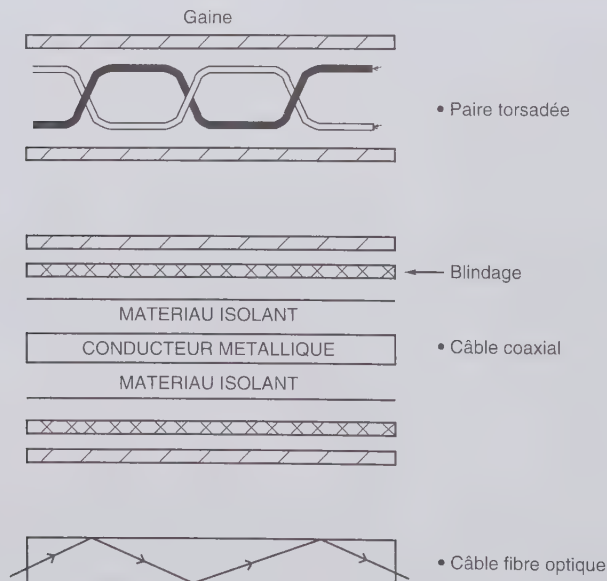
Les câbles en fibre optique sont formés d'une fibre très fine en verre ou en plastique, entourée par une gaine protectrice. Les signaux sont transmis dans la fibre sous forme lumineuse. Les signaux lumineux peuvent se propager dans la fibre sur une longue distance, sans qu'il y ait nécessité d'amplification comme avec les câbles métalliques. La fibre optique présente une très bonne immunité aux bruits, comparée aux autres supports, et elle possède aussi une très large bande passante.

Les bruits sont souvent occasionnés par des appareils électriques, comme des moteurs électriques, des tubes fluorescents, des appareils radio, etc. Comme la fibre optique transporte un signal sous forme lumineuse, elle est insensible aux bruits électromagnétiques.

Aujourd'hui, les réseaux locaux à haut débit utilisent un support en fibre optique pour transmettre les données, bien que les coûts des équipements terminaux soient plus élevés que pour les câbles de cuivre.

Figure 3.8

Les différents types de câbles.



Fonctionnement des réseaux Ethernet

Robert Metcalfe, David Boggs et d'autres chercheurs de Xerox Corporation développèrent un réseau local fondé sur des mécanismes de détection de porteuse. Ce réseau possédait une longueur d'un kilomètre, supportait cent stations et transmettait les données à 2,94 Mbit/s. Ce système a été appelé Ethernet du nom de cette substance impalpable, l'éther, imaginée au XVIII^e siècle pour expliquer la propagation des ondes électromagnétiques (le terme "net" rappelle, lui, qu'il s'agit d'un réseau).

Ethernet a été proposé comme standard par Digital Equipment Corporation, Intel et Xerox. Le premier standard a été publié en septembre 1981 et a été appelé DIX 1.0. DIX signifie Digital, Intel et Xerox. DIX 1.0 a été suivi par DIX 2.0, publié en novembre 1982. Le standard DIX 2.0 est aussi appelé Ethernet II.

Pendant ce temps, l'IEEE a continué la standardisation des réseaux locaux, commencée par le projet 802. Digital, Intel et Xerox proposaient l'adoption d'Ethernet comme standard. IBM proposa Token Ring (basé sur des prototypes construits dans les laboratoires d'IBM à Zurich) comme standard. La proposition Ethernet devint le standard IEEE 802.3, et Token Ring devint le standard 802.5.

Du fait de la nature même du comité, le standard IEEE 802.3 n'est pas tout à fait identique à celui d'Ethernet. Il existe même d'importantes différences. Bien que les standards 802.3 et Ethernet soient incompatibles, le terme *Ethernet* est utilisé dans les réseaux TCP/IP pour désigner les réseaux 802.3. Ce livre adopte l'usage commun et utilise le terme Ethernet pour les deux standards, en notant les différences quand cela est nécessaire.

Ethernet est aussi connu sous d'autres noms. En 1982, l'Association européenne des constructeurs d'ordinateurs l'a adopté comme standard 80/82/82. En 1989, l'ISO l'adopta comme standard ISO/IEC 8802-3.

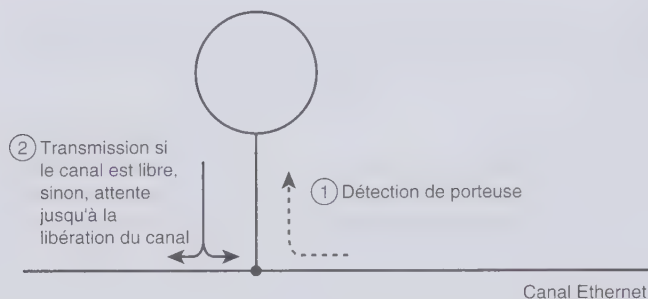
Les paragraphes suivants présentent Ethernet pour les réseaux TCP/IP.

• Méthode d'accès sur les réseaux Ethernet

Quand une station d'un réseau Ethernet veut émettre, elle commence à écouter les trames qui circulent sur le câble (voir Figure 3.9). Si elle détecte une émission en cours, elle attend avant de transmettre ses données.



Figure 3.9

Méthode d'accès sur Ethernet.

Après les derniers bits de la trame en cours de transmission, la couche de liaison des données attend encore 9,6 microsecondes avant de faire passer une autre trame. A la fin de ce temps, si une trame attend et si personne n'émet, la transmission de cette trame commence.

Si une station essaie d'émettre quand le canal de transmission est occupé, une *collision* se produit.

Si le média est libre (aucune porteuse n'est détectée), la station peut émettre. Comme plusieurs stations du réseau utilisent ce mécanisme de détection de porteuse, on dit que la méthode d'accès est distribuée, et elle est dite à accès multiple et à détection de porteuse (CSMA : *Carrier Sense with Multiple Access*).

Si deux stations décident d'émettre en même temps, une collision se produit. Les collisions se produisent pendant les opérations normales de transmission, car les stations envoient leur données en se fondant uniquement sur la présence ou non d'une porteuse sur le média. Les stations ignorent si des trames, provenant d'autres stations, attendent avant d'être transmises. De plus, la transmission sur les réseaux locaux s'accompagne d'un délai de propagation. Pour Ethernet, par exemple, le signal se propage à 0,77 fois la vitesse de la lumière dans les câbles standards (gros Ethernet) et à 0,65 fois la vitesse de la lumière pour les câbles fins.

Les collisions sont normales dans les réseaux Ethernet. Les stations minimisent les effets des collisions en les détectant quand elles se produisent. C'est pour cela que la méthode d'accès s'appelle à *détection de collisions* (CD : *Collision Detect*). Les stations émettant pendant une collision arrêtent cette émission. La première station ayant détecté la collision envoie un message pour alerter les autres stations qu'une collision a eu lieu. Après la collision, chaque station attend pendant un délai aléatoire avant de tenter une nouvelle émission.

Si des collisions multiples se suivent, les délais aléatoires sont doublés pendant dix collisions successives. Au-delà, le doublement du temps ne modifie plus les performances du réseau.

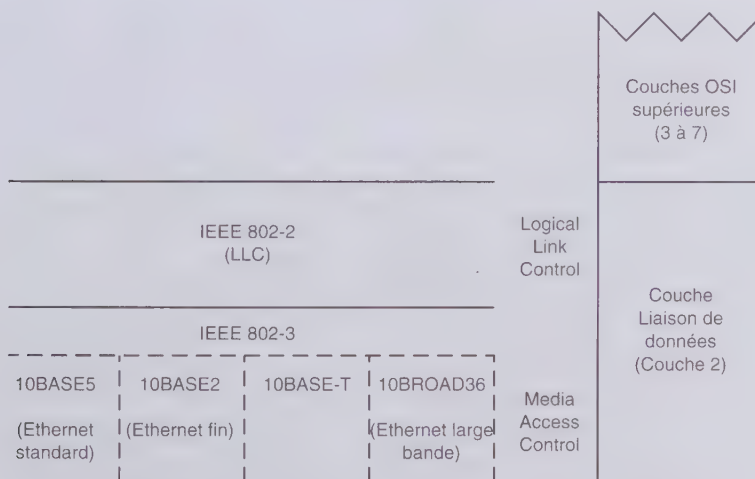
Combien de temps une station doit-elle attendre pour transmettre ? Il se pourrait qu'une station attende indéfiniment, mais ce ne serait pas acceptable ; beaucoup de réseaux utilisent des systèmes modifiés du CSMA/CD afin de garantir la disponibilité du réseau. Ces Ethernet modifiés ne suivent pas le standard Ethernet présenté dans ce livre.

Le paragraphe suivant examine les différentes options Ethernet.

• Options de câblage Ethernet

Le câble coaxial pour Ethernet existe en deux modèles possibles : le câble standard et le câble Ethernet fin. Ethernet peut aussi être installé sur des câbles en paires torsadées (voir Figure 3.10).

Figure 3.10
*Options IEEE
pour Ethernet 802.3.*



• Règles de câblage du câble standard Ethernet

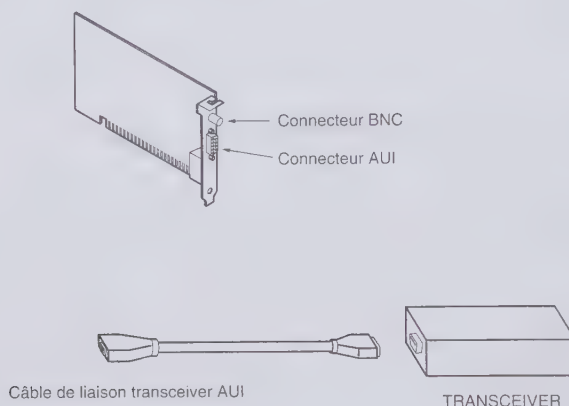
Le câble standard Ethernet porte aussi le nom d'Ethernet épais, car ce câble est plus épais que le câble Ethernet fin. Le câble standard est de type RG-8, et la version IEEE s'appelle 10BASE5. La valeur 10 se rapporte au débit de 10 Mbit/s, BASE signifie transmission en bande de base, et 5 correspond à la longueur maximale de 500 mètres pour un segment.



La Figure 3.11 montre quelques éléments matériels pour Ethernet standard. La carte réseau possède un connecteur AUI et un connecteur coaxial. Le connecteur coaxial permet d'utiliser le câble fin.

Figure 3.11

Éléments matériels pour les réseaux en Ethernet épais.



Les stations sur un câble épais sont reliées au câble par des transceivers. Le câble est appelé *segment*, et sa longueur ne doit pas dépasser 500 mètres à cause de l'affaiblissement des signaux.

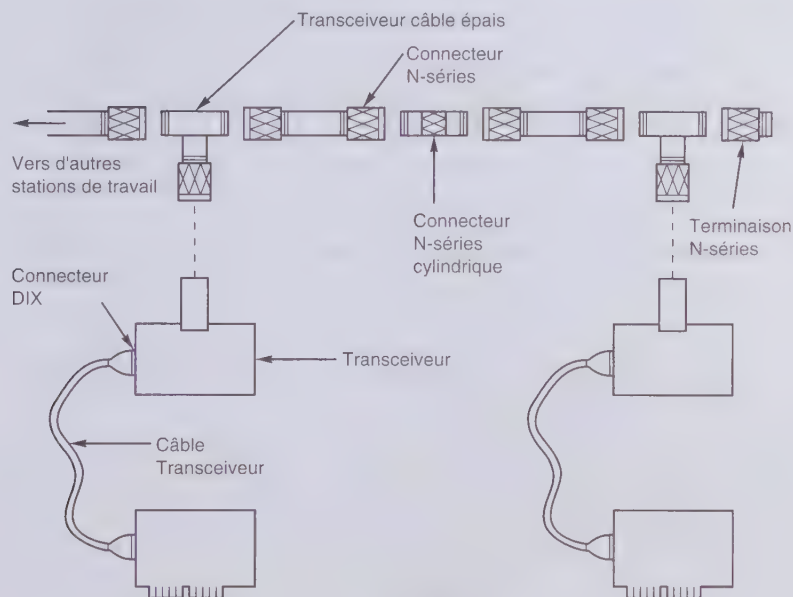
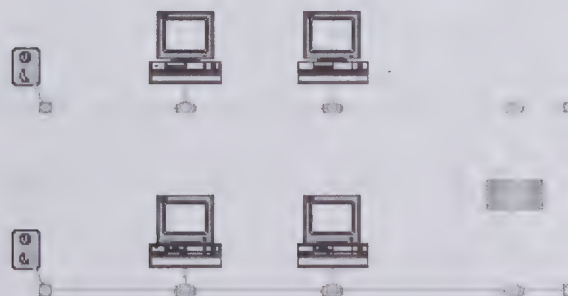
Le transceiver et la carte réseau sont reliés par un câble. Sur chaque connecteur DIX, un loquet permet d'éviter un débranchement accidentel du câble.

La Figure 3.12 montre le câble Ethernet épais dans un réseau Ethernet. Son épaisseur est de 2,14 millimètres et son impédance de 50 ohms. Chaque extrémité d'un segment est terminée par une résistance de 50 ohms qui absorbe les réflexions. Les résistances de terminaison possèdent une mise à la terre, mais une seule terminaison doit être mise à la masse pour éviter les courants parasites engendrés par l'anneau que constitueraient deux terminaisons reliées à la terre.

La Figure 3.13 montre un exemple de réseau en Ethernet épais. Deux segments sont reliés par un *répéteur*. Un répéteur est un élément actif qui amplifie et régénère le signal. Des répéteurs multiports peuvent relier plus de deux segments entre eux.

Le Tableau 3.2 recense les règles à respecter dans le câblage en Ethernet épais.

Sur un réseau coaxial Ethernet, les stations sont connectées sur des segments principaux (voir Figure 3.14), et un signal ne peut traverser plus de deux répéteurs joignant directement deux segment principaux. Des câbles en fibre optique ou en paires torsadées peuvent relier des segments principaux entre eux. Il ne peut en exister que deux sur un réseau Ethernet, et aucune station ne doit être connectée sur ces câbles intersegments qui forment des segments de liaison ou IRL (*Inter Repeater Link Segments*).

Figure 3.12*Connecteurs pour câbles épais.***Figure 3.13***Exemple d'un réseau en câble épais.***Tableau 3.2 : Paramètres et règles des câbles épais**

Paramètres d'Ethernet épais	Valeur
Débit maximal	10 Mbit/s
Nombre maximal de répéteurs sans segment de liaison	2
Maximum de répéteurs avec segments de liaison	4

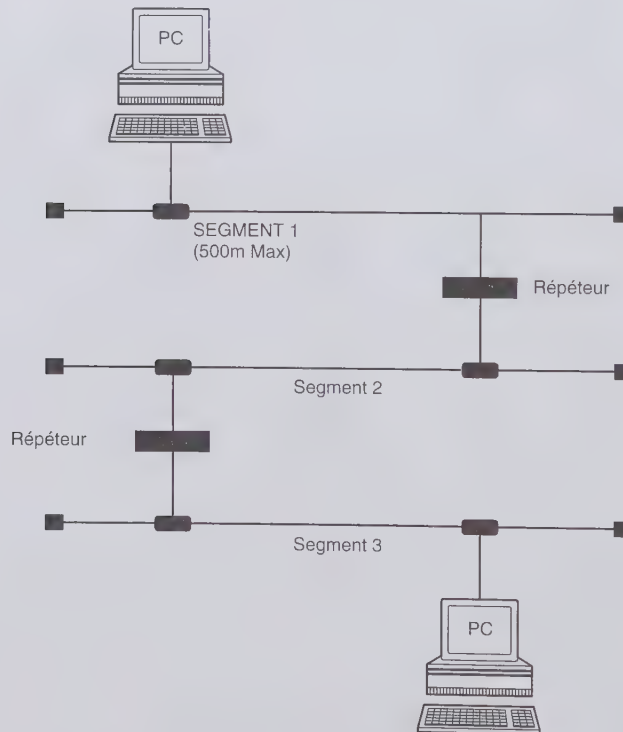


Tableau 3.2 : Paramètres et règles des câbles épais (suite)

Paramètres d'Ethernet épais	Valeur
Longueur maximale d'un segment	500 mètres
Longueur maximale d'un câble transceiver	50 mètres
Nombre maximal de segments de liaison	2
Longueur maximale des segments de liaison	1 000 mètres
Nombre maximal de stations par segment	100
Nombre total de stations	1 024
Distance minimale entre deux stations	2,5 mètres

Figure 3.14

Longueur totale d'un réseau local Ethernet sans câble intersegment.



Des répéteurs relient un segment à un câble intersegment. La longueur totale des liaisons intersegments ne peut dépasser 1 000 mètres, et celle d'un segment, 500 mètres. A partir de ces valeurs, il est facile de calculer la longueur totale

d'un réseau local Ethernet (voir Figure 3.15). Sur cette figure, T1 à T6 représentent des transceivers.

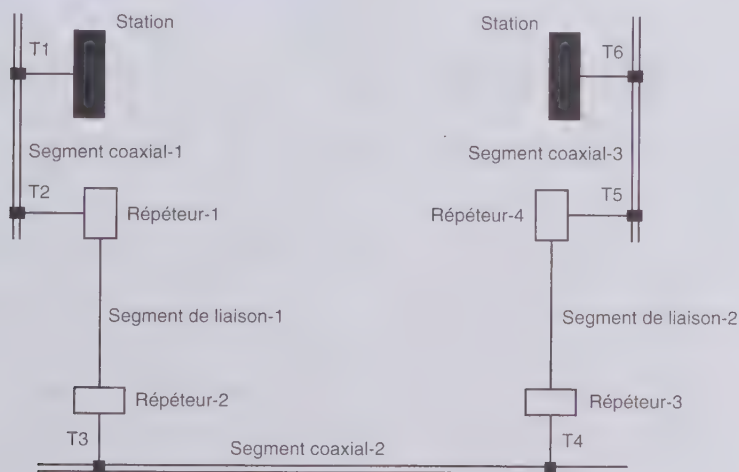
A partir de la Figure 3.15, vous pouvez calculer la longueur totale d'un réseau local Ethernet :

Longueur du segment 1	= 500 mètres
Longueur du segment 2	= 500 mètres
Longueur du segment 3	= 500 mètres
Longueur des deux liaisons intersegments	= 1 000 mètres
Longueur totale du réseau	= 2 500 mètres

Vous pouvez tirer avantage de la longueur des câbles transceivers en ajoutant ces câbles aux transceivers T1, T2, T3, T4, T5 et T6 (voir Figure 3.15). Comme la longueur maximale d'un câble transceiver est de 50 mètres, cela ajoute une longueur de 300 mètres au réseau, soit une longueur totale de 2 800 mètres.

Figure 3.15

*Longueur maximale
d'un réseau local Ethernet.*



Légende : T1, T2, T3, T4, T5, T6 = Transceivers

 = Répéteur

 = Station



Le nombre de stations sur un segment épais ne doit pas dépasser 100, et le nombre total de stations sur un réseau local ne peut excéder 1 024. Un répéteur compte pour une station. La distance minimale entre deux stations est de 2,5 mètres. Il est recommandé de disposer les stations à des distances multiples de 2,5 mètres afin d'éviter les interférences causées par des ondes stationnaires sur le segment.

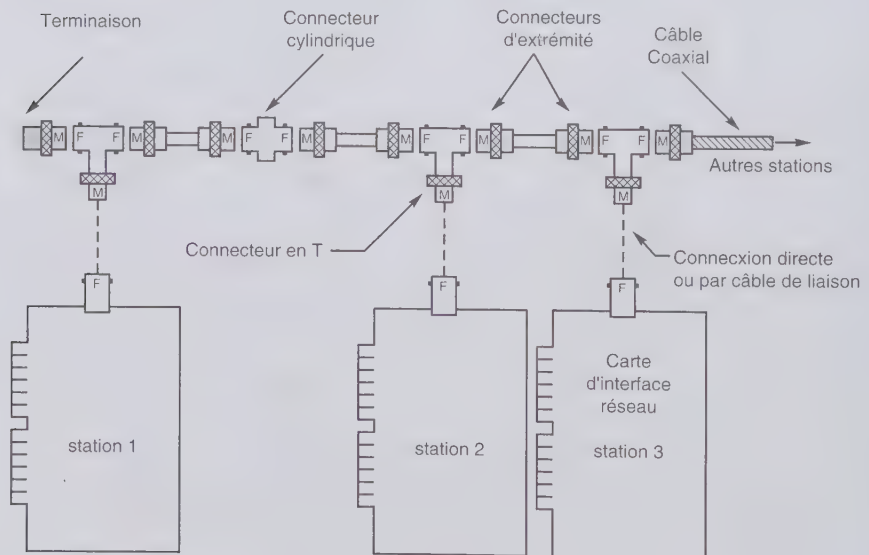
- Règles de câblage du câble fin Ethernet

Le câble coaxial fin Ethernet porte aussi les noms de *Thinnet* ou de *Cheapnet*. Ce câble est plus fin que le câble Ethernet standard. La version IEEE du câble fin est appelée *10BASE2*. 10 signifie 10 Mbit/s, BASE indique une transmission en bande de base, et 2 signifie que la longueur d'un segment doit être inférieure à 200 mètres (exactement 185 mètres).

La Figure 3.16 montre quelques éléments d'un réseau Ethernet en câble coaxial fin.

Figure 3.16

Les éléments d'un réseau Ethernet en câble fin.



Les fonctions des transceivers d'un réseau en Ethernet fin sont effectuées par la carte réseau. Il n'est donc pas nécessaire d'installer de transceiver sur le réseau. Des connecteurs BNC en T relient les cartes au câble. Les deux extrémités opposées d'un T permettent de connecter deux segments de câble fin pour continuer le bus.

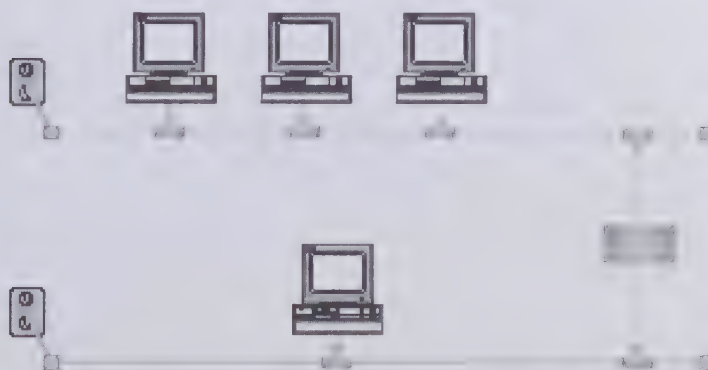
La longueur d'un câble fin ne peut excéder 185 mètres, à cause de l'atténuation des signaux. Le câble fin est de type RG-58. Son diamètre est de 0,5 centimètres, et son impédance de 50 ohms. Ce câble est disponible en longueurs fixes, et les

extrémités contiennent les connecteurs BNC. Vous pouvez fabriquer des câbles de longueur quelconque et sertir les connecteurs BNC au moyen d'une pince spéciale. Vous pouvez aussi utiliser des prolongateurs BNC pour relier deux segments de câble. Un segment doit être terminé aux deux extrémités par une résistance de 50 ohms. Une seule résistance de terminaison doit être mise à la terre.

La Figure 3.17 montre un exemple d'un réseau local Ethernet en coaxial fin. Le répéteur de la Figure 3.17 possède deux ports pour y connecter deux segments.

Figure 3.17

Exemple d'un réseau local Ethernet en coaxial fin.



Le Tableau 3.3 recense les différentes règles à respecter dans un réseau Ethernet en coaxial fin.

Tableau 3.3 : Paramètres d'Ethernet fin

Paramètre	Valeur
Débit	10 Mbit/s
Nombre maximal de répéteurs sans câble d'interconnexion	2
Nombre maximal de répéteurs avec câbles d'interconnexion	4
Longueur maximale d'un segment	185 mètres
Nombre maximal de câbles de connexion	2
Nombre maximal de stations par segment	30
Nombre maximal de stations	1 024
Distance maximale entre deux stations	0,5 mètre



Les règles à observer pour les répéteurs sont les mêmes que pour le câble standard.

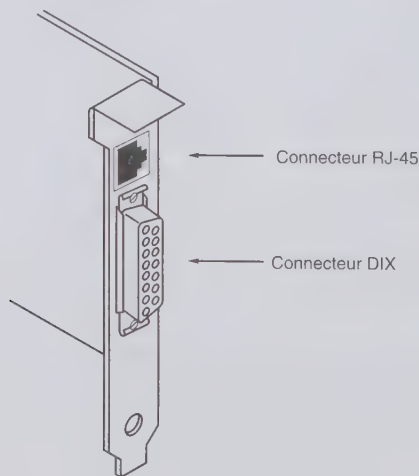
• Règles de câblage des câbles 10BASE-T

L'intérêt pour les réseaux 10BASE-T commença dans les années 90 avec la baisse des prix des éléments actifs. 10BASE-T est une option d'implémentation du standard IEEE 802.3. 10 signifie 10 Mbit/s, BASE signifie que les émissions se font en bande de base, et T se réfère aux paires torsadées (*twisted pairs*) utilisées dans ce type de câblage.

Sur la Figure 3.18, la carte réseau possède une prise RJ-45, appelée officiellement port MDI (*Media Dependent Interface*). La carte réseau représentée sur cette figure possède aussi un connecteur DIX. Ce connecteur permet de relier la carte à un câble épais par l'intermédiaire d'un transceiver. Ainsi, cette carte peut être utilisée soit avec un câble épais, soit avec un câble 10BASE-T. Certaines cartes nécessitent un inverseur pour activer le port 10BASE-T ou le port AUI. D'autres possèdent un système de détection automatique du type de câble. En mode 10BASE-T, les fonctions du transceiver sont effectuées par les composants de la carte réseau.

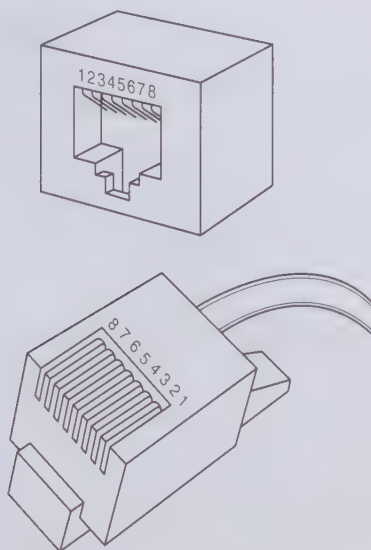
Figure 3.18

Connecteurs sur une carte réseau 10BASE-T.



La topologie physique utilisée avec la technologie 10BASE-T est une étoile avec un *hub*, servant de répartiteur entre les stations. Les connecteurs mâle et femelle de la technologie 10BASE-T sont représentés sur la Figure 3.19. Les câbles provenant des stations sont connectés au hub. Les fils utilisés dans les paires torsadées ont de 0,4 à 0,6 millimètre de diamètre (26 à 22 AWG) et forment des câbles multipaires. Les spécifications de performance indiquent une longueur maximale de 100 mètres pour la paire.

Figure 3.19
Connecteurs 10BASE-T.



Le câble reliant la carte réseau au hub comprend deux paires torsadées (voir Figure 3.20). L'une des paires sert à l'émission, l'autre à la réception des données.

Le Tableau 3.4 recense les fonctions des broches d'un câble quatre paires. Seules deux paires sont utilisées, l'une pour la transmission (TD), l'autre pour la réception (RD).

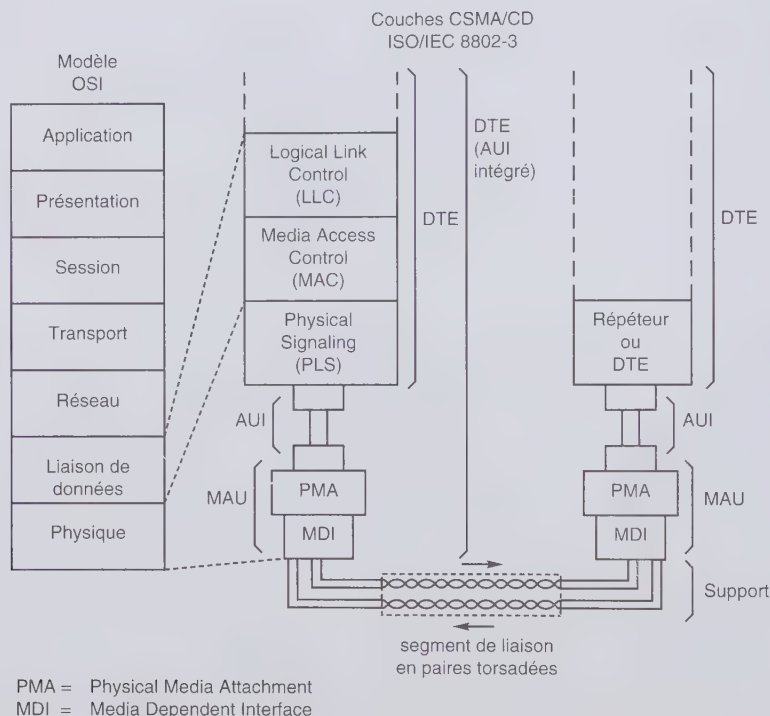
Tableau 3.4 : Affectation des broches dans un câble quatre paires

Broche	Signal
1	TD +
2	TD -
3	RD +
4	Non utilisé en 10BASE-T
5	Non utilisé en 10BASE-T
6	RD -
7	Non utilisé en 10BASE-T
8	Non utilisé en 10BASE-T



Figure 3.20

Les deux paires utilisées
dans les réseaux 10BASE-T.



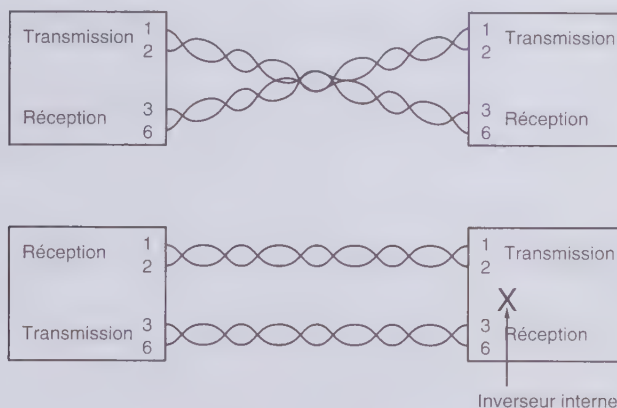
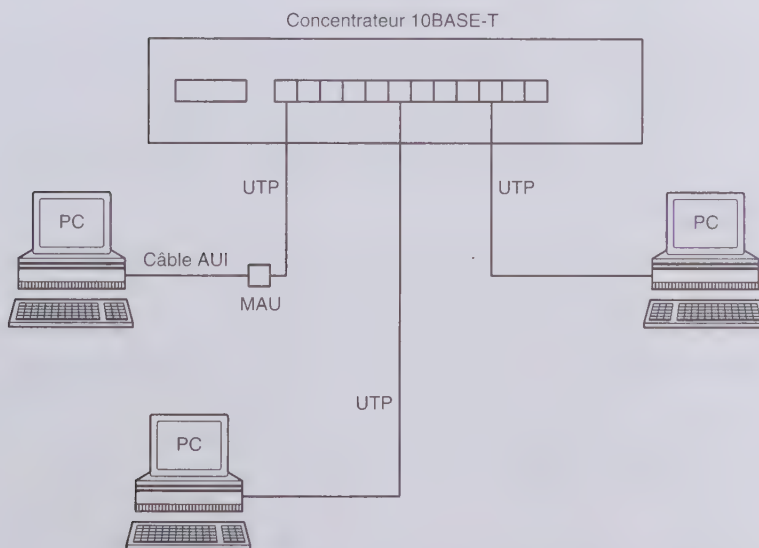
Quand deux cartes réseau sont reliées directement entre elles, le câble doit être croisé pour que la paire émission d'un port corresponde à la paire réception de l'autre. Dans les hubs, cette fonction de croisement est effectuée au niveau d'un port qui peut être soit droit soit croisé. Un port possédant cette fonction est marqué du symbole X. La Figure 3.21 montre les deux façons de croiser les ports.

La Figure 3.22 montre un hub 10BASE-T.

Le hub possède douze ports RJ-45. Si la carte réseau de la station possède un connecteur 10BASE-T, la connexion au hub est effectuée simplement par un câble en paires torsadées. Dans le cas de stations équipées de connecteurs AUI, un transceiver 10BASE-T est nécessaire pour connecter la sortie AUI de la carte au hub.

Un hub 10BASE-T joue le rôle d'un répéteur. Il effectue les fonctions suivantes :

- resynchronisation des paquets ;
- test de l'intégrité des liens ;
- autpartition pour déconnecter le port dans le cas de trente collisions consécutives.

Figure 3.21*Croisement des paires.***Figure 3.22***Un hub 10BASE-T.*

Les réseaux 10BASE-T CSMA/CD nécessitent une taille limitée à cause des délais de propagation (temps que met un signal pour aller vers une extrémité du réseau et en revenir). Les règles à observer quand on utilise plus d'un hub sont les suivantes :

- maximum de quatre hubs entre deux stations ;
- longueur maximale du câble de liaison entre le hub et une station de 100 mètres.



Câblage en paires torsadées non blindées (UTP)

Le câble UTP utilisé pour les transmissions de données a beaucoup évolué depuis le temps où il servait à la transmission de la voix. Ce câble peut maintenant être utilisé dans une version 100 Mbit/s d'Ethernet appelé 100BASE-T.

L'utilisation des câbles UTP dans le câblage des réseaux apporte simplification et réduction de coûts. Mais, s'il n'est pas réalisé correctement, il peut tout aussi bien compliquer les installations et augmenter les coûts de maintenance.

L'utilisation d'un câble UTP non blindé rend l'installation plus simple et moins coûteuse qu'une installation en câble UTP blindé. Cependant, comme il n'est pas blindé, ce type de câble se comporte facilement comme une antenne qui capte toutes les émissions radio. Aux fréquences de 10 à 100 Mbit/s, les câbles UTP peuvent perdre des signaux. L'inductance et la capacitance deviennent aussi des facteurs importants de ces fréquences.

L'*inductance* est causée par le champ électromagnétique qui entoure le câble quand des signaux haute fréquence le traversent. Il ressemble à un transformateur qui induit une tension dans un circuit secondaire. La *capacitance* provient de la présence d'un matériau isolant entre les deux conducteurs du câble. Les effets qui en résultent altèrent la qualité du signal et limite la distance entre deux périphériques reliés par un câble UTP.

Les torsades créées dans une paire "torsadée" réduisent l'inductance en créant un champ magnétique opposé qui annule ses effets. Pour cette raison, le nombre de torsades par centimètre est essentiel.

Les câbles plats, valables pour des débits inférieurs à 19,2 kbit/s n'ont pas de torsade. Ce type de câble est utilisé dans les installations téléphoniques. Dans un réseau, ils peuvent créer un véritable désastre.

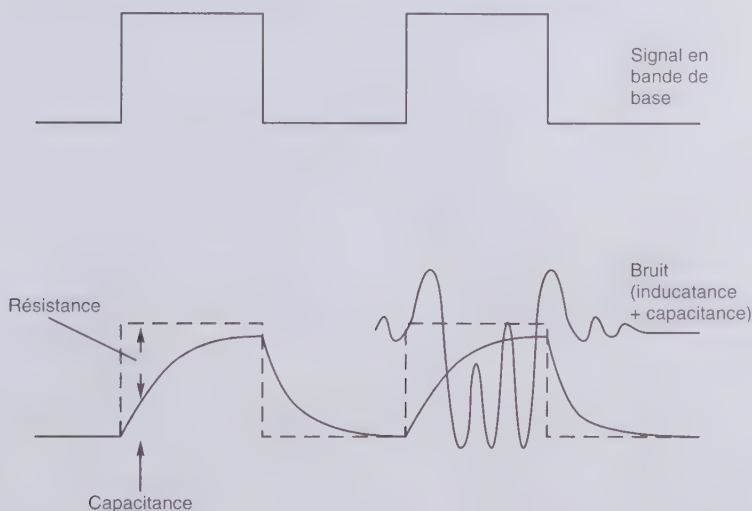
Les signaux qui se propagent dans les réseaux Ethernet ou Token Ring sont émis en *bande de base*, ce qui signifie que ces signaux sont des signaux carrés comme ceux qui sont représentés dans la Figure 3.23. Les fronts de montée et de descente sont pratiquement verticaux. La capacitance dans un câble rend les fronts de montée et de descente plus arrondis, tandis que la résistance atténue les signaux. L'inductance et la capacitance rendent les signaux vulnérables aux bruits, ce qui peut complètement modifier la forme de ces signaux (voir Figure 3.23).

Un autre facteur à considérer concerne la rapidité de changement des alternances des signaux. Mathématiquement parlant, le taux de changement du signal (ds/dt) est très grand pour un signal en bande de base. Si le signal est périodique, il peut être représenté comme la somme d'ondes sinusoïdales harmoniques de la fréquence

fondamentale. Cela signifie qu'un signal de 20 Mbit/s est, en réalité, la somme d'un signal de fréquence fondamentale égale à 20 MHz et d'harmoniques à 40, 60, 80, 100, 120 MHz, etc. Les plus hautes harmoniques sont d'amplitude faible. Toute cette démonstration est faite afin de comprendre que le câble doit pouvoir transmettre les harmoniques du signal, sinon il présentera une grande distorsion.

Figure 3.23

Distorsion des signaux.



Certains de ces facteurs peuvent être minimisés en utilisant des câbles en paires torsadées d'excellente qualité. Les caractéristiques d'un câble sont définies par l'atténuation qui s'exprime en décibels.

Cette atténuation détermine la longueur maximale des câbles pour une fréquence donnée. C'est pour cette raison que les câbles 10BASE-T doivent avoir une longueur maximale de 100 mètres.

Le type de gaine qui entoure le câble en paires torsadées affecte sa résistance au feu. Le PVC (chlorure de polyvinyle) est le plus utilisé, mais n'est pas résistant au feu. Un autre type, le TFETP (à base de Téflon) est plus résistant au feu. Il possède aussi une faible constante diélectrique. Plus la constante diélectrique est faible, plus la capacitance et donc la distorsion sont faibles. Grâce à ces caractéristiques, le TFETP permet des transmissions de meilleure qualité.

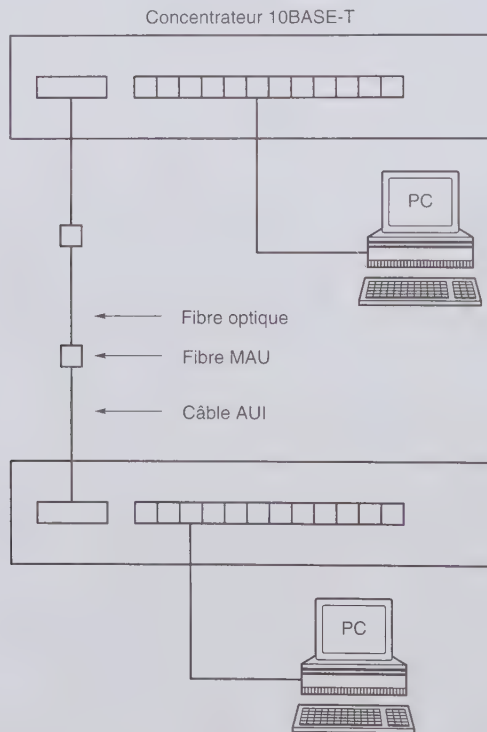


Combinaison de supports dans les réseaux Ethernet

Vous pouvez combiner différents médias dans la construction d'un réseau local Ethernet. Vous pouvez par exemple utiliser de la fibre optique ou du câble standard pour relier deux segments distants de plusieurs dizaines de mètres. Les Figures 3.24 et 3.25 montrent des exemples de réseaux associant plusieurs types de médias.

Figure 3.24

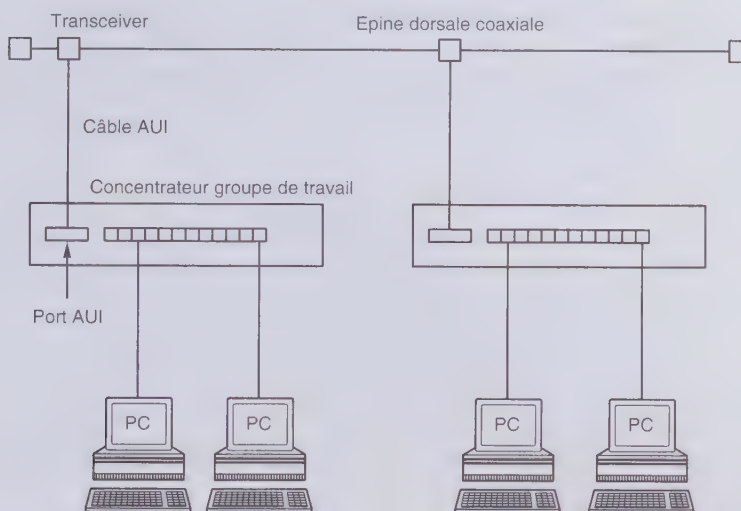
Réseau local formé de deux hubs reliés par une fibre optique.



Le Tableau 3.5 donne les caractéristiques de transmission des différents médias. Ce tableau permet de choisir le meilleur média compatible avec vos spécifications. Des appareils de tests, comme un réflectomètre, sont souvent nécessaires pour vérifier les caractéristiques du réseau.

Figure 3.25

Réseau local formé de deux hubs reliés par une épine dorsale en câble standard.

**Tableau 3.5 : Caractéristiques des différents médias Ethernet**

Type de média	Nombre maximal de connexions	Longueur maximale d'un segment	Vitesse de propagation (c = vitesse de la lumière)	Délai maximal par segment (nanosecondes)
Segment coaxial				
10BASE5	100	500	0,77c	2 165
10BASE2	30	185	0,65c	950
Segment de liaison				
FOIRL	2	1 000	0,66c	5 000
10BASE-T	2	100	0,59c	1 000
Câble transceiver				
AUI	DTE/1 MAU	50	0,65c	257

C = 300 000 000 mètres par seconde

FOIRL = Fiber Optic Inter Repeater Link (liaison interrépéteur en fibre optique)

MAU = Transceiver 10BASE-T



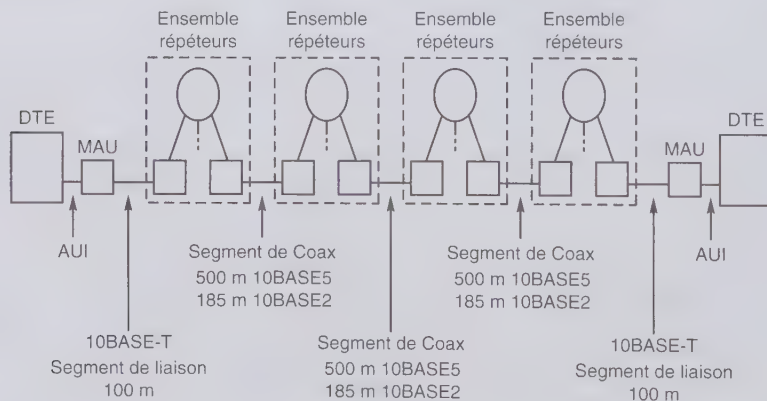
Les règles suivantes s'appliquent à des réseaux comprenant plusieurs types de médias :

1. Des répéteurs sont nécessaires pour tous les segments d'interconnexion.
2. Entre deux stations, il ne faut pas dépasser cinq segments, quatre répéteurs et deux MAU.
3. Si un réseau est formé de cinq segments et de quatre répéteurs, trois segments doivent être des câbles coaxiaux et les deux autres, des segments de liaison. Si le réseau comprend cinq segments, et si un lien FOIRL est utilisé comme segment de liaison, celui-ci ne doit pas dépasser 500 mètres.
4. Si un réseau comprend quatre segments et trois répéteurs, les segments FOIRL ne doivent pas dépasser 1 000 mètres chacun.

La deuxième règle est illustrée dans la Figure 3.26. Notez qu'elle n'indique pas le nombre de segments coaxiaux, ni le nombre de segments de liaison. La troisième règle clarifie ce problème.

Figure 3.26

*Chemin maximal
avec trois segments coaxiaux.*

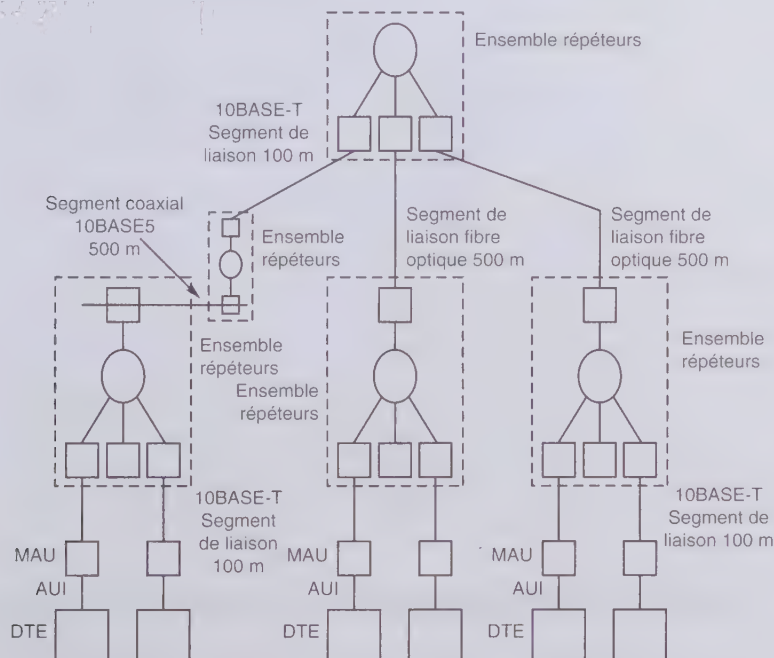


La troisième règle est illustrée par la Figure 3.27. Cette figure décrit un réseau Ethernet comprenant plusieurs médias de type coaxial, fibre optique et paire torsadée. Notez qu'il y a dans ce réseau cinq ensembles de répéteurs. Cela peut sembler en contradiction avec la règle d'un maximum de quatre répéteurs, mais, entre deux stations, il n'y a pas plus de quatre répéteurs. Ce réseau comprend, au total, dix segments : sept en paires torsadées, deux en fibre optique et un en coaxial. Cependant, il n'y a pas plus de cinq segments entre deux stations. De même, il y a un maximum d'un seul segment coaxial. Quand le segment coaxial est compris dans le chemin de transmission, les quatre autres segments sont des segments de

liaison : trois paires torsadées et une fibre optique. La longueur maximale d'un lien FOIRL est de 500 mètres puisqu'il y a cinq segments et quatre répéteurs dans le chemin de transmission. Cela suit la troisième règle. La longueur maximale de ce réseau est de 1 300 mètres, non compris les câbles de liaison AUI (*drop cables*).

Figure 3.27

*Chemin maximal
avec des segments coaxiaux,
des liens 10BASE-T
et des liens fibre optique.*



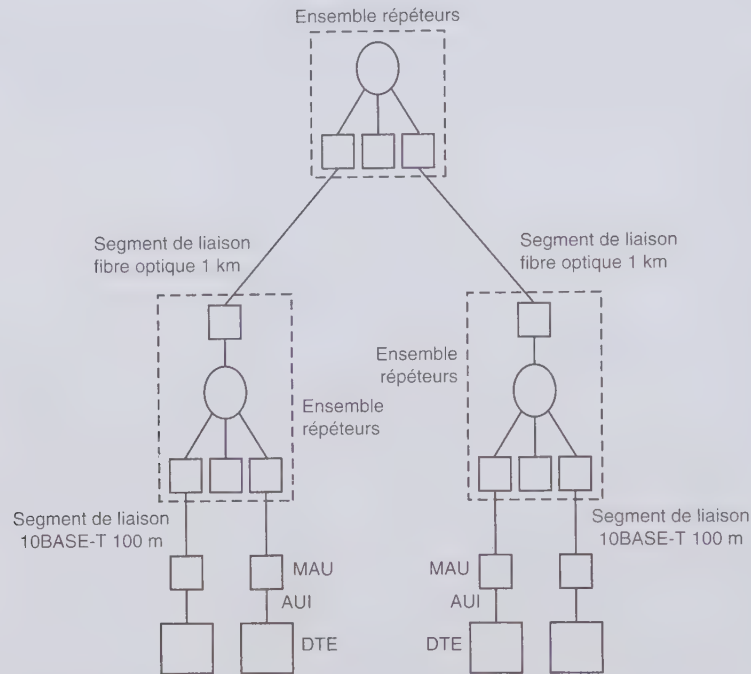
La quatrième règle est illustrée par la Figure 3.28. Ce réseau comprend trois répéteurs et six segments : quatre en paires torsadées et deux en fibre optique. Il n'existe pas de câble coaxial. Entre deux stations, le chemin de transmission comprend un maximum de quatre segments et de trois répéteurs. Les quatre segments consistent en deux en fibre optique et deux en paires torsadées. Chacun des liens FOIRL a une longueur maximale de 1 000 mètres. La longueur maximale de ce réseau est de 2 200 mètres sans compter les câbles de liaison AUI (*drop cables*).

Si vous envisagez d'étendre un réseau existant, avec différents types de câbles, utilisez les méthodes présentées dans les paragraphes suivants.



Figure 3.28

*Chemin maximal
avec trois répéteurs
et quatre segments de liaison.*



• Combinaison de câbles fin et épais dans un segment

Vous pouvez combiner des câbles fin et épais dans un seul segment, en utilisant le plus de câble fin possible. Ce câble est peu coûteux et plus facile à installer que le câble épais. La Figure 3.29 illustre un réseau construit avec des câbles coaxiaux fins et épais.

La longueur d'un câble mixte fin-épais est comprise entre 185 mètres et 500 mètres. La longueur minimale est de 185 mètres puisque tout segment inférieur doit être en coaxial fin, et le maximum est de 500 mètres puisque cette valeur représente la longueur maximale d'un segment en coaxial épais.

Pour calculer le nombre de mètres de câble fin nécessaires, utilisez la relation suivante :

$$\text{Longfin} = (500 - \text{longtotal}) / 3,28 \text{ mètres}$$

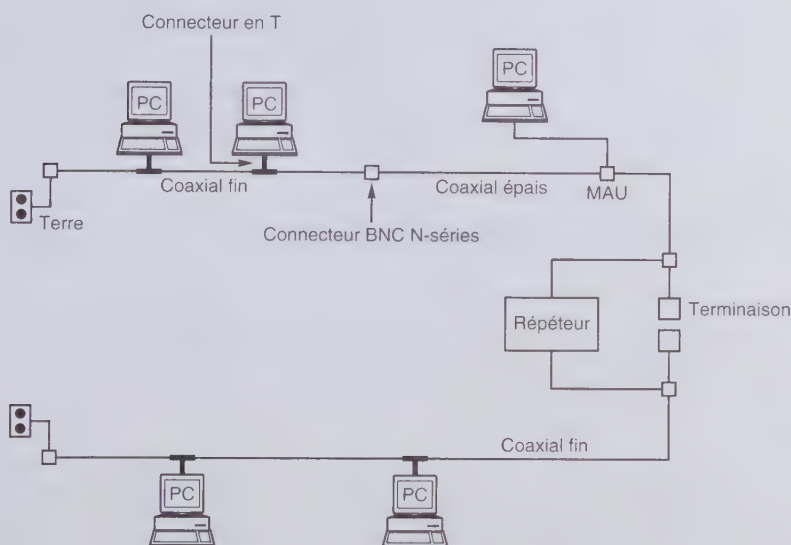
Par exemple, si vous désirez construire un réseau de 400 mètres, la longueur maximale de coaxial fin est donnée par :

$$(500 - 400) / 3,28 = 30,48 \text{ mètres}$$

Vous pouvez donc utiliser 30,48 mètres de coaxial fin et $400 - 30,48 = 369,52$ mètres de coaxial épais. La liaison entre le coaxial épais et le coaxial fin se fait par l'intermédiaire de connecteurs spéciaux N-series/BNC.

Figure 3.29

Réseau Ethernet combinant le câble coaxial fin et épais.



• Liaison de réseaux avec les standards 10BASE-FB et 10BASE-FL

Les standards 10BASE-FB et 10BASE-FL sont utilisés pour étendre des réseaux Ethernet existants. Le 10BASE-FB est utilisé pour les connexions Ethernet synchrones, et 10BASE-FL pour les connexions asynchrones. 10BASE-FB a été ratifié par le comité IEEE 802 en septembre 1993. Avant ces deux standards, on utilisait le lien FOIRL pour relier des segments sur des distances de 1 kilomètre. 10BASE-FB et 10BASE-FL permettent des liens jusqu'à deux kilomètres. La valeur de deux kilomètres est une spécification de norme. Les produits actuels peuvent permettre des liens jusqu'à quatre kilomètres.



Comprendre les trames dans les réseaux TCP/IP

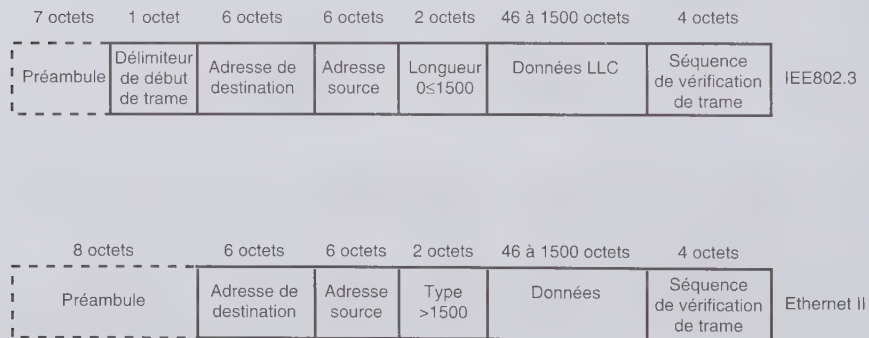
Dans les deux standards Ethernet-II et IEEE 802.3, les trames ont une taille minimale de 64 octets et une taille maximale de 1 518 octets. Les structures de ces deux types de trames sont différentes. Il est possible de configurer les nœuds d'un réseau soit avec la trame Ethernet-II, soit avec la trame IEEE 802.3. Mais un nœud configuré avec la trame Ethernet-II ne pourra communiquer avec un nœud configuré avec la trame IEEE 802.3. Les paragraphes suivants présentent les différences entre ces deux trames.

• La trame Ethernet-II

La trame Ethernet-II (voir Figure 3.30) commence avec un préambule de 8 octets, constitué d'octets 10101010 et finissant par 10101011. A 10 Mbit/s, ce préambule dure 6,4 microsecondes : ce temps est suffisant pour que la station qui le reçoit se synchronise et puisse recevoir la trame.

Figure 3.30

Comparaison des trames Ethernet II et 802.3.



Les champs d'adresse de destination (DA) et de source (SA) suivent le préambule. Chaque champ d'adresse comprend six octets. Les trois premiers octets représentent un code spécifique identifiant le constructeur de la carte réseau, et les trois autres octets sont définis par le constructeur. Cette assignation permet d'attribuer à une carte une adresse unique. Cette adresse est souvent contenue dans la ROM de la carte. Le bit de poids le plus faible (LSB) du premier octet est un bit de diffusion vers un groupe de stations (*multicast*). Sa valeur est 0 pour une adresse Ethernet. Une valeur 1 indique une adresse de diffusion. Les codes constructeurs ont été attribués par Xerox.

Le champ Type (appelé aussi EtherType) consacre deux octets à l'indication du type de données du champ Données. Par exemple, si la trame Ethernet transporte des données TCP/IP, la valeur de ce champ est 800 en hexadécimal. Le champ EtherType est utilisé par les pilotes réseau ou par la couche réseau pour démultiplexer les paquets de données en fonction de la pile de protocoles. Il permet à plusieurs piles de protocoles de tourner sur une seule carte Ethernet.

Le champ Données a une longueur variant de 46 à 1500 octets. La somme des autres champs, de taille fixe, est de 18 octets.

Le champ FCS (*Frame Check Sequence*, "séquence de vérification de trame") est généré après le champ de données. C'est un champ de 4 octets validant l'intégrité de la trame. Il utilise un CRC englobant les deux champs d'adresse : le champ Type et le champ Données.

• La trame IEEE 802.3

La trame 802.3 (voir Figure 3.30) commence avec un préambule de 7 octets, consistant en octets 10101010. Ce préambule a un temps de transmission de 5,6 microsecondes.

Le délimiteur de trame (*Start Frame Delimiter*, "délimiteur de début de trame") suit le préambule et contient la valeur 10101011. Notez que le préambule 802.3 et le champ SFD correspondent aux 8 octets du préambule Ethernet.

Les champs d'adresse de destination et de source suivent le SFD. Chaque champ d'adresse peut être de 6 ou de 2 octets, mais le champ de 6 octets reste le plus répandu. Les trois premiers octets représentent le code constructeur, et les trois autres sont attribués par le constructeur lui-même. Cette assignation est faite de façon que les cartes Ethernet et IEEE aient une unique adresse sur 6 octets. Cette adresse est contenue dans une ROM de la carte IEEE 802.3. Le LSB du premier octet représente le champ dénommé Individual/Group. Il est semblable au champ Physique/Multicast d'Ethernet. Le bit suivant est le champ Universal/Local et indique si l'adresse est globale ou locale.

Le champ suivant les champs d'adresse comprend 2 octets. Il indique la taille des données de la couche LLC. Cette taille varie de 46 à 1 500 octets, ce qui donne une taille maximale de la trame de 1 518 octets.



Le champ FCS, à la fin du champ de données, comprend un CRC de 32 bits portant sur les champs adresses, type et données. Il détecte les erreurs de transmission. Les trames défectueuses sont réémises.

• Différences entre Ethernet-II et IEEE 802.3

Il existe plusieurs différences entre les deux standards. Dans Ethernet-II, le champ Type indique le type de données, alors que dans IEEE 802.3, ce champ indique la longueur du champ des données. La longueur du champ dans les paquets Ethernet est assurée par les couches supérieures. Dans certains cas, la carte réseau peut déterminer la longueur de la trame à partir de la durée du signal et passe cette information aux couches supérieures. Pour IEEE 802.3, ce type d'information est fournie par la trame 802.2, qui fait partie du champ de données.

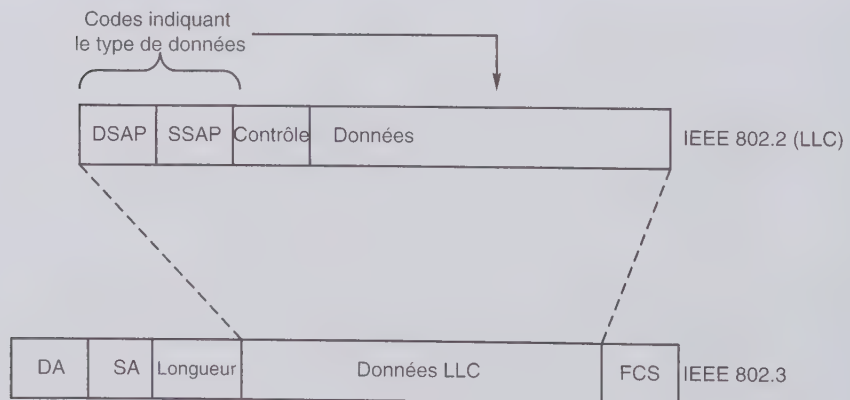
Ethernet ne donne aucune information sur le nombre d'octets de données, alors que les trames IEEE 802.3 contiennent un champ longueur indiquant si le champ de données contient un PAD (séquence de bourrage servant à remplir le champ de données pour obtenir au moins 46 octets).

• L'en-tête SNAP

Le format de la trame LLC est représenté dans la Figure 3.31.

Figure 3.31

Trame LLC dans l'IEEE 802.3.



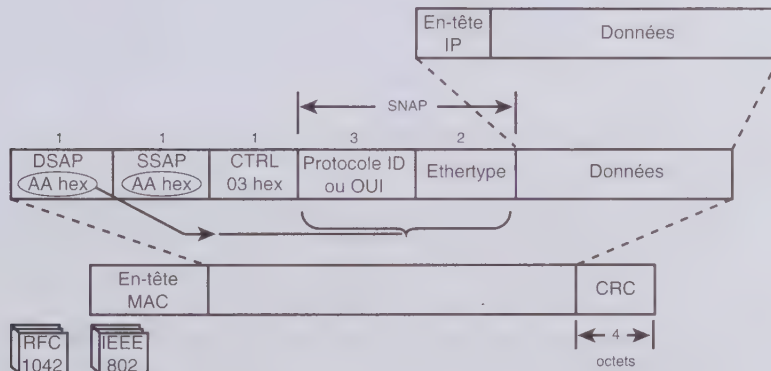
Normalement, les communications sont effectuées par le LCC 802.2 type 1. Nous avons déjà vu que le type 1 représentait une communication sans connexion, et que le type 2 était orienté connexion. Certains systèmes sur le même réseau IEEE 802

peuvent utiliser le type 2 après la vérification que ce type est bien supporté par les deux nœuds. Cette vérification est effectuée par le mécanisme d'identification de l'échange (XID), lequel consiste à envoyer des trames spéciales LLC afin de déterminer les paramètres et la transmission d'un autre nœud IEEE 802.2. Le type 1 est la méthode de transmission recommandée et doit être supportée par toutes les implémentations.

Un code hexadécimal AA pour SSAP et DSAP dans la couche LLC est réservé pour transmettre les paquets des couches supérieures, générés par des réseaux locaux non IEEE. Ce code spécial AA indique que le mécanisme SNAP (*Sub Net Access Protocol*) est utilisé. Le protocole SNAP décrit un en-tête qui est placé juste à côté de l'en-tête LLC, comme cela est représenté sur la Figure 3.32.

Figure 3.32

En-tête SNAP
avec une encapsulation IP.



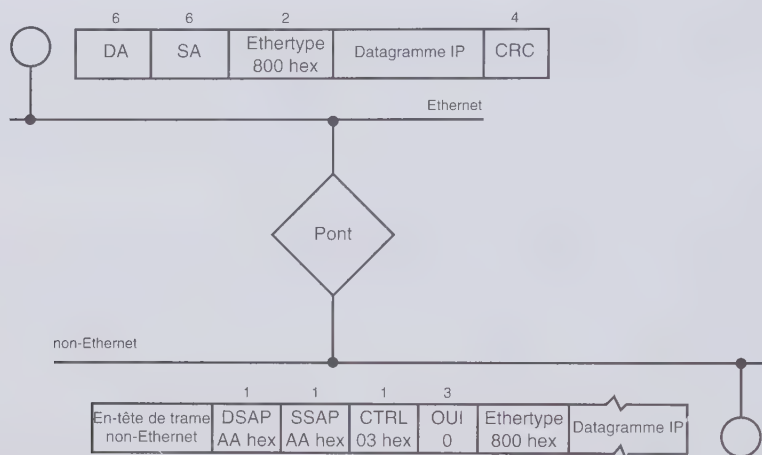
L'en-tête SNAP comprend cinq octets, à savoir trois octets OUI (*Organizational Unit Identifier*) et un champ EtherType de deux octets. Le champ OUI représente l'autorité responsable de l'encodage des octets qui suivent le OUI. Pour les réseaux TCP/IP, la valeur OUI est 0. Cela signifie qu'un champ EtherType de deux octets suit. En encodant le champ EtherType dans l'en-tête SNAP, il est possible de connecter des réseaux 802 non IEEE sans perdre les informations du champ EtherType. Ces réseaux non IEEE doivent utiliser des trames LLC avec des en-têtes SNAP.

Une description complète de SNAP se trouve dans la RFC 1042.



Figure 3.33

Connexion entre un réseau Ethernet et un réseau non Ethernet.



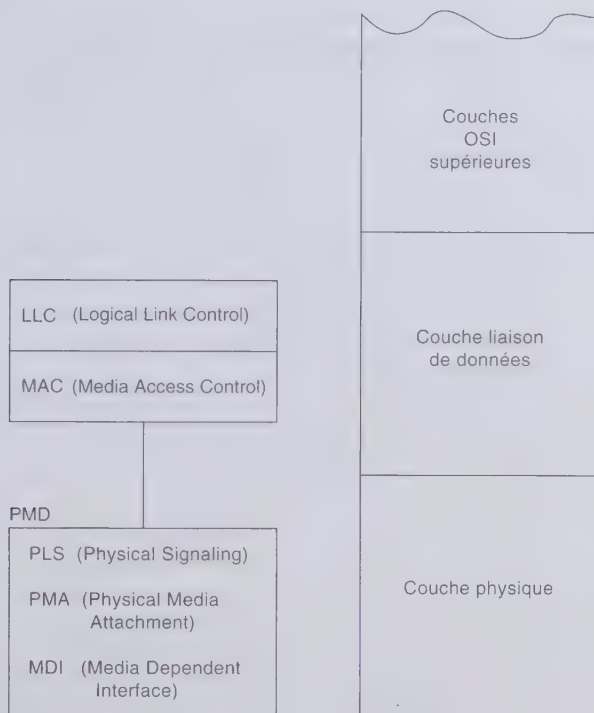
Fast Ethernet et TCP/IP

Comme les applications réseau deviennent de plus en plus complexes, les réseaux nécessitent une bande passante toujours plus grande. Le multimédia, la vidéo exigent des temps de réponse très courts et donc une bande passante très large. Bien que les réseaux Ethernet à 10 Mbit/s puissent être utilisés avec ces applications, des réseaux plus rapides sont de plus en plus nécessaires. Fast Ethernet et 100BASE-VG (décrits dans ce chapitre) sont des réseaux qui opèrent à 100 Mbit/s.

Avant de présenter les standards Ethernet à haut débit, voici un bref aperçu de ces standards par rapport à la base des réseaux Ethernet installés.

Les spécifications IEEE 802.3 définissaient un protocole de communication PMD (*Physical Media Dependent*), qui consistait en une couche LLC/MAC et en une couche PMD (voir Figure 3.34). Les couches LLC (*Logical Link Control*) et MAC (*Media Access Control*) correspondent à la couche 2 (couche liaison des données) du modèle OSI. La couche PMD correspond à la couche 1 (la couche physique) de ce même modèle.

La caractéristique particulière de l'IEEE 802.3 est la couche MAC. Elle est responsable du format de la trame Ethernet et du mécanisme CSMA/CD utilisé pour accéder à un support partagé. En rendant la couche MAC indépendante de la couche physique, cela permet aux spécifications de l'IEEE 802.3 d'autoriser plusieurs types de médias (voir Figure 3.35). Il permet aussi à la couche MAC d'être définie indépendamment du débit. À l'exception du temps minimal entre deux paquets (*interpacket gap*), tous les paramètres MAC sont exprimés en intervalles de temps par bit, ce qui signifie qu'il est possible d'utiliser Ethernet à des taux de transfert de données plus rapides que le débit original de 10 Mbit/s.

Figure 3.34*Spécifications de l'IEEE 802.3.*

Fast Ethernet opère à 100 Mbit/s et représente dix fois le débit d'un réseau Ethernet à 10 Mbit/s. Ce changement d'échelle est possible grâce à l'indépendance de la couche physique par rapport à la couche MAC. La couche physique de l'IEEE 802.3 est responsable de l'émission-réception du flux des bits de données, de l'encodage-décodage du flux de bits, de la détection de porteuse, de la détection des collisions et de l'interface mécanique et électrique avec le support physique. La distance maximale d'un réseau Fast Ethernet est de 250 mètres, à comparer aux 2 500 mètres d'un réseau Ethernet à 10 Mbit/s. La longueur maximale est dix fois moindre pour un débit dix fois plus grand. Dans un réseau CSMA/CD, une station située à une extrémité du réseau doit transmettre assez longtemps pour détecter une collision se produisant avec une station émettant à l'autre extrémité. A 10 Mbit/s et à une distance maximale de 2 500 mètres, le paquet doit être émis

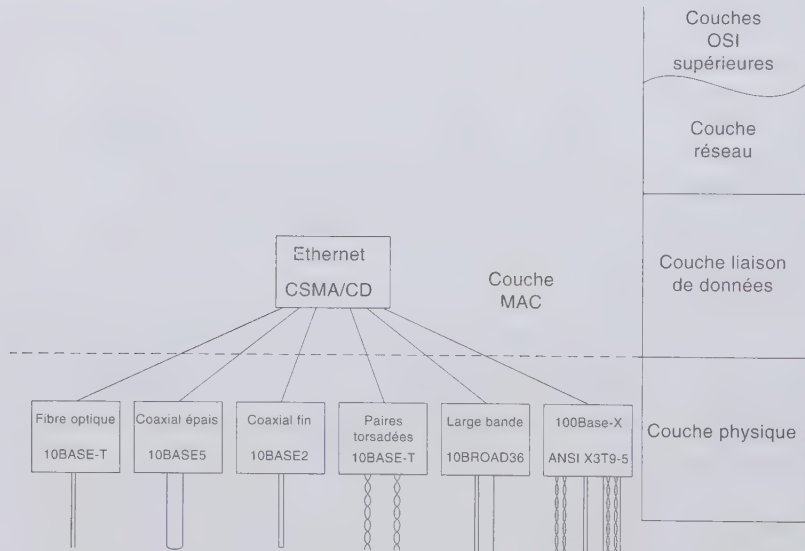


pendant 57,6 microsecondes pour correspondre à ces spécifications. Le nombre minimal de bits transmis se calcule de la façon suivante :

$$\begin{aligned}\text{Nombre minimal de bits} &= 10 \text{ Mbit/s} * 57,6 \text{ microsecondes} \\ &= 576 \text{ bits}\end{aligned}$$

Figure 3.35

*Support physique
de l'IEEE 802.3.*



Ces 576 bits comprennent le préambule qui contient 64 bits. Ainsi, la taille minimale d'une trame Ethernet /IEEE 802.3 est de $576 - 64 = 512$ bits (64 octets). Pour Fast Ethernet, la couche MAC est la même que pour Ethernet. La taille minimale d'une trame Fast Ethernet est aussi de 64 octets. Une collision doit être détectée pour 576 bits. Comme Fast Ethernet transmet dix fois plus vite qu'Ethernet, 576 bits seront transmis en 5,76 microsecondes. Ainsi, la distance maximale d'un réseau Fast Ethernet doit être dix fois moindre qu'un réseau Ethernet pour détecter une collision, soit $2\,500 / 10 = 250$ mètres.

En général, si le débit de transfert est de D Mbit/s pour un réseau CSMA/CD, la longueur du réseau pour maintenir la compatibilité avec le format et la taille de la trame Ethernet doit être : $L = 25\,000 / D$ mètres.

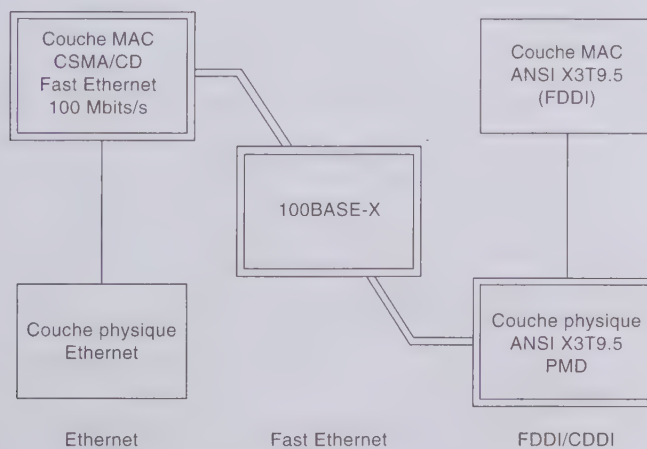
- Pour un réseau à 10 Mbit/s, la longueur est $25\,000 / 10 = 2500$ mètres.
- Pour un réseau à 100 Mbit/s, la longueur est $25\,000 / 100 = 250$ mètres.
- Pour un réseau hypothétique à 500 Mbit/s, la distance serait $25\,000 / 500 = 50$ mètres.

Une étude AT&T, à la fin des années 80, a montré que la majorité (plus de 90 %) des réseaux comprennent des stations situées à moins de 100 mètres d'un centre de concentration, soit un éloignement inférieur à la distance maximale d'un réseau Fast Ethernet.

Fast Ethernet supporte plusieurs types de médias. L'un d'eux, 100BASE-TX, a été proposé par Grand Junction Networks, Inc., l'un des premiers acteurs de Fast Ethernet. Le média est formé de deux paires UTP catégorie 5, IBM Type 1 (deux paires de paire torsadée blindée) et de fibre optique. 100BASE-TX utilise la couche ANSI Twisted Pair X3T9.5 PMD. L'ANSI a été accepté en 1992 comme une technologie fiable et économique pour transmettre un signal à 100 Mbit/s sur des paires torsadées pour CDDI (*Copper Distributed Data Interface*). CDDI est la version en paire torsadée de FDDI (*Fiber Distributed Data Interface*). La Figure 3.36 montre la relation entre les couches MAC de Fast Ethernet et de l'ANSI X3T9.5 PMD. 100BASE-TX combine la couche MAC d'Ethernet avec le standard ANSI X3T9.5 PMD. L'ANSI X3T9.5 supporte les câbles UTP catégorie 5, l'IBM Type 1 et la fibre optique.

Figure 3.36

Relation entre le Fast Ethernet, le 100BASE-TX et l'ANSI X3T9.5 PMD.



Le TP-PMD utilise un chiffrement de brouillage pour la sécurité et un encodage MLT-3. 100BASE-TX peut aussi utiliser la fibre FDDI multimode. La fibre PMD n'est pas codée et utilise un encodage 4B/5B et un signal NRZI (*Non Return to Zero Inverted*). Afin d'utiliser la fibre FDDI, la norme 100BASE-TX indique l'utilisation d'une sous-couche spéciale de convergence (voir Figure 3.37). Cette



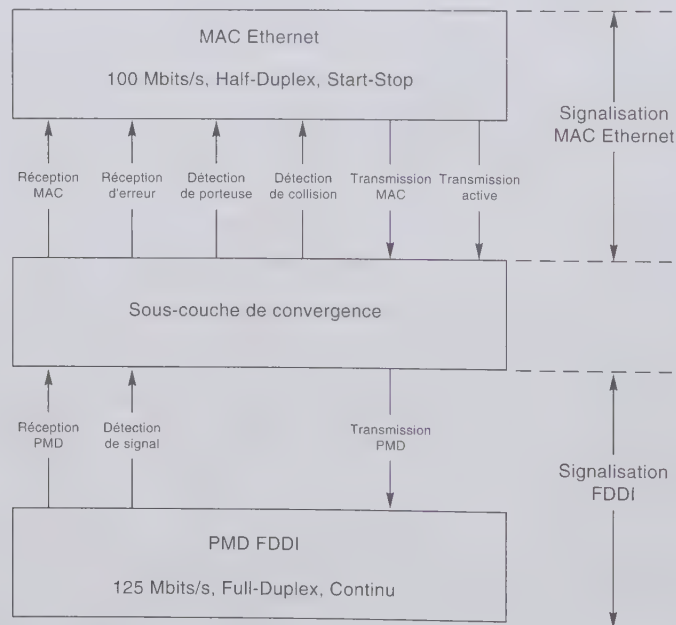
sous-couche de convergence convertit le signal continu de la fibre en signal half duplex pour MAC Ethernet.

100BASE-TX supporte le câble en paires torsadées non blindées comprenant deux paires UTP de catégorie 5. Les spécifications 10BASE-T indiquent deux paires UTP de catégorie 3. Pour ces sites qui doivent passer à Fast Ethernet, mais ne désirent pas modifier leur câblage en catégorie 5, une proposition appelée "4T+" a été définie par l'Alliance Fast Ethernet. 4T+ nécessite quatre paires de catégorie 3. La distance entre une station et un hub est de 100 mètres, mais la distance entre deux hubs ne doit pas dépasser 10 mètres. Ce type de réseau est appelé 100BASE-T4.

4T+ est un nouveau système de codage utilisant un huit binaire/six ternaire. Il est semblable au MLT-3 utilisé avec la catégorie 5, mais possède une largeur de bande limitée à 30 MHz. La Figure 3.38 montre l'utilisation des quatre paires. La paire 1 est utilisée pour indiquer les collisions et les paires 2, 3 et 4 pour transmettre les données.

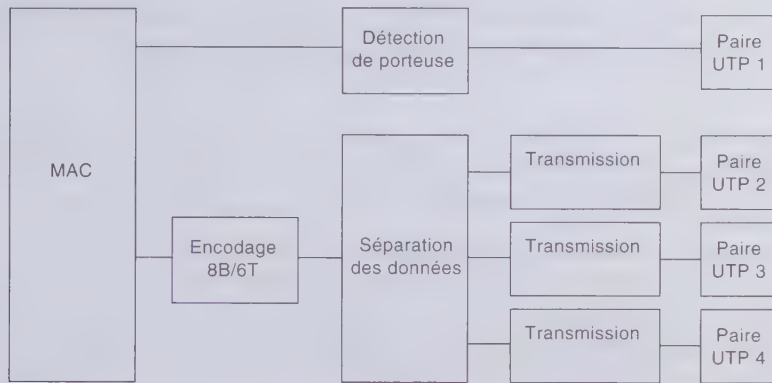
Figure 3.37

Sous-couche de convergence pour la fibre PMD dans le 100BASE-TX.



Un autre type de Fast Ethernet utilise la fibre optique pour les transmissions. Ce type est connu sous le nom 100BASE-TF.

Figure 3.38
Utilisation des quatre paires 4T+.



Les paramètres de Fast Ethernet sont indiqués dans le Tableau 3.6.

Tableau 3.6 : Paramètres de Fast Ethernet comparés à ceux d'Ethernet

Paramètre	Ethernet	Fast Ethernet
SlotTime	512 bit-times	512 bit-times
InterFrameGap	9,6 microsecondes	960 nanosecondes
AttemptLimit	16	16
BackoffLimit	10	10
Jamsize	32 bits	32 bits
MaxFrameSize	1 518 octets	1 518 octets
MinFrameSize	64 octets (512 bits)	64 octets (512 bits)
AddressSize	48 bits	48 bits

Les paquets TCP/IP sont envoyés sur Fast Ethernet, soit avec l'encapsulation dans une trame Ethernet II, ou le LLC avec l'encapsulation d'en-tête SNAP. La Figure 3.39 montre les deux méthodes.

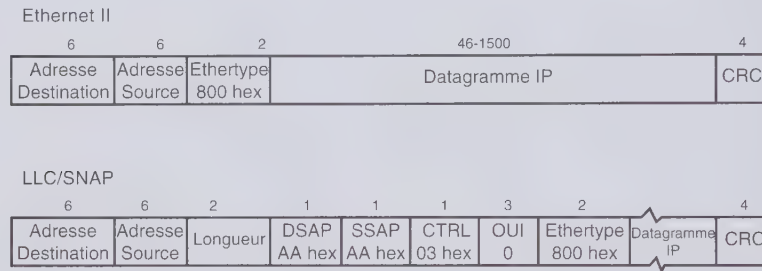
Une alternative à Fast Ethernet a été proposée par Hewlett-Packard et AT&T. Le système est fondé sur une méthode d'accès par priorité de demande dans laquelle un hub contrôle l'accès au réseau. Comme la méthode d'accès est différente du CSMA/CD, le système proposé par HP et AT&T ne peut faire partie d'Ethernet,



même si la trame a la même structure que celle d'Ethernet. Ce système constitue le sujet d'une étude par le comité IEEE 802.12.

Figure 3.39

Encapsulation de TCP/IP dans Fast Ethernet.

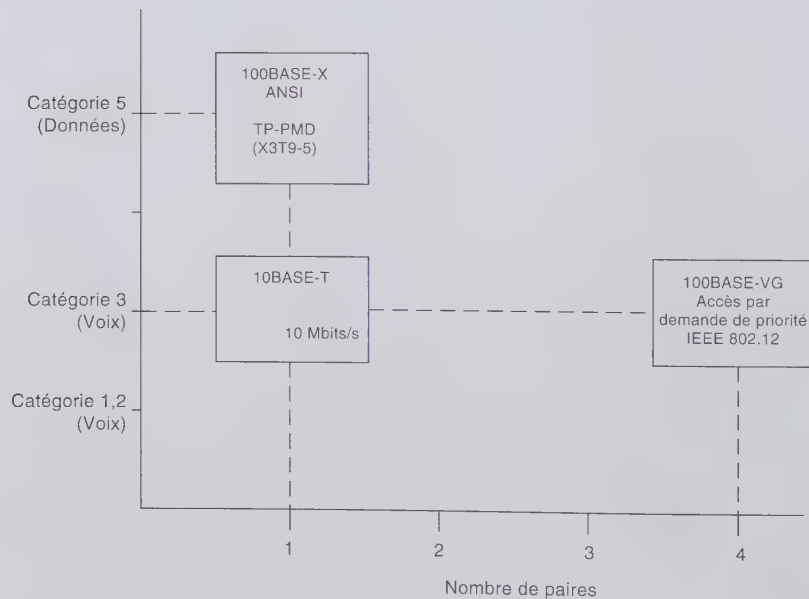


La proposition HP et AT&T utilise quatre paires de catégorie 3. La méthode d'accès est appelée 100BASE-VG, car elle utilise la catégorie 3 réservée à la voix (*voice grade*). Le 100BASE-VG peut aussi utiliser de la paire torsadée blindée de catégorie 5 ou de la fibre optique.

La Figure 3.40 résume les câblages du Fast Ethernet et de l'IEEE 802.12.

Figure 3.40

Type UTP utilisé par Fast Ethernet et l'IEEE 802.12.



La politique du numérotage : IEEE 802.13, IEEE 802.14, IEEE 802.30 ou IEEE 803.3 ?

Ce paragraphe risque de vous décontenancer par son étrangeté mais il va vous permettre de comprendre pourquoi les propositions de Fast Ethernet ont été appelées IEEE 802.14 puis 802.30.

Fast Ethernet fut considéré initialement comme faisant partie des spécifications IEEE 802.3. Le responsable du comité IEEE 802.3 sentit que l'utilisation de TP-PMD (*Twisted Pair-Physical Media Dependency*) apportait de tels changements à la couche MAC qu'il devenait nécessaire de créer un nouveau groupe. Le numéro suivant IEEE 802.12 aurait dû être IEEE 802.13. Mais c'était sans compter avec les superstitions qui rendent le nombre 13 maléfique. En conséquence, Fast Ethernet hérita du numéro IEEE 802.14 qui, plus tard, fut changé en IEEE 802.30. Le comité IEEE 802 fit remarquer que, si l'on permettait à Fast Ethernet de rester dans le comité 802.3, ses partisans seraient indûment avantagés par rapport à la proposition IEEE 802.12. A la réunion suivante du groupe 802, en 1993, le comité se résigna à garder le groupe IEEE 802.12. Le nouveau responsable ramena le groupe IEEE 802.30 sous l'égide du groupe IEEE 802.3 où il avait commencé.

IEEE 802.12 pour les réseaux TCP/IP

Le premier constructeur à proposer la méthode d'accès à demande de priorité de l'IEEE 802.12 fut Hewlett-Packard. D'autres constructeurs suivirent : AT&T, Microelectronics, IBM, Proteon et Ungermann-Bass. Ce standard est aussi connu sous les noms de 100BASE-VG et 100VG-AnyLAN.

Ce standard présente des performances meilleures que Fast Ethernet dans les réseaux chargés. Il permet aussi l'utilisation des câbles UTP de catégorie 3.

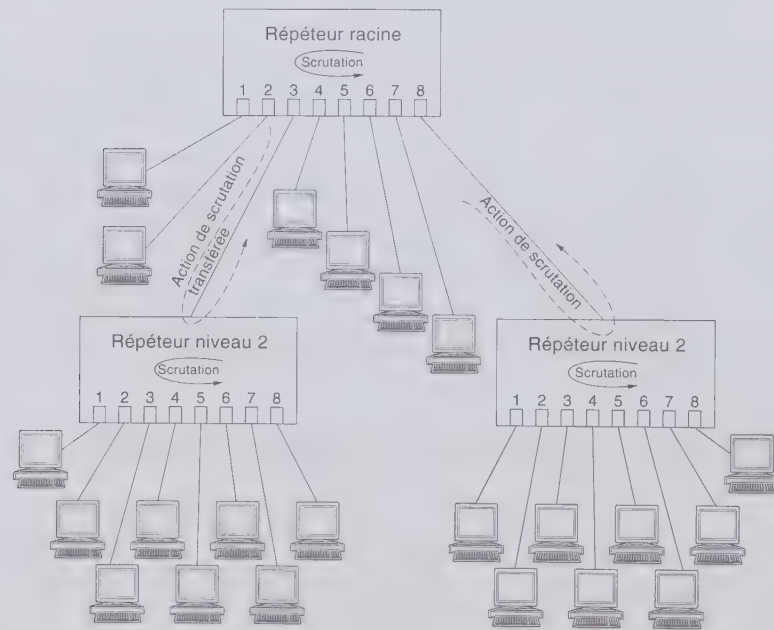
Dans les propositions IEEE 802.12, les réseaux s'articulent autour d'un répéteur central appelé le répéteur racine (*root repeater*). Ce répéteur scrute les ports dans un certain ordre pour déterminer si une station ou un autre hub répéteur relié à un port attend pour émettre des données. La Figure 3.41 montre un exemple de réseau IEEE 802.12. Le répéteur racine scrute le port 1, puis le port 2, puis le port 3. Ce port est relié à un autre répéteur de niveau 2 pour le distinguer du niveau 1 (répéteur racine). La scrutation du port 3 du répéteur racine entraîne la scrutation



des ports du répéteur de niveau 2. Si ce répéteur a huit ports, ceux-ci sont scrutés dans l'ordre 1, 2, 3, etc. A la fin de la scrutation des ports du niveau 2, celle-ci reprend sur les ports 4 et suivants du répéteur racine. Sur la Figure 3.41, un autre répéteur de niveau 2 se trouve connecté au port 8 du répéteur racine. Après avoir scruté le port 7, la scrutation reprend sur les ports du répéteur de niveau 2 relié au port 8 du répéteur racine. Les ports sont scrutés dans l'ordre 1, 2, 3, etc. Puis la scrutation reprend sur le répéteur racine en recommençant à partir du port 1.

Figure 3.41

*Scrutation des ports
dans un réseau IEEE 802.12.*



La Figure 3.41 ne montre que deux niveaux de répéteurs. L'IEEE 802.12 permet de mettre en cascade trois niveaux de répéteurs. La distance entre une station et un répéteur ne doit pas dépasser 100 mètres. Un réseau IEEE 802.12 peut s'étendre sur un maximum de 1200 mètres.

La scrutation des ports permet de déterminer avec précision le délai d'attente pour accéder au réseau. C'est très important pour les applications temps réel. Un système de priorité permet à des applications d'accéder immédiatement au réseau. Dans ce système, les requêtes d'accès sont classifiées en requêtes haute priorité et priorité normale. Plusieurs niveaux de priorités sont disponibles pour supporter le trafic en temps réel.

Dans ce système de priorités, il pourrait arriver qu'une station de basse priorité ne puisse jamais accéder au réseau. Pour éviter ce problème, les requêtes de priorité

normale qui attendent depuis plus de 250 millisecondes se voient attribuer une priorité d'accès au réseau. De plus, aucune station ne peut transmettre deux fois de suite quand une autre station attend au même niveau. Une haute priorité au niveau racine ne peut prendre la main sur une requête de plus basse priorité en cours. Cependant, à un répéteur de niveau plus bas (2 ou 3), une requête de haute priorité peut passer avant une requête de plus basse priorité, même si cette requête est en cours.

Le standard 802.12 utilise indifféremment des trames Ethernet ou Token Ring. Dans sa forme standard, la norme 802.12 exige que tous les hubs d'un domaine de priorité supportent la trame Ethernet ou Token Ring, mais pas les deux. Une extension du standard admet qu'un hub supporte les deux types de trames.

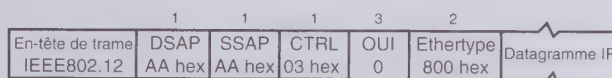
La méthode d'encodage de l'IEEE 802.12 est le quartet. Cette méthode utilise quatre codes séquentiels envoyés sur quatre paires UTP de catégorie 3. L'encodage est le NRZ (*Non Return to Zero*) avec un schéma d'encodage 5B/6B. L'encodage NRZ minimise les changements de signaux pour envoyer les données. L'encodage 5B/6B utilise 6 bits pour coder 2^5 valeurs possibles.

La proposition 802.12 contient une option permettant d'utiliser deux paires de catégorie 5, deux paires de STP, ou un câble en fibre optique. Avec ces options de câblage, les quatre codes séquentiels qui doivent normalement être envoyés sur quatre paires de catégorie 3 sont multiplexés sur deux paires de catégorie 5 ou STP, ou sur une seule fibre.

L'IEEE 802.12 utilise les trames LLC. Les paquets TCP/IP sont encapsulés dans une trame LLC avec un en-tête SNAP (voir Figure 3.42). Nécessaires pour SNAP, les valeurs DSAP et SSAP sont AA hex, et la valeur OUI vaut zéro.

Figure 3.42

Trame TCP/IP pour IEEE 802.12.



Commutateurs Ethernet

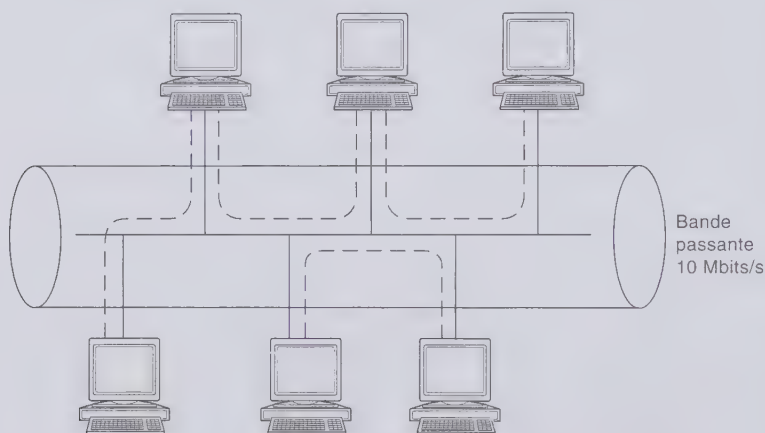
Une évolution intéressante d'Ethernet concerne les commutateurs. Normalement, Ethernet utilise un média partagé par tous les périphériques connectés au réseau. La bande passante maximale d'un réseau Ethernet est de 10 Mbit/s. Cette bande doit être partagée par tous les périphériques du réseau (voir Figure 3.43). Si cette bande était partagée également entre n périphériques, chacun d'eux aurait $10 / n$ Mbit/s de disponibles. Pour $n = 1024$ (nombre maximum de stations d'un réseau Ethernet),



chaque station aurait 10 kbit/s. Mais toutes les stations ne transmettent pas en même temps sur le réseau. En supposant que 10 stations ($n = 10$) transmettent sur le réseau, cela donnerait 1 Mbit/s pour chacune d'elles.

Figure 3.43

Bande passante d'un réseau Ethernet.



Bande passante Ethernet partagée entre toutes les stations

Pour augmenter la bande passante disponible pour chaque station, des constructeurs proposent des hubs appelés *commutateurs Ethernet* (voir Figure 3.44) dans lesquels chaque port du hub dispose de 10 Mbit/s. Chaque port du commutateur doit stocker les adresses des périphériques connectés à ce port. Il doit donc disposer de suffisamment de mémoire associée à chaque port pour stocker toutes ces adresses.

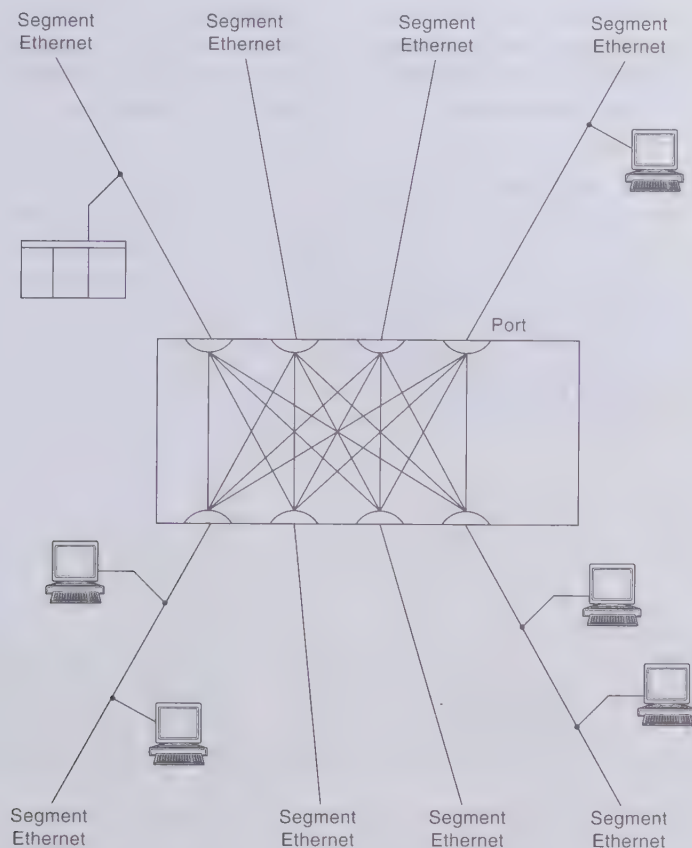
Le commutateur utilise ces adresses pour envoyer un paquet reçu sur un port au port concerné par l'adresse de destination située dans la trame du paquet. Chaque segment Ethernet connecté à un port du commutateur est indépendant des autres segments et les collisions d'un segment n'affectent pas les autres segments. On peut dire que le commutateur agit comme un pont. La capacité totale d'un réseau utilisant un commutateur est multipliée par le nombre de ports du commutateur. Si un commutateur contient n ports, sa capacité est $10 \times n$ Mbit/s.

Ainsi, avec un switch de 12 ports, le réseau peut distribuer les données à un taux de 120 Mbit/s.

Certains constructeurs de commutateurs proposent plusieurs options. En voici les principales :

- stockage des paquets avant l'envoi (mode *store and forward*) ;
- envoi après vérification d'en-tête ;
- étude du fragment avant l'envoi.

Figure 3.44
Commutateur Ethernet.



Dans la première option, toutes les données d'un paquet sont reçues par le commutateur qui les stocke avant de les envoyer. Cela permet de vérifier les erreurs de CRC. Les paquets ayant un CRC invalide sont rejetés. Cette méthode, la plus fiable, est aussi celle qui prend le plus de temps car il faut attendre de recevoir tout le paquet avant de l'envoyer.

Dans la deuxième méthode, la décision d'envoi est prise dès que le commutateur a reçu l'en-tête de l'adresse. Cela réduit le temps d'attente. Mais le CRC n'étant pas testé, on court le risque d'envoyer un paquet avec un CRC invalide. Ce mode est le plus rapide et le moins fiable. Les erreurs peuvent encore être détectées par les couches plus hautes au niveau des stations.

Figure 3.44



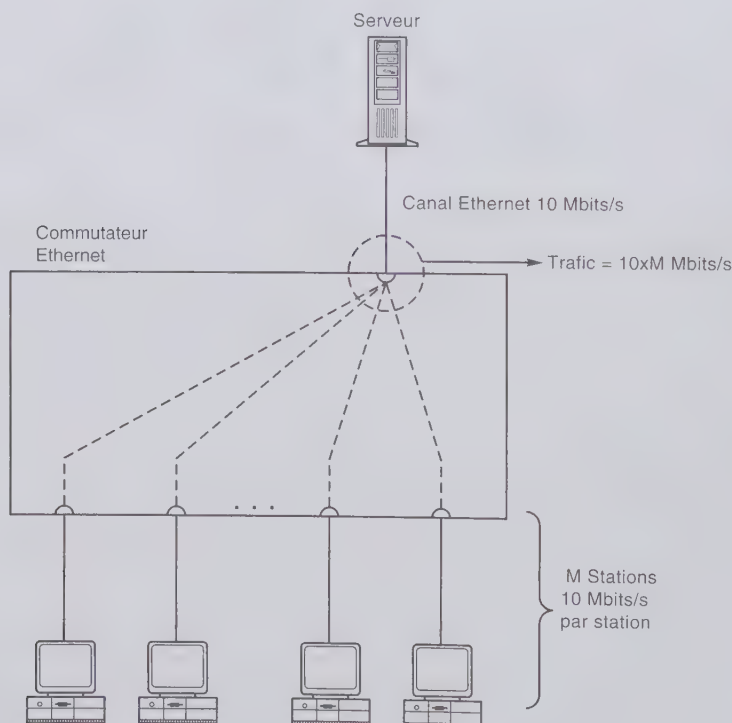
La troisième méthode est une combinaison des deux autres. Dans ce mode, le switch détermine la longueur du paquet et vérifie qu'il possède au moins 64 octets, la taille minimale d'un paquet Ethernet. Les paquets de taille inférieure à 64 octets sont appelés des fragments et proviennent de collisions ou d'un mauvais fonctionnement d'une carte réseau.

Une autre caractéristique des commutateurs concerne le nombre de stations connectées à un port. Un segment Ethernet connecté à un port profite d'une grande bande passante fournie par le commutateur. Cela évite les congestions du segment.

Souvent, le trafic le plus important d'un réseau se situe au niveau des serveurs. Les ports rapides (à 100 Mbit/s, par exemple) servent à connecter ces serveurs. Une autre solution consiste à installer une carte full-duplex à 20 Mbit/s dans le serveur et à relier deux ports à une seule carte. Cela permet de doubler la bande passante entre le serveur et le commutateur.

Figure 3.45

*Problème de trafic à un port
d'un commutateur Ethernet.*



Certains commutateurs possèdent un mode d'adresses sécurisées qui permet d'entrer dans une table les adresses des stations associées à un port et donc d'interdire l'utilisation du commutateur à certaines stations. Dans les commutateurs où une seule station est connectée à un port, un dispositif permet de désactiver le port lorsqu'une station non autorisée est détectée.

Dans la plupart des réseaux, le goulet d'étranglement principal se trouve au niveau du serveur. Imaginez un commutateur fournissant 10 Mbit/s à chaque station connectée à un port et un serveur connecté à l'un de ces ports. Un serveur auquel accéderont n stations recevra $10 \times n$ Mbit/s de données (voir Figure 3.45). Si $n = 10$, cela représente $10 \times 10 = 100$ Mbit/s de données. Un commutateur Ethernet standard ne permet pas d'apporter une solution à ce problème. Il faut un commutateur Ethernet possédant des ports capables de gérer 100 Mbit/s. Pour 12 ports à 10 Mbit/s et un port à 100 Mbit/s, cela donne une bande passante de :

$$(12 \times 10) + (1 \times 100) = 220 \text{ Mbit/s.}$$

TCP/IP sur les réseaux Token Ring

Le Tableau 3.1 indique qu'il existe deux normes basées sur le passage d'un jeton (*token*). Le réseau Token Ring que vous rencontrerez le plus souvent répond à la norme IEEE 802.5. Il est souvent associé à IBM car IBM a été le premier à le proposer. Les réseaux IEEE 802.4 utilisent aussi la méthode du jeton, mais sur un bus, et non pas sur un anneau. Ce paragraphe présente la norme Token Ring IEEE 802.5.

INFO



La méthode du bus à jeton simule l'anneau à jeton sur un réseau en bus. Il est rarement utilisé dans des réseaux commerciaux. Les réseaux Token Bus ont été conçus pour le protocole MAP (Manufacturing Automation Protocol) qui est un réseau historique. ARCnet est aussi basé sur un bus à jeton mais il tend à disparaître. C'est pour ces raisons que le bus à jeton ne sera pas présenté dans ce chapitre.

• Fonctionnement des réseaux Token Ring

Les réseaux Token Ring sont utilisés depuis plusieurs années. Les réseaux à anneau découlent des liens point à point et, en tant que tels, ne sont pas des réseaux de diffusion comme Ethernet. Ils peuvent être considérés comme des émissions séquentielles avec des liens formant un cercle. La technique de l'anneau est digitale alors que celle des réseaux Ethernet est analogique. Une

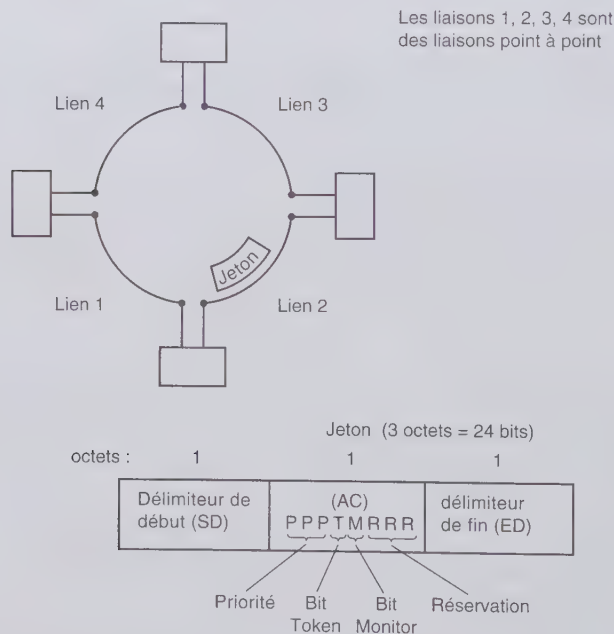


caractéristique essentielle des réseaux à anneau se rapporte à leur temps de réponse déterministe, même sous une charge importante. Un *temps de réponse déterministe* signifie que l'on peut prédire le délai maximum d'attente d'un nœud avant d'envoyer des données. Au contraire, les réseaux Ethernet sont non déterministes car on ne peut jamais savoir au bout de combien de temps une station pourra émettre sur le réseau.

La Figure 3.46 illustre le fonctionnement d'un réseau Token Ring. Un tel réseau peut être vu comme une suite de liens point à point. Chaque station joue le rôle d'un répéteur fournissant l'amplification nécessaire aux signaux. Les liens peuvent être n'importe quel média comme le câble coaxial, les paires torsadées ou la fibre optique. La paire torsadée reste un média de choix pour les réseaux IBM Token Ring. La fibre optique est souvent utilisée pour étendre un réseau Token Ring sur de longues distances.

Figure 3.46

Fonctionnement d'un réseau Token Ring.



Le *jeton* est une trame spéciale utilisée par les stations pour accéder au réseau. Les stations qui désirent envoyer des données attendent le jeton pour le faire. Un bit de la trame indique si le jeton est utilisé. Après qu'une station a acquis un jeton libre, elle le modifie en jeton occupé et envoie les données. Pour un fonctionnement correct, le jeton doit circuler sans arrêt même s'il n'y a aucune activité sur l'anneau. Le jeton comporte 24 bits (3 octets). L'anneau doit donc avoir assez de

temps de latence pour contenir 24 bits. Si le débit sur l'anneau est de 4 Mbit/s, l'anneau doit avoir un temps de latence de $24/4 \text{ Mbit/s} = 6 \text{ microsecondes}$. Six microsecondes semblent un délai très court mais imaginez une paire torsadée dans laquelle la vitesse de propagation est 0,59 fois la vitesse de la lumière. La formule suivante permet de calculer la taille de l'anneau avec un temps de latence de six microsecondes :

$$\begin{aligned}
 \text{Taille de l'anneau} &= \text{Temps de latence} \times \text{Vitesse de propagation sur le média} \\
 &= 0,0000006 \times 0,59 \times 3 \times 100\,000\,000 \text{ mètres} \\
 &= 1062 \text{ mètres} \\
 &= 1,062 \text{ kilomètre}
 \end{aligned}$$

Ainsi, la taille minimale de l'anneau doit être de 1 km ! C'est très long, surtout si vous désirez installer plusieurs stations dans une pièce. Pour cette raison, une station spéciale désignée comme moniteur actif ajoute un buffer de 24 bits sur l'anneau.

Dans un fonctionnement normal de l'anneau, une station peut être éteinte. Mais que se passe-t-il si le jeton arrive dans une station éteinte ? En fait, les réseaux Token Ring sont câblés en étoile avec un hub central. Chaque connexion entre le hub et une station est contrôlée par un relais situé dans le hub. Sur la Figure 3.47, les relais sont maintenus ouverts par les stations en fonctionnement. Quand une station est éteinte, le relais se ferme et shunte la connexion.

Une station d'un réseau Token Ring opère dans l'un des quatre modes suivants :

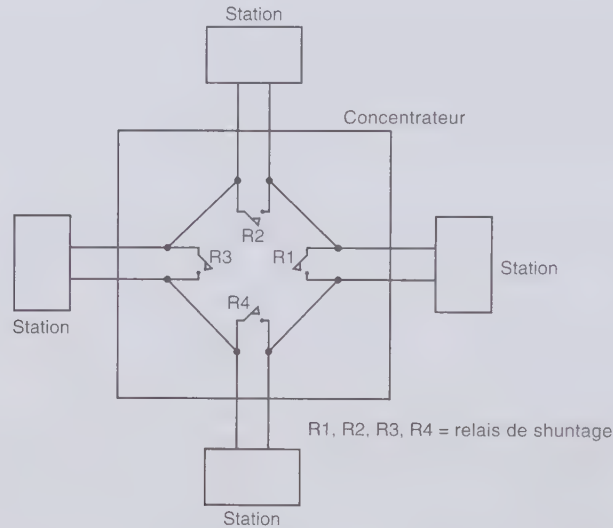
- mode de transmission ;
- mode d'écoute ;
- mode shunt ;
- mode de réception.

La Figure 3.48 montre quatre stations opérant chacune dans un mode différent. La station A transmet. Pour entrer dans ce mode, elle saisit un jeton libre. Le jeton possède un bit appelé le bit *T*. La valeur de ce bit détermine l'état du jeton. Il est libre si le bit *T* est à 1. La station qui émet modifie ce bit. Elle le met à 0, ce qui indique qu'il est occupé, et transmet ses données. La station A envoie ses données à la station D. Le champ d'adresse de destination contient l'adresse de D, tandis que le champ d'adresse source contient l'adresse de A.



Figure 3.47

Système de shunt des stations
sur un réseau Token Ring.



La station B est en mode d'écoute. Elle vérifie le champ d'adresse de destination pour savoir si les données lui sont destinées. Comme la trame est envoyée à la station D, la station B reste en mode d'écoute. Dans ce mode, la station copie les données entrantes dans le câble de sortie.

La station C a été éteinte et se trouve en mode shunté. Les données passent directement sans traverser la station C.

La station D examine le champ d'adresse de destination et découvre que les données lui sont destinées. Elle entre en mode de réception. Dans ce mode, la trame est copiée dans la mémoire de la station et envoyée aussi sur l'anneau.

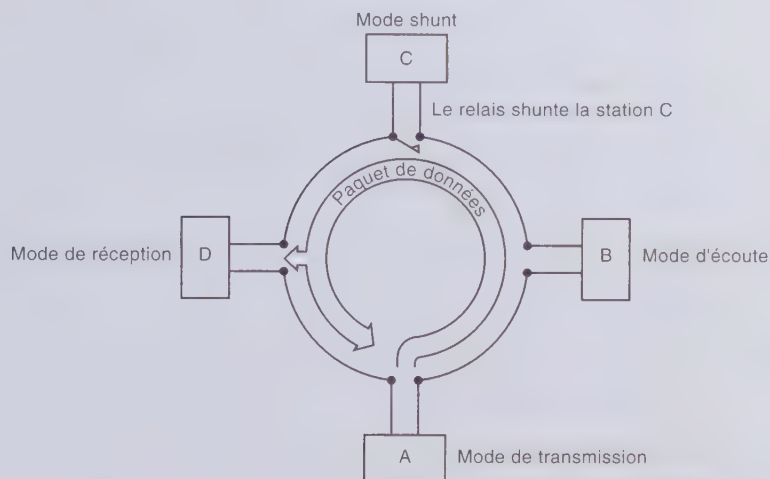
Plusieurs drapeaux appelés *drapeaux d'état de la trame* (*Frame Status*) sont modifiés pour indiquer la réception des données. La station A reçoit la trame de données envoyée et examine les drapeaux. Ces drapeaux jouent le rôle d'accusés de réception. La station émettrice vérifie ainsi si la trame a été correctement reçue par la station D. Voici la signification des drapeaux :

- Drapeau A : Adresse reconnue
- Drapeau C : Trame recopiée
- Drapeau E : Erreur

Le Drapeau E est lu et activé par chaque station. Les Drapeaux A et C sont activés uniquement par la station réceptrice. Le Tableau 3.7 définit ces drapeaux. Il convient de noter (voir Figure 3.48) que les drapeaux A et C sont dupliqués dans le champ de statut de la trame (FS) pour pallier le fait qu'ils ne sont pas protégés par le FCS.

Figure 3.48

Modes des stations d'un réseau
Token Ring.



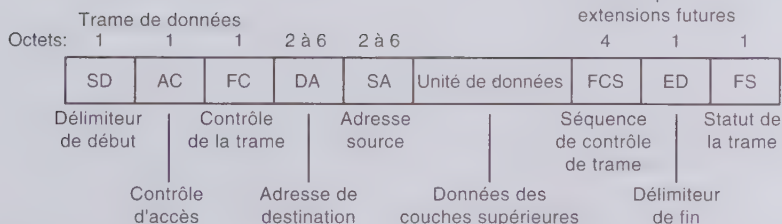
FS = Statut de la trame

A C r r A C r r

A = Adresse reconnue

C = Trame copiée

r = Réserve pour des
extensions futures

**Tableau 3.7 : Drapeaux d'état de la trame**

Drapeaux de la trame	Valeur	Signification
A	1	Adresse reconnue
A	0	Adresse non reconnue
C	1	Trame recopiée avec succès
C	0	Trame non recopiée
E	1	Trame altérée (erreur de CRC)
E	0	Trame correcte



Les combinaisons possibles de ces drapeaux sont les suivantes :

- AC = 00 implique que l'adresse n'a pas été reconnue et donc que la copie n'a pu avoir lieu.
- AC = 10 indique que la station existe mais que la trame n'a pas été copiée. Si E = 1, une trame altérée a été reçue. Si E = 0, la trame, pour une raison inconnue, n'a pas été copiée.
- AC = 11 indique que la station existe et que la trame a été copiée. Si E = 1 et AC = 11, la trame a bien été copiée, mais une erreur s'est produite après la copie de la trame.

La seule combinaison illégale est AC = 01 qui indiquerait que la station n'est pas reconnue mais qu'un utilisateur a copié la trame. En d'autres termes, une station a copié la trame illégalement. C'est une condition impropre.

Quand les bits envoyés par la station A y reviennent, ils sont supprimés de l'anneau.

Si une station est éteinte avant que la trame ne lui revienne, cette trame circule indéfiniment sur le réseau. Il existe d'autres scénarios semblables qui dérèglent les opérations normales sur l'anneau. Un jeton peut être détruit par des bruits parasites sur l'anneau et les stations peuvent attendre le jeton indéfiniment. Le système Token Ring contient des mécanismes automatiques pour résoudre ce genre de problèmes. Ces situations sont détectées et gérées par des trames de contrôle appelées Trames MAC. C'est l'une des raisons pour lesquelles les opérations de l'IEEE 802.5 sont plus complexes que celles de l'IEEE 802.3. Les sections suivantes présentent quelques-uns de ces mécanismes automatiques.

Bien que toutes les stations semblent à égalité sur le réseau, certaines le sont plus que d'autres. L'une des stations est appelée le *moniteur actif*. Il existe, dans le jeton, un bit *moniteur* (*M-bit*) qui est mis à 0 par la station qui transmet. Le moniteur actif examine le M-bit et le met à 1 s'il est à 0. Si le moniteur actif trouve une valeur du M-bit à 1, c'est que la trame de données a circulé au moins une fois dans l'anneau. Cela peut provenir d'un problème lié à la station émettrice qui n'a pu supprimer la trame de l'anneau.

Si le jeton est perdu à cause des bruits sur l'anneau, le moniteur actif génère un nouveau jeton. Le moniteur actif garde la trace du temps de rotation d'un jeton et le supprime si le temps dépasse un certain seuil. Pour les petits réseaux Token Ring, la valeur de ce temps de rotation est de huit microsecondes. Cette valeur peut augmenter en cas de trafic plus important sur le réseau.

Le moniteur actif n'est pas une station comportant un matériel réseau spécifique. Toute station du réseau peut devenir un moniteur actif. Dès qu'une station est devenue moniteur actif, les autres se trouvent en attente de le devenir. Le choix de la station moniteur est effectué au cours d'une procédure d'initialisation de

l'anneau. Si le moniteur actif a un problème, c'est l'une des stations en attente qui prend le relais.

Quand aucune trame de données ne circule sur l'anneau, le moniteur actif envoie une trame MAC de présence (AMP, *Active Monitor Present*). Cette trame est envoyée à intervalles réguliers de 7 secondes. Les autres stations en attente d'être moniteur actif se signalent en envoyant des trames MAC de moniteur en attente (SMP, *Stand-by Monitor Present*). Les moniteurs en attente détectent les trames AMP et en déduisent que le moniteur actif est en fonctionnement. Si celui-ci n'envoie pas la trame au bon moment, l'un des moniteurs en attente prend le rôle de moniteur actif. Le moniteur en attente qui détecte le problème du moniteur actif envoie une demande sur le réseau sous la forme d'une trame de demande de jeton (CL_TK, *Claim Token*). Le moniteur en attente arrête l'envoi de ces trames si l'une des conditions suivantes est réalisée :

- **Une autre trame CL_TK est reçue et l'adresse de la source de cette trame est de valeur plus élevée que l'adresse de la trame envoyée par la station.** Si deux ou plusieurs stations envoient un CL_TK, la station dont l'adresse a la valeur la plus élevée devient le moniteur actif.
- **Une trame de balise (BCN, *beacox*) est reçue.** Cette trame est envoyée en cas de problème grave comme une coupure de l'anneau. La trame BCN est utilisée pour localiser et isoler le problème. Dans ce cas, l'anneau doit être réparé avant de décider qui doit devenir moniteur actif.
- **Une trame de purge est reçue (PRG).** Cette trame est envoyée à la fin d'une procédure CL_TK par la station qui devient moniteur actif. Cela signifie que la course a été gagnée par une station et qu'il n'est plus nécessaire de continuer.

Dans tous les cas précédents, les moniteurs en attente le restent. Si une station reçoit la trame CL_TK qu'elle a générée, elle devient le moniteur actif et envoie une trame PRG pour informer les autres stations qu'un nouveau moniteur actif a été élu. A ce moment, le nouveau moniteur actif ajoute un buffer de temps de latence de 24 bits sur l'anneau et commence la gestion du réseau.

Avant de rejoindre un anneau, une nouvelle station envoie une trame *Duplicate Adress Test* (DAT) dans la procédure d'initialisation. La trame DAT est envoyée avec sa propre adresse dans le champ DA. Si une autre station répond avec les bits AC mis à 11, cela signifie que l'autre station a la même adresse. La nouvelle station renvoie un code d'état. Le logiciel de gestion du réseau détecte ce code et envoie un message d'erreur approprié.



Une autre caractéristique de l'IEEE 802.5 concerne le mécanisme de priorité d'accès. Le jeton comporte deux champs :

- un champ priorité ;
- un champ de réservation.

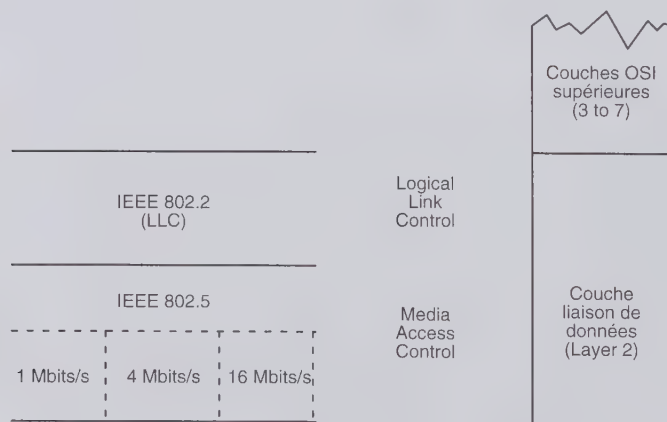
Chaque champ contient trois bits. Huit priorités sont définies (de 0 à 7). Le champ de réservation est mis à 0 par la station qui transmet. Si une station demande une priorité d'accès, elle peut placer une valeur correspondant à la priorité dans le champ Réservation. Après réception de la trame envoyée par la station émettrice, elle copie la valeur du champ de réservation dans le champ de priorité du nouveau jeton qui est généré. Le jeton possède maintenant la priorité demandée. Seules les stations possédant une priorité supérieure ou égale peuvent accéder à ce jeton.

• Options Token Ring

L'IEEE 802.5 indique les options d'implémentation de Token Ring à 1 Mbit/s, 4 Mbit/s et 16 Mbit/s (voir Figure 3.49). Le débit de 1 Mbit/s utilise le câblage UTP. Initialement, les débits 4 et 16 Mbit/s utilisaient le câblage STP. Mais des industriels souhaitaient utiliser le 4 et 16 Mbit/s sur des câbles UTP. Plusieurs produits supportent le câble UTP pour 4 et 16 Mbit/s. Pendant longtemps, IBM ne supportait pas la version 16 Mbit/s sur câblage UTP. Actuellement, IBM et Synoptics Communications ont proposé à l'IEEE 802.5 un standard 16 Mbit/s sur câblage UTP.

Figure 3.49

Options IEEE 802.5
pour Token Ring.



Les stations à 16 Mbit/s n'attendent pas le retour de la trame de données pour injecter un jeton sur le réseau. Le mécanisme (*early token release*) permet à des trames de données de sources différentes d'être transmises en même temps sur un réseau Token Ring.

• Éléments d'un réseau local Token Ring

La carte réseau Token Ring possède un connecteur à neuf broches pour le câblage en paires torsadées blindées. Il existe aussi une option pour le câblage en paires torsadées non blindées avec des connecteurs RJ-45.

Le concentrateur appelé *Multistation Access Unit* (MAU) – à ne pas confondre avec le *Media Attached Unit* (MAU) de l'IEEE 802.3 – permet de connecter huit stations. Les deux ports extrêmes, appelés *Ring In* (RI) et *Ring Out* (RO), servent uniquement à connecter d'autres MAU. Il existe des MAU, dits intelligents, contenant des possibilités de gestion de réseau.

Un dispositif (*MAU Setup Aid*) est utilisé pour tester chaque port en vérifiant le mécanisme de relais.

Le câble IBM Token Ring est de type 1. Le type 6 peut aussi être utilisé mais il ne doit pas être recommandé comme câble de connexion entre les MAU. L'une des extrémités de ce câble est connectée à la carte réseau et l'autre extrémité comporte un connecteur hermaphrodite à connecter à un port du MAU IBM 8228.

INFO



Bien que le câble IBM type 3 soit le même que celui de catégorie 3 présenté dans ce livre, les types 1, 2, 6 et 9 sont des câbles IBM et ne doivent pas être confondus avec les types de classifications UTP.

La façon de connecter des MAU et des stations est représentée Figure 3.50.

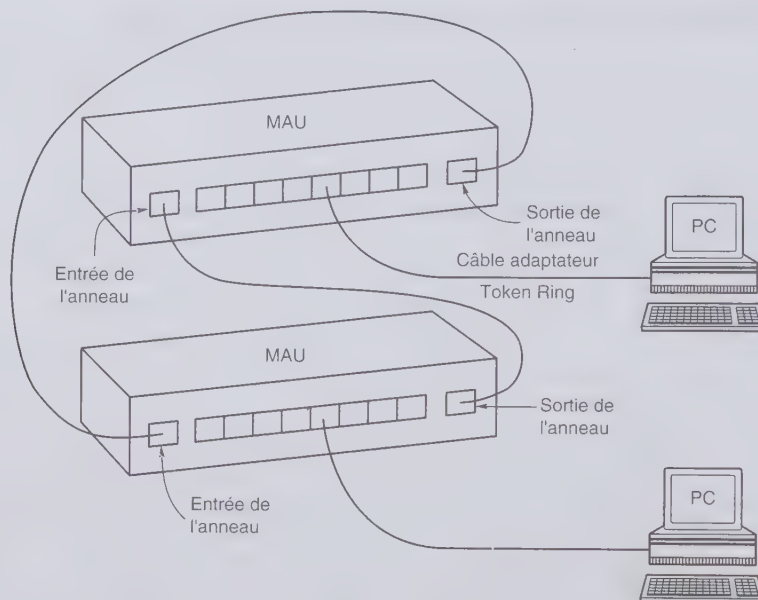
• Règles de câblage de l'IEEE 802.5

Le Tableau 3.8 résume les règles de câblage de l'IEEE 802.5.

Tableau 3.8 : Règles de câblage de l'IEEE 802.5

Paramètres Token Ring	Type 1, 2	Type 3
Nombre maximal de périphériques par anneau	260	96
Vitesse	16 Mbit/s	4 Mbit/s
Distance station MAU (Un seul MAU)	300 mètres	100 mètres
Distance station MAU (plusieurs MAU)	100 mètres	45 mètres
Nombre maximal de MAU par réseau	12	2
Distance entre deux MAU	200 mètres	120 mètres



Figure 3.50*Eléments d'un réseau
Token Ring.*

Câblages types 1 et 2

Bien qu'un maximum de 260 stations puissent être connectées sur un câblage de type 1 ou 2, il vaut mieux ne pas dépasser la valeur 100. Si vous utilisez un MAU à huit ports, en reliant douze MAU, vous pourrez connecter $8 \times 12 = 96$ stations sur le réseau.

Pour les câbles de type 1 ou 2, la distance maximale entre une station et un MAU est de 300 mètres (en utilisant un seul MAU). Cette distance tombe à 100 mètres avec plusieurs MAU. En pratique, il vaut mieux choisir la distance de 100 mètres, même s'il n'existe qu'un seul MAU. Un réseau a tendance à s'étendre et, si vous augmentez le nombre de stations, il faudra ajouter des MAU et donc réduire les longueurs existantes.

La distance maximale entre deux MAU est de 200 mètres pour les types 1 et 2.

Câblage type 3

Pour le câblage type 3, la distance maximale entre une station et un MAU est de 100 mètres pour un seul MAU, mais cette distance tombe à 45 mètres pour plusieurs MAU. Là encore, il vaut mieux choisir des distances de 45 mètres, même avec un seul MAU pour assurer l'avenir.

La distance maximale entre deux MAU est de 120 mètres pour le câblage de type 3.

Conseils pour le câblage Token Ring

Voici quelques conseils pour un câblage Token Ring :

1. Les stations situées à moins de 2,5 mètres d'un MAU peuvent être connectées avec des câbles de 2,5 mètres de long.
2. Les stations situées à plus de 2,5 mètres d'un MAU peuvent être connectées avec des câbles d'extension.
3. Pour former un anneau, reliez le connecteur RO d'un MAU au connecteur RI du second MAU. Continuez ainsi jusqu'au dernier. Reliez le connecteur RO du dernier MAU au connecteur RI du premier.
4. Vous ne pouvez pas connecter de station aux prises RO et RI d'un MAU. Ces deux connecteurs ne servent qu'à l'interconnexion de MAU.
5. Les câbles type 6 IBM ne doivent pas être épissés.
6. Les câbles type 6 IBM ne doivent pas être utilisés dans des conduits ou dans des cheminées. Utilisez pour cela le type 9 IBM.

• Pour éviter les problèmes dans les réseaux Token Ring TCP/IP

Pour qu'un réseau fonctionne correctement, la circulation du jeton doit être maintenu. Tout problème qui stoppe le flux de données sur le câble entraîne un dysfonctionnement du réseau. De même, comme les données passent de station en station, une station présentant un problème quelconque peut faire tomber tout le réseau. Pour ces raisons, le dépannage d'un réseau Token Ring peut représenter un véritable challenge !

INFO



Les réseaux Token Ring modernes, spécialement ceux d'IBM, contiennent des logiciels de diagnostic internes qui facilitent les dépannages et détectent les défauts des câbles. Des logiciels de dépannage des réseaux Token Ring sont disponibles dans le commerce.

• La taille d'un Token Ring

Les réseaux Token Ring utilisant le câblage STP peuvent accepter jusqu'à 260 stations. Augmenter ce nombre provoquerait des problèmes pouvant faire tomber le réseau. Le nombre minimal de stations est de deux.

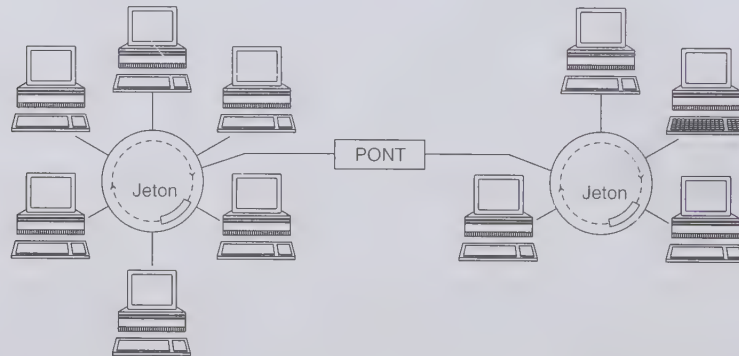


Pour déterminer le nombre de stations d'un réseau, il faut construire un anneau assez grand pour un travail confortable, mais suffisamment petit pour rendre les dépannages simples. Dans un petit réseau, il est plus facile d'isoler une station posant problème ou de détecter ce qui ne va pas sur l'anneau.

Pour construire un réseau Token Ring de taille raisonnable, utilisez un *pont* reliant deux anneaux. Dans la Figure 3.51, deux anneaux sont reliés par un pont. Chaque anneau possède son propre jeton. Le jeton reste sur l'anneau physique auquel il appartient. De cette façon, il est possible de construire des réseaux complexes en anneaux dans lesquels les anneaux opèrent indépendamment les uns des autres mais peuvent communiquer entre eux par les ponts. Le pont participe au niveau de la couche MAC.

Figure 3.51

Deux anneaux reliés par un pont.



• Diagnostics internes pour les réseaux Token Ring

Les spécifications de l'IEEE 802.5 pour les réseaux Token Ring spécifient des diagnostics internes utilisés par les stations pour isoler les problèmes. Plusieurs de ces diagnostics se réfèrent au concept de domaine en erreur.

Un *domaine en erreur* est représenté par une section d'anneau (voir Figure 3.52) contenant les trois éléments suivants :

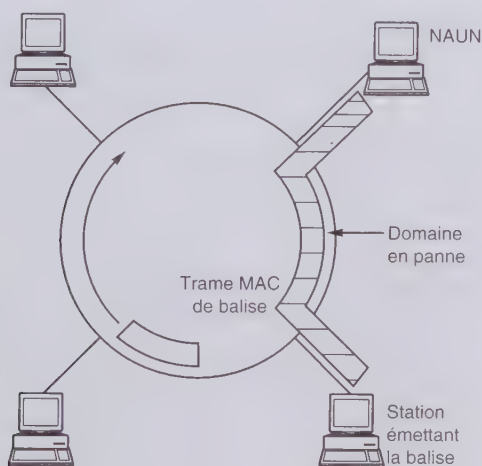
- une station transmettant une trame de balise BCN ;
- une station *Nearest Active Upstream Neighbor* (NAUN) proche de la station émettant la balise ;
- le câble qui relie la station émettant la balise et la NAUN

Comme le flux de données dans un anneau est unidirectionnel, chaque station a une station voisine en amont. Cette station en amont est la station à partir de laquelle la trame de données est reçue. Elle est appelée NAUN. Le mot *Active* implique que la NAUN peut ne pas être la première station physiquement en

amont. La NAUN pour chaque station est indiquée dans le processus de notification de voisin activé par le moniteur actif.

Quand une station découvre une erreur causée par un dysfonctionnement matériel, il envoie une trame MAC de balise qui contient son adresse de carte d'interface réseau NIC (*Network Interface Card*), la station NAUN et le type de balise contenant des informations sur la nature du problème. Le contenu de la balise définit essentiellement quel est le domaine en faute. Le type de balise aide au diagnostic de l'erreur. Un type de balise indique, par exemple, une erreur de câble. La trame de balise est diffusée à toutes les stations. Toutes les stations entrent dans le mode de répétition de la balise. Elles copient la trame autour de l'anneau. La trame de balise est générée par la station qui détecte l'erreur. Après que la NAUN a copié la trame de balise huit fois, elle se supprime de l'anneau et lance une série d'autotests tels que DAT (*Duplicate Address Test*) et *Lobe Media Test*. Dans ce dernier test, la carte d'interface NIC transmet une trame *MAC Lobe Test* vers le port du MAU pour vérifier qu'il reçoit correctement la trame. Si ce n'est pas le cas, le *lobe* (le câble entre le NIC et le MAU) est suspect.

Figure 3.52
Domaine d'erreur Token Ring.



Si la station NAUN tombe en panne, elle se déconnecte de l'anneau. Comme la station NAUN, qui était la cause du problème, ne se trouve plus sur l'anneau, la station qui émet la balise reçoit la trame BCN qu'elle a envoyée. Dans ce cas, la station arrête l'envoi de la balise et retransmet le jeton sur l'anneau. L'anneau fonctionne à nouveau.

P 30 7



Si la station NAUN passe les tests correctement, elle se réinsère dans l'anneau. Après un certain temps, la station qui émet la balise suppose que la station NAUN a passé les tests correctement. Le problème se trouve sans doute dans la station qui émet la balise. Cette station se teste à son tour. Si elle est en erreur, elle se déconnecte de l'anneau. Le moniteur actif initialise alors une récupération de l'anneau par émission d'une trame de purge suivie par une trame de contrôle. Si la station qui émet la balise passe ce test, elle se réinsère dans l'anneau. Si les conditions d'erreur persistent, le problème nécessite un dépannage manuel. Le problème relève sans doute du câblage (y compris le MAU) qui relie la station qui émet la balise à la station NAUN.

• Trame TCP/IP dans les réseaux IEEE 802.5

Le standard IEEE 802.5 utilise la trame LLC pour encapsuler les données des couches supérieures. L'en-tête SNAP est utilisé pour envoyer les données TCP/IP (voir Figure 3.53).

Figure 3.53

Données TCP/IP dans une trame Token Ring.

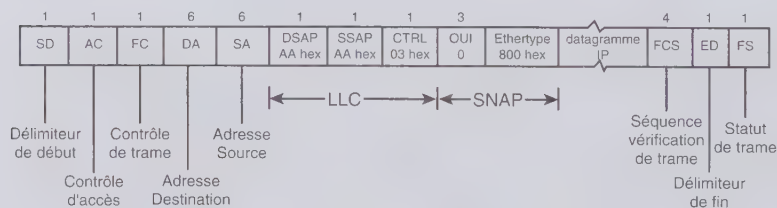
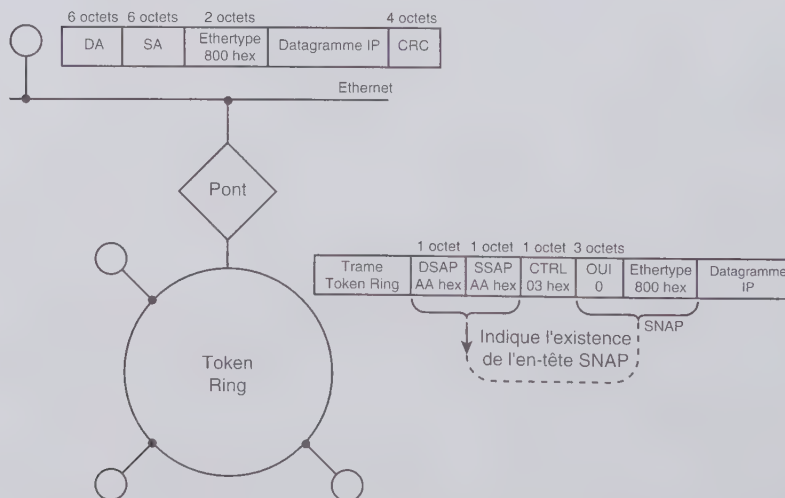


Figure 3.54

Trame TCP/IP sur des réseaux Token Ring et Ethernet.



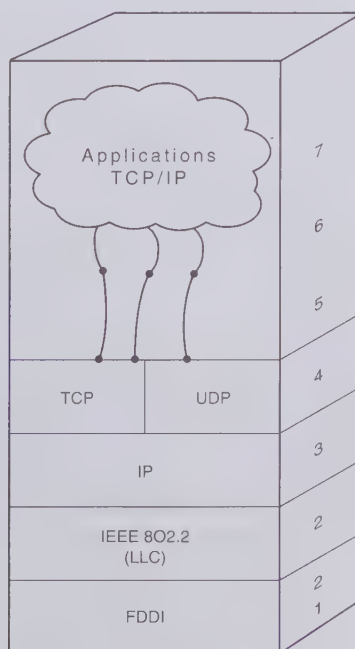
La valeur du champ EtherType dans l'en-tête SNAP est 800 (hexadécimal). C'est la valeur standard assignée pour l'encapsulation des données TCP/IP dans des trames Ethernet. En utilisant l'en-tête SNAP, pour lequel les valeurs LSAP et DSAP sont AA (hexadécimal), la valeur du champ EtherType est préservée dans le cas où des réseaux Ethernet et Token Ring seraient connectés. C'est le cas de la Figure 3.54 dans laquelle les trames Ethernet et Token Ring transportent des données TCP/IP.

TCP/IP sur les réseaux FDDI

FDDI (*Fiber Distributed Data Interface*) est considéré comme un réseau local à haut débit (100 Mbit/s). Comme il peut s'étendre sur une distance de 100 km, il possède par là même les caractéristiques d'un réseau étendu dans la mesure où il peut interconnecter des réseaux locaux. FDDI recouvre les couches 1 et 2 du modèle OSI et peut être utilisé pour fournir les services IEEE 802.2 ou LLC aux couches supérieures (voir Figure 3.55).

Figure 3.55

FDDI recouvre les couches 1 et 2 du modèle OSI.



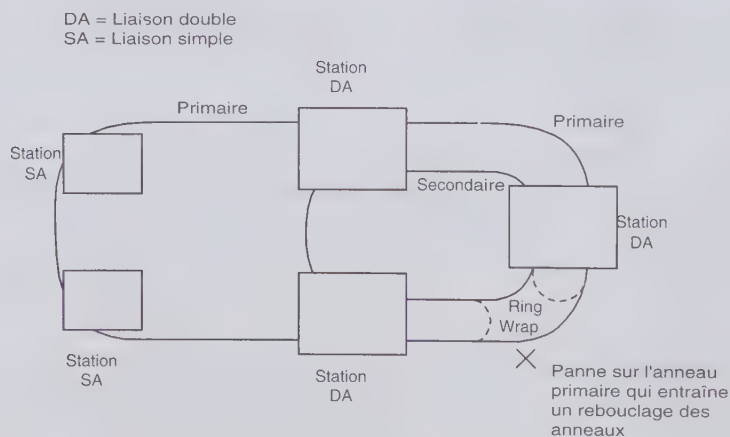
FDDI peut aussi être utilisé pour faire tourner des applications client/serveur basées sur les services IEEE 802.2 comprenant NetWare. Les adresses physiques des stations FDDI suivent la convention d'adressage de l'IEEE à 48 bits (6 octets).

• Fonctionnement de FDDI

Une configuration complète FDDI est formée de deux anneaux en fibre optique. L'anneau primaire est utilisé pour les transferts de données et l'anneau secondaire sert d'anneau de sauvegarde en cas de problème sur le premier anneau. Si l'anneau primaire tombe en panne, un dispositif automatique déroute le trafic sur l'anneau secondaire (voir Figure 3.56). Seules les stations raccordées sur les deux anneaux peuvent bénéficier de cette sauvegarde.

Figure 3.56

Anneaux FDDI avec les deux types de raccordement.



FDDI est basé sur la méthode d'accès Token Ring fonctionnant à 100 Mbit/s. Un jeton est utilisé pour contrôler l'accès à l'anneau mais la gestion du jeton est différente de celle des réseaux IEEE 802.5. La longueur maximale d'un réseau FDDI est de 200 km (100 km pour les doubles anneaux) et la distance entre deux nœuds d'un réseau FDDI ne peut excéder 2 km. Les distances sont fondées sur un délai maximum de 1,617 milliseconde. La taille maximale d'une trame FDDI est de 4500 octets. Cette taille convient bien aux transferts de fichiers demandant de la rapidité comme le transfert des graphiques, des images, etc. Comme la taille de la trame est grande, plus de données peuvent être contenues dans une trame et moins de paquets sont nécessaires pour envoyer un gros fichier. Le nombre total de connexions sur un anneau FDDI ne peut excéder 2000 (1000 pour des stations connectées à un double anneau).

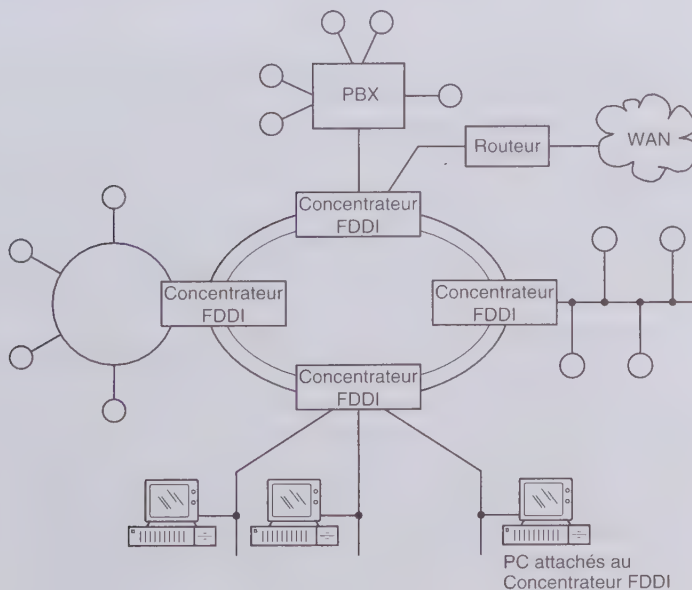
Dans les réseaux FDDI, les stations PC ne sont pas connectées directement au réseau mais le sont par l'intermédiaire de concentrateurs ou de routeurs (voir Figure 3.57). Si les stations sont connectées directement au réseau, leur mise en route et leur arrêt entraînent de fréquentes reconfigurations qui peuvent entraîner des coûts importants sur un grand réseau. Les stations connectées directement sur le réseau peuvent aussi ne pas fonctionner aux débits élevés de FDDI.

Les concentrateurs FDDI, appelés aussi *Multistation Access Unit (MAU)* jouent le rôle d'un fan-out auquel plusieurs stations peuvent se connecter. Plusieurs concentrateurs peuvent être reliés en cascade pour augmenter le nombre de ports. Bien que les concentrateurs acceptent les deux anneaux, les stations reliées à un concentrateur ne possèdent qu'une seule connexion pour diminuer les coûts des cartes réseau. Les concentrateurs peuvent rester connectés continuellement pour éviter les reconfigurations au démarrage.

La gestion du jeton sur un réseau FDDI permet d'avoir plusieurs trames en même temps sur l'anneau pour mieux en utiliser la bande passante.

Figure 3.57

Réseau FDDI doté de routeurs.



L'anneau FDDI opère dans deux modes :

- synchrone ;
- asynchrone.

Dans le *mode synchrone*, les stations sont assurées d'obtenir un certain pourcentage de la bande passante totale disponible sur l'anneau. Cette allocation de bande passante est calculée d'après le pourcentage d'un temps appelé TTRT (*Target Token Rotation Time*). Le temps TTRT représente le temps escompté de rotation du jeton pour un trafic donné sur le réseau. Il est négocié pendant l'initialisation de l'anneau. Une station possédant une allocation de bande synchrone transmet les données pendant une période qui n'excède pas le pourcentage de TTRT qui lui est alloué. Le temps restant après que toutes les stations ont fini leurs transmissions synchrones est alloué aux nœuds restants. Ainsi, si le temps de rotation TRT (*Token Rotation Time*) est inférieur à TTRT, la différence de temps (TTRT – TRT) est utilisée pour les transferts asynchrones.

Dans le *mode asynchrone*, les transferts ont lieu sous deux modes :

- restrictif ;
- non restrictif.

Une station peut effectuer un transfert dans le mode asynchrone restrictif. La station de gestion négocie un temps maximum restrictif. Les stations tournant en mode asynchrone ne peuvent occuper l'anneau entier pendant une période de temps supérieure à TTRT.

Dans le mode non restrictif, le temps restant est divisé entre les nœuds qui désirent envoyer des données. Ce mode d'opération est le mode par défaut. Le partage du temps est basé sur les schémas de priorité dans lesquels les stations ont un seuil TRT.

FDDI utilise la fibre optique multimode. On peut utiliser aussi le FDDI avec de la fibre monomode. Alors que la fibre multimode utilise plusieurs fréquences, une fibre monomode utilise un rayon laser et un diamètre de fibre plus petit que la fibre multimode. Les fibres monomodes possèdent une atténuation du signal plus faible que les fibres multimodes et peuvent donc être utilisées sur de plus longues distances. Deux stations peuvent alors se situer à 60 km l'une de l'autre. Une autre extension de FDDI, appelée FDDI-II, permet de fournir des circuits commutés dans le mode synchrone avec 16 canaux synchrones de 6,144 Mbit/s chacun.

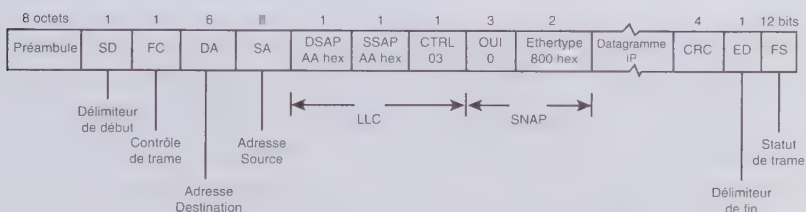
Les constructeurs s'intéressent aux médias en cuivre pour les réseaux FDDI. Certains utilisent de la paire torsadée non blindée alors que d'autres préfèrent la paire torsadée blindée. Les câbles en cuivre sont moins chers que la fibre optique, mais ils posent le problème de l'atténuation des signaux qui limite à 100 mètres la distance entre une station et le concentrateur FDDI. Des produits basés sur FDDI avec support en cuivre, appelés CDDI (*Copper Distributed Data Interface*), sont actuellement disponibles dans le commerce.

• Trame TCP/IP dans les réseaux FDDI

Même dans les réseaux locaux FDDI développés par l'ANSI et pas par l'IEEE, FDDI utilise LLC pour encapsuler les protocoles des couches supérieures. Cela permet d'utiliser l'en-tête SNAP avec la trame LLC pour obtenir des paquets TCP/IP. La trame des paquets TCP/IP des réseaux locaux FDDI est représentée Figure 3.58. Les champs DSAP et SSAP de l'en-tête LLC ont pour valeur AA (hexadécimal) et le champ EtherType contient la valeur 800 (hexadécimal) des paquets IP.

Figure 3.58

Trame des paquets TCP/IP dans les réseaux locaux FDDI.



Extension des réseaux locaux vers les réseaux étendus

Quand plusieurs réseaux locaux doivent être connectés sur de longues distances, des liens ou des réseaux étendus sont nécessaires. Les réseaux locaux et les réseaux étendus sont reliés par des périphériques spéciaux appelés ponts ou routeurs. La suite de ce chapitre explique le principe des ponts et des routeurs et présente les différentes techniques des réseaux étendus.

• Rôle des répéteurs, ponts et routeurs

Les répéteurs, les ponts et les routeurs permettent d'étendre la portée des réseaux locaux. L'extension d'un réseau local par des répéteurs est limitée par le nombre de répéteurs autorisés. Les ponts et les routeurs permettent d'étendre la portée d'un réseau. Par exemple, dans un réseau Ethernet dont la distance maximale est de 2500 mètres, des ponts ou des routeurs vous permettront d'augmenter cette distance. Chaque réseau local Ethernet suit encore les règles des 2500 mètres et des quatre répéteurs mais un paquet de ce réseau peut passer dans un autre réseau à travers un pont.



Les répéteurs, ponts ou routeurs permettent aussi d'augmenter le nombre de nœuds disponibles sur un réseau. Par exemple, sur un réseau Ethernet 10BASE5, le nombre maximum de stations est de 100 par segment. Si vous désirez augmenter ce nombre, vous pouvez utiliser un répéteur pour relier un autre segment. Si vous désirez dépasser la limite des 1024 stations par réseau local Ethernet, il vous faudra relier deux réseaux par un pont ou un routeur.

Les ponts et les routeurs ne se contentent pas d'augmenter les distances d'un réseau, ils permettent de connecter des réseaux différents et de partager un réseau existant pour en diminuer les goulets d'étranglement.

Les ponts et les routeurs sont des ordinateurs sur lesquels tournent des algorithmes effectuant des fonctions de pontage ou de routage. Pour obtenir de bonnes performances, ces ordinateurs sont construits sous forme de "boîte noire". Pour les ponts, les algorithmes sont souvent codés dans des EPROM en vue d'une exécution plus rapide. Les routeurs sont plus compliqués que les ponts et nécessitent souvent des mises à jour du firmware pour tenir compte de certains problèmes. Pour une mise à jour rapide, il est plus simple d'implémenter le logiciel de routage dans des mémoires "flash".

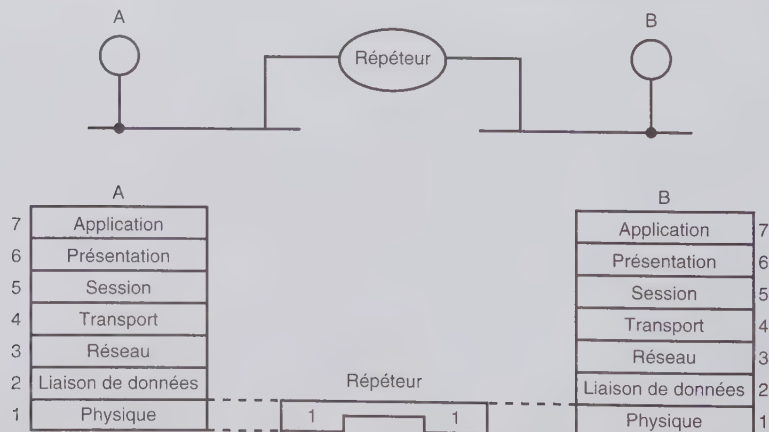
Les ponts et les routeurs actuels comportent souvent des slots permettant d'insérer des cartes additionnelles qui augmentent leurs possibilités.

• Le répéteur

Un *répéteur* agit au niveau de la couche physique du modèle OSI (voir Figure 3.59). Il prend le signal de l'un des réseaux, le reconditionne et l'envoie vers l'autre réseau. Le reconditionnement consiste en une amplification du signal. Le répéteur ne sait pas quelle est la signification des bits qu'il transmet et il ne peut reconnaître aucune adresse de la trame.

Figure 3.59

Le répéteur et le modèle OSI.



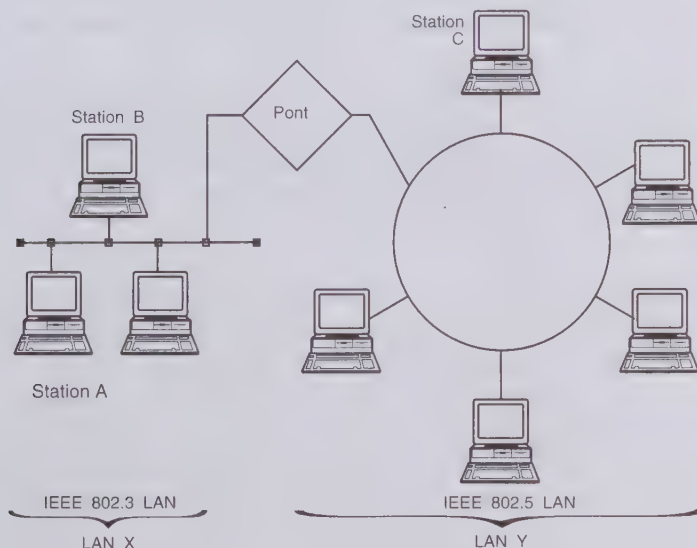
La fonction du répéteur est simple : il détecte le signal, l'amplifie et l'envoie à tous les ports à l'exception de celui sur lequel il a reçu le signal. Dans le cas des réseaux Ethernet, les signaux sont retransmis et même les collisions. Les segments reliés par des répéteurs participent de la même façon aux mécanismes d'accès au média. Pour les réseaux Token Ring, chaque station joue le rôle d'un répéteur. Certains supports en fibre optique pour Token Ring doivent utiliser des répéteurs spéciaux pour régénérer le signal sur les longues distances.

• Le pont

Les *ponts* connectent deux réseaux séparés pour former un réseau logique. La Figure 3.60 montre un réseau IEEE 802.3 et un réseau IEEE 802.5 reliés par un pont. Le pont possède deux cartes réseau : une carte Ethernet vers le réseau Ethernet et une carte Token Ring vers le réseau Token Ring.

Figure 3.60

Pont entre un réseau IEEE 802.3 et un réseau IEEE 802.5.



Le pont a une "double personnalité". Il se comporte comme une station Token Ring du côté du réseau Token Ring et comme une station Ethernet du côté Ethernet. Dans la Figure 3.61, un paquet envoyé de la station A vers la station B ne traverse pas le pont car les deux stations se situent du même côté du pont.



Si la station A envoie un paquet vers la station C, le pont accepte le paquet venant de A et l'envoie sur le réseau Token Ring vers la station C. Il ne peut transmettre directement le paquet d'un réseau à l'autre puisque ces deux réseaux n'utilisent pas le même type de trames. Le pont supprime l'en-tête Ethernet et le remplace par un en-tête Token Ring contenant l'adresse de C. Le pont doit attendre l'arrivée d'un jeton libre pour envoyer le paquet. Pendant cette attente, d'autres paquets peuvent arriver pour passer vers le réseau Token Ring. Ils sont placés dans une file d'attente. Un pont doit donc posséder une certaine capacité de stockage des trames en attente.

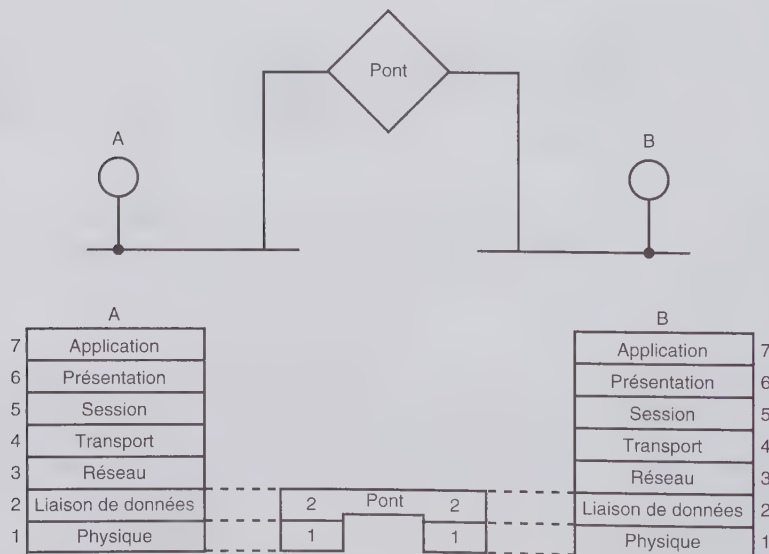
Dans la Figure 3.60, plusieurs stations du réseau X communiquent. Occasionnellement, certaines d'entre elles doivent communiquer avec des stations du réseau Y. Chaque réseau a son propre trafic. Une règle générale consiste à faire en sorte que 80 % du trafic reste dans un réseau et que 20 % seulement passent à travers le pont (règle du 80/20).

Si cette règle n'est pas respectée, les stations qui génèrent un trafic interréseau trop important doivent être déplacées sur un autre réseau. Des analyseurs de protocoles permettent de détecter ces stations.

Un pont opère au niveau de la couche liaison des données du modèle OSI (voir Figure 3.61) et effectue presque tout son travail au niveau de la couche 2. Le pont examine l'en-tête MAC du paquet dont l'adresse correspond à l'adresse physique de la station et est unique pour chaque station.

Figure 3.61

Le pont et le modèle OSI.



A la différence d'un répéteur, un pont "voit" le paquet de données. Les ports des ponts ont une adresse MAC unique. Un pont décode les données de la couche liaison des données. Il isole les mécanismes d'accès au média entre deux réseaux. Ainsi, les collisions d'un réseau CSMA/CD ne se propagent pas à travers le pont. De même, le jeton d'un réseau Token Ring ne traverse pas le pont. Les ponts permettent d'isoler les trafics et de diminuer les goulets d'étranglement.

Les ponts sont très intéressants pour relier un petit nombre de réseaux. Mais, si le nombre de réseaux augmente, il existera plusieurs chemins entre deux stations de réseaux différents et le pont ne saura pas choisir le chemin optimal car il agit au niveau de la couche liaison des données alors que les informations de routage se situent au niveau de la couche réseau. Les ponts ne peuvent pas prendre de décision quant aux routages des paquets. En effet, les informations concernant les routes se trouvent dans l'adresse réseau, qui n'est accessible que par la couche réseau.

Le pont reste un système d'interconnexion simple et bon marché. Pour permettre un routage efficace et intelligent, il faut utiliser un routeur.

• Routeurs et passerelles

Un *routeur* opère au niveau de la couche réseau du modèle OSI (voir Figure 3.62). Les routeurs examinent l'adresse réseau pour prendre des décisions de routage. Pour les réseaux TCP/IP, l'adresse réseau est représentée par *l'adresse IP*.

Comme les adresses réseau comportent des informations de routage, les routeurs peuvent prendre des décisions quant au routage. Ainsi, une *route* dans un réseau consiste en adresses réseau et en chemins. Les routeurs doivent savoir où envoyer un datagramme et le chemin optimal pour l'envoyer.

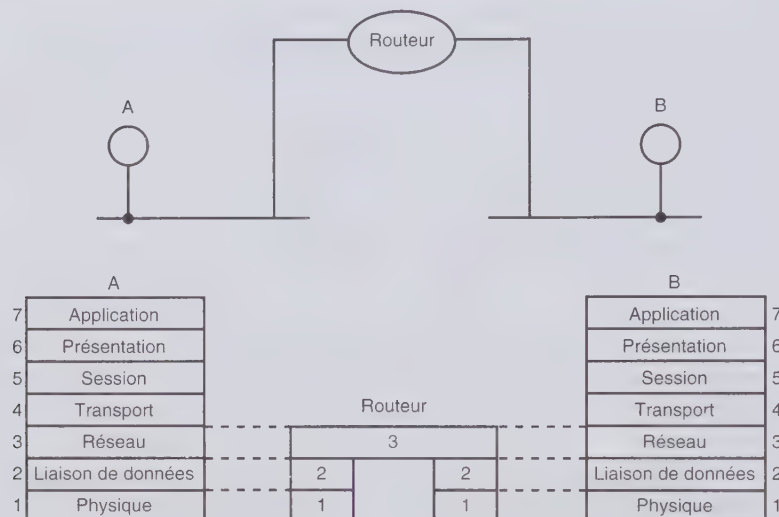
Les routeurs sont plus complexes que les ponts car ils doivent décoder des informations d'une couche située au-dessus de la couche où agissent les ponts. Cette complexité rend les routeurs plus coûteux que les ponts. Ils sont aussi plus lents.

Les routeurs sont d'excellents appareils pour relier de grands réseaux TCP/IP entre eux. Ils vous permettent aussi de connecter des réseaux sur de longues distances. Au début de l'Internet, les routeurs étaient appelés *passerelles*. Dans les réseaux actuels, une passerelle est un périphérique qui opère à des niveaux plus hauts que les routeurs.



Figure 3.62

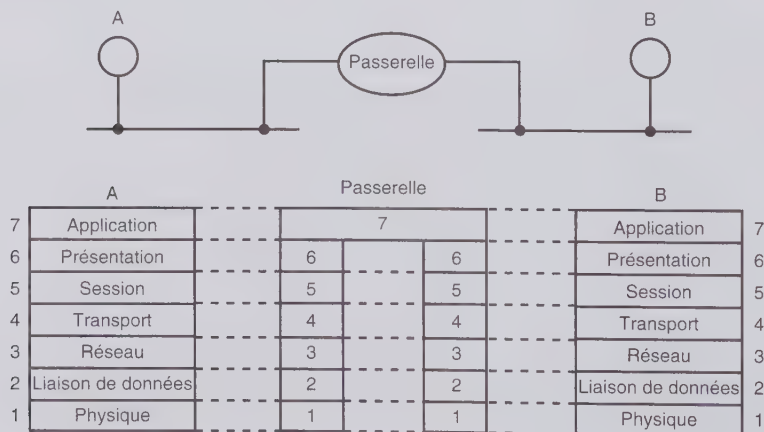
Les routeurs et le modèle OSI.



Dans la Figure 3.63, une passerelle opère au niveau 7 de modèle OSI (couche application). En général, une passerelle opère à n'importe quelle couche du modèle OSI et fournit une translation entre deux protocoles incompatibles.

Figure 3.63

Les passerelles et le modèle OSI.



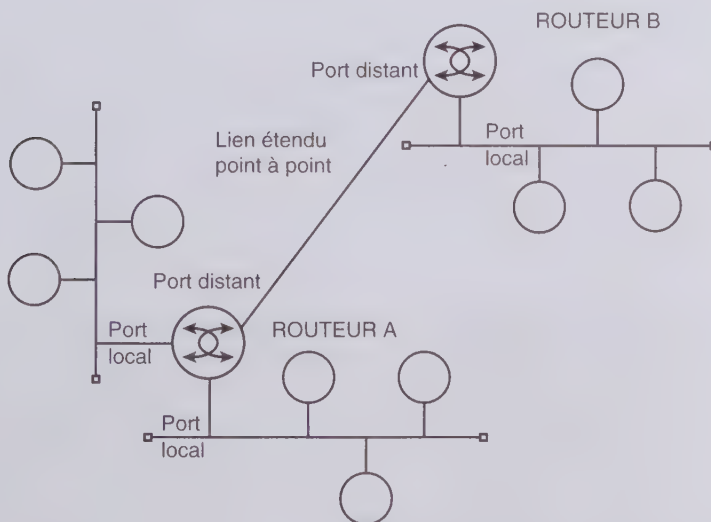
• Connexions locales et connexions distantes

Les ponts et les routeurs peuvent aussi être classés en fonction des possibilités de connexions locales ou distantes qu'ils offrent. Les différences dépendent des interfaces réseaux et de leurs ports.

Les ponts et les routeurs locaux possèdent des ports qui se connectent au média comme les stations du réseau.

Les ponts et les routeurs distants nécessitent des ports qui se connectent à des supports de transmission à longue distance. Il existe peu d'interfaces pour ce type de média. Les plus utilisés sont les ports RS-232 et V.35. Ces ponts et routeurs distants possèdent un ou plusieurs ports distants et au moins un port pour la connexion locale. La Figure 3.64 montre le routeur A possédant deux ports locaux et un port distant. Le routeur B possède un port local et un port distant. Les ports distants sont connectés par des liens point à point. Ces liens sont des liens X.25, Frame Relay, T1, SONET ou SMDS.

Figure 3.64
Routeurs distants.



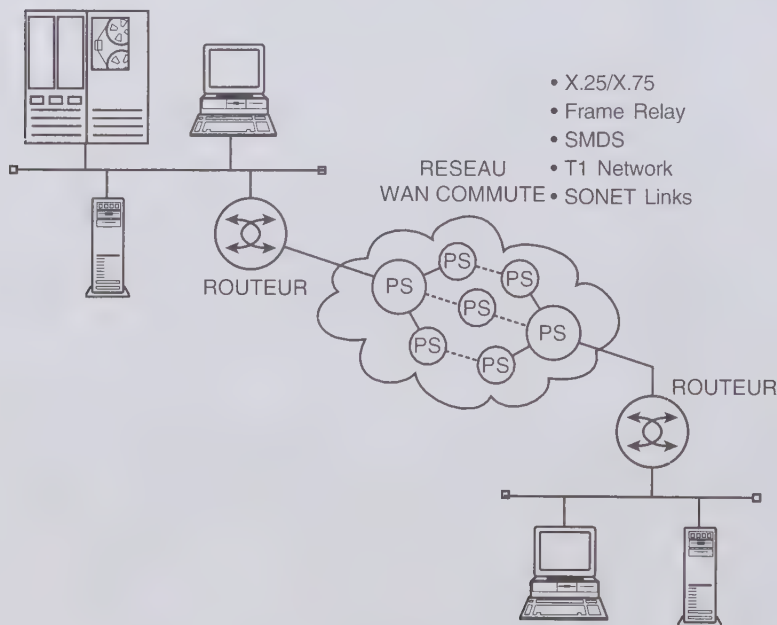
Vous pouvez aussi utiliser des circuits virtuels pour connecter des réseaux. Les liens point à point sont des circuits téléphoniques ou des circuits T1 loués. Comme ces lignes sont dédiées entre deux points, on les appelle des liens point à



point. Les circuits virtuels, comme celui représenté Figure 3.65, sont basés sur des systèmes de commutation. Les détails du circuit ne sont pas connus du réseau. Le circuit peut utiliser un protocole X.25, Frame Relay ATM ou une technologie propriétaire. Le circuit virtuel est géré par l'organisation qui fournit ce service.

Figure 3.65

Routeurs connectés par des circuits virtuels.



La différence entre ponts et routeurs locaux et distants se situe au niveau du coût des connexions. Comme les connexions locales sont gérées entièrement par l'entreprise qui possède les réseaux, aucun coût supplémentaire n'est nécessaire. Pour les connexions distantes, vous devez payer pour la connexion établie par le fournisseur du service.

• Fonctionnement d'un pont

Les ponts sont souvent utilisés pour étendre de petits réseaux TCP/IP. Comme les ponts opèrent au niveau 2 (couche liaison des données) du modèle OSI, ils ne connaissent pas le protocole IP qui correspond à la couche 3.

Les deux méthodes les plus employées pour les ponts sont :

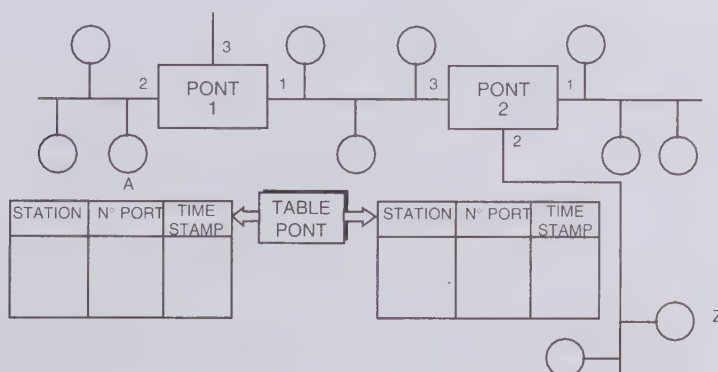
- le pont transparent (*transparent bridge*, TB) ;
- le routage par la source (*source routing*, SR).

Dans le pont transparent, la décision de relayer les paquets est prise par le pont de façon transparente pour les stations.

La Figure 3.66 montre un réseau à ponts transparents. Chaque pont gère une table qui contient les adresses des stations. Les ponts examinent l'adresse source de chaque paquet et enregistrent cette adresse dans la table avec le numéro du port d'arrivée du paquet. Les ponts contiennent aussi un champ de durée de vie (*time stamp*) pour chaque entrée de la table afin que les entrées les plus anciennes soient supprimées.

Figure 3.66

Réseau à ponts transparents.



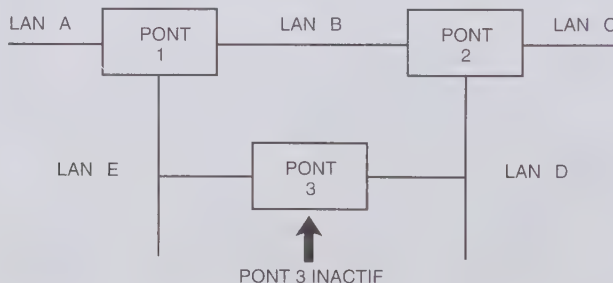
Considérons ce qui se passe quand la station A transmet un paquet à la station Z. Le pont 1 examine le paquet et consulte sa table pour savoir s'il possède une entrée pour la station Z. S'il n'en trouve pas, il envoie le paquet vers tous ses autres ports à l'exclusion du port qui a transmis le paquet. Le pont 1 vérifie aussi le champ d'adresse source et enregistre dans sa table que la station A est reliée au port 2. Quand le pont 2 voit le paquet, il répète le mécanisme précédent. S'il ne trouve aucune entrée dans sa table, il envoie le paquet vers tous ses ports. Il enregistre aussi l'adresse de la station A associée au port 3. Quand la station Z accepte le message de la station A, elle envoie un paquet contenant l'adresse source Z et l'adresse de destination A. Quand le pont 2 consulte sa table, il note que la station A peut être accédée par le port 3 et envoie le paquet uniquement par le port 3. Il enregistre aussi que l'accès à la station Z peut se faire par le port 2.



Pour empêcher que les paquets ne circulent indéfiniment, les réseaux à ponts transparents ne peuvent pas avoir de boucle. Le chemin des données forme un arbre qui relie toutes les stations du réseau. S'il y a des ponts qui peuvent former une boucle, comme le montre la Figure 3.67, ils doivent être inactifs. Les ponts inactifs agissent comme des ponts redondants. Ces ponts ne deviennent actifs que si la topologie de l'arbre change. Les changements sont transmis par le protocole *Bridge Protocol Data Units* (BPDU). Ce protocole spécial est utilisé pour maintenir la topologie de l'ensemble de l'arbre. Le processus qui conduit à l'obtention de l'arbre (*spanning tree*) est appelé *spanning tree algorithm*. L'un des ponts devient la racine et les autres ponts transmettent les trames en direction du pont racine en utilisant le chemin de moindre coût.

Figure 3.67

Boucle dans un réseau à ponts transparents.

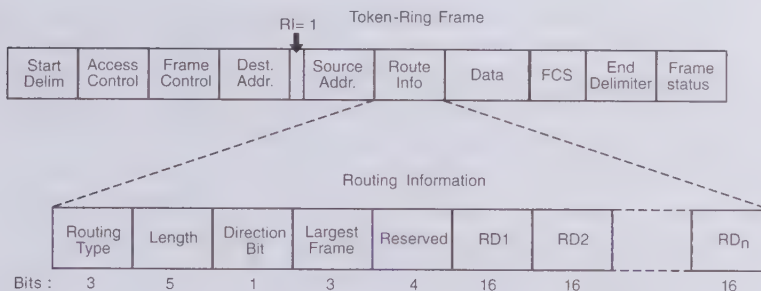


Les ponts spanning tree fournissent la méthode la plus efficace pour les réseaux Ethernet et IEEE 802.3.

Le routage par la source (*source routing*) est utilisé dans les ponts reliant des réseaux Token Ring. Dans ce mécanisme, l'émetteur (source) doit déterminer le meilleur chemin pour arriver à destination. Après avoir défini ce chemin, la station source l'enregistre dans sa table de routage et l'inclut dans le champ *d'information de routage* (RI) de chaque paquet qu'il envoie. Le champ RI apparaît dans la Figure 3.68 représentant le paquet Token Ring. Le champ RI existe si le bit I/G de l'adresse source est à 1. Les champs *Routing Designator* (RD) contiennent le chemin que le paquet doit suivre pour arriver à destination. Ces champs consistent en un numéro d'anneau et un numéro de pont. Les ponts SR suivent les renseignements du champ d'information de routage. Il peut y avoir 14 champs RD dans la trame, ce qui limite le chemin à 13 ponts. Les implémentations IBM limitent le total des champs RD à 8, ce qui correspond à 7 ponts.

Figure 3.68

Champ d'information de routage.

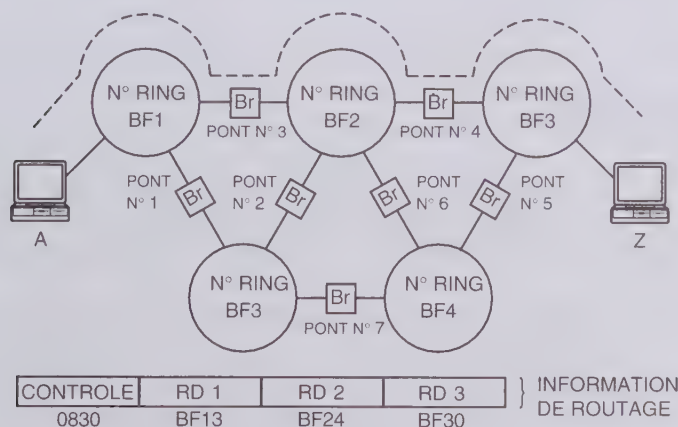


La Figure 3.69 montre le chemin des données de la station A à la station Z en utilisant le routage par la source.

La difficulté du routage par la source est de découvrir la route initiale. C'est l'envoi d'une trame de découverte (*discovery frame*) diffusée par la source qui permettra de découvrir le meilleur chemin. Cette trame circule sur tout le réseau et arrive à destination avec l'enregistrement du chemin d'accès. Les trames de découverte sont retournées à l'expéditeur qui choisit ensuite le meilleur chemin.

Figure 3.69

Transmission au moyen du routage par la source.



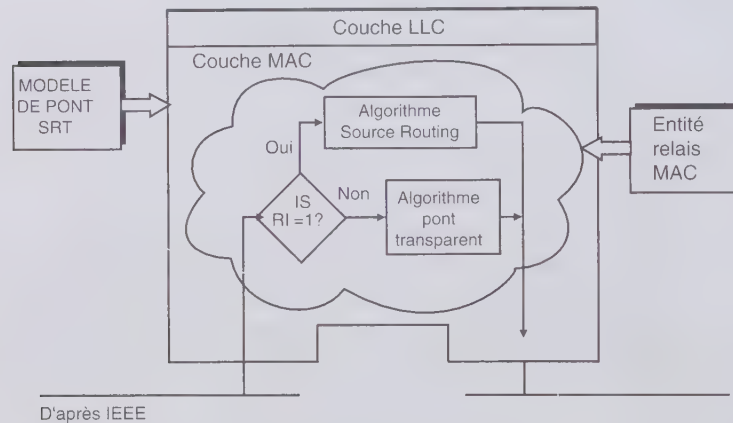
Les réseaux IEEE 802.5 utilisent de préférence le routage par la source. Pour connecter des réseaux qui utilisent les ponts transparents et des réseaux qui utilisent le routage par la source, un pont, comme le pont IBM 8209, traduit les informations de routage entre les deux types de ponts. Une autre méthode consiste à utiliser un pont qui soit une combinaison du routage par la source et du pont transparent. Ces



ponts sont appelés *Source Routing Transparent* (SRT). Dans ces ponts, le pont transparent est utilisé quand il n'existe pas de champ RI. Sinon, c'est le routage par la source qui est utilisé. Le modèle des ponts SRT est représenté Figure 3.70. La couche MAC consiste en algorithmes SR et TB. Ces algorithmes dépendent du bit qui indique la présence d'un champ RI.

Figure 3.70

Ponts SRT.



Récapitulatif

Les réseaux TCP/IP tournent sur plusieurs systèmes matériels. Même si les matériels et logiciels réseau sont transparents pour l'utilisateur d'un réseau, le professionnel doit en connaître toutes les finesses pour la maintenance et l'évolution du réseau.

Le matériel réseau pour les réseaux TCP/IP relève des couches 1 et 2 du modèle OSI (couches physique et liaison des données). Ce chapitre a décrit les différentes connexions pour TCP/IP des réseaux locaux et étendus. Il a aussi traité dans le détail des différents types de câblages et montré comment les répéteurs, ponts ou routeurs peuvent connecter plusieurs réseaux entre eux.

Testez vos connaissances

1. Pourquoi y a-t-il tant de types différents de réseaux pour faire tourner TCP/IP ? Quels en sont les avantages ?
2. Nommez deux périphériques utilisés pour connecter des réseaux utilisant des matériels différents.
3. Quelles sont les deux couches du modèle OSI décrites par les standards réseau IEEE ?
4. Quelle est la relation entre la couche LLC et la couche MAC ? Dessinez un diagramme décrivant cette relation et la relation entre ces couches et le modèle OSI.
5. Ecrivez les noms des liens SAP pour la couche LLC.
6. Décrivez brièvement les différents types de services disponibles sur la couche LLC.
7. Quel type de services LLC utilisent les réseaux Type 1, Type 2 ou Type 3 ? Donnez la ou les raisons de ces choix.
8. Ecrivez les noms des standards IEEE décrivant les mécanismes CSMA/CD et le passage du jeton.
9. Listez les différents types de réseaux Ethernet/802.3 et donnez une description de chacun de ces types.
10. Représentez une trame Ethernet montrant comment sont transportées les données TCP/IP. Donnez le nom de chaque champ.
11. Définissez le mécanisme de l'extension d'en-tête SNAP ? Pourquoi est-il employé ? Quels sont les réseaux qui utilisent ce mécanisme pour transporter des données TCP/IP ?
12. Représentez une trame Token Ring transportant des données TCP/IP. Nommez chaque trame.
13. Représentez une trame FDDI en montrant comment elle transporte des données TCP/IP. Nommez chaque champ.
14. Quelle est la fonction d'un moniteur actif dans les réseaux IEEE 802.5 ? Pourquoi un tel système n'est-il pas nécessaire dans les réseaux Ethernet ?
15. Comment pouvez-vous étendre la portée d'un réseau local ?
16. Donnez une description des différents types de périphériques utilisés pour étendre la portée d'un réseau local.

• 30 •



17. Expliquez, en vous aidant d'un dessin, le fonctionnement du pont spanning tree.
18. Expliquez, en vous aidant d'un dessin, le fonctionnement du pont à routage par la source.
19. Quels avantages apporte l'utilisation d'un commutateur dans les réseaux Ethernet par rapport aux réseaux sans commutateur ? Quelles sortes d'applications TCP/IP sont les mieux adaptées à la communication Ethernet ?
20. Existe-t-il une similitude entre Ethernet commuté et Ethernet par pont spanning tree ? Quelles en sont les différences ?
21. Quelles sont les principales différences entre un pont et un routeur ?

INFRASTRUCTURE DES RÉSEAUX TCP/IP INTERCONNECTÉS

4	<i>L'adressage IP</i>	153
5	<i>Protocoles de résolution d'adresse</i>	193
6	<i>Internet Protocol</i>	241
7	<i>Le routage IP</i>	295
8	<i>Le protocole ICMP</i>	325
9	<i>Sous-réseaux et surréseaux</i>	357
10	<i>Protocoles de routage IP</i>	403
11	<i>Protocoles de transfert</i>	443



4

L'adressage IP

Une des tâches fondamentales lors de la mise en place d'un réseau TCP/IP consiste à affecter des adresses Internet aux nœuds du réseau. Dans un réseau TCP/IP, les adresses Internet sont appelées des adresses IP. Si l'affectation des adresses IP semble aisée à première vue, vous devez prendre en considération un certain nombre de points. Ainsi, chaque adresse réseau IP doit être distincte. Vous ne pouvez vous contenter de connecter un nœud à un réseau IP et d'y affecter une adresse IP distincte. Vous devez aussi vous assurer que l'adresse IP est cohérente en regard des adresses IP des autres nœuds de ce segment de réseau. Si vous implémentez TCP/IP sur un réseau, une de vos tâches consistera à sélectionner et à configurer correctement les adresses IP.

Ce chapitre traite en détail des adresses IP. Nous définirons soigneusement le concept, parlerons des adresses IP spéciales et traiterons aussi bien de l'affectation que de la configuration des adresses IP. Ce chapitre traite principalement des adresses IP version 4, mais IP version 6 sera brièvement abordé.

INFO



Il existe deux versions d'IP : IP version 4 et IP version 6 (ou IPv6). IP version 4, qui reste le protocole dominant sur l'Internet, utilise des adresses sur 32 bits. IPv6, destiné à remplacer à terme IPv4, utilise, quant à lui, des adresses sur 128 bits.

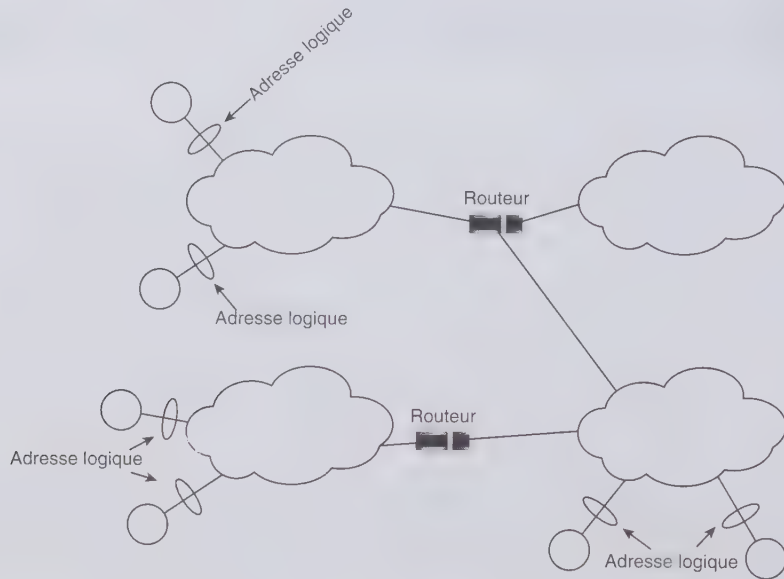
Qu'est-ce qu'une adresse de réseau ?

Pour connecter des nœuds dans un interréseau constitué de plusieurs réseaux (voir Figure 4.1), il est nécessaire d'utiliser sur l'ensemble de réseau un mécanisme d'adressage logique cohérent. Les adresses de réseau sont les identificateurs SAP (point d'accès de service) au niveau de la couche réseau du modèle OSI (voir Figure 4.2). Les adresses NSAP (point d'accès de service de réseau) doivent être distinctes pour chacune des entités de protocole communiquant au niveau de la couche réseau du modèle OSI.

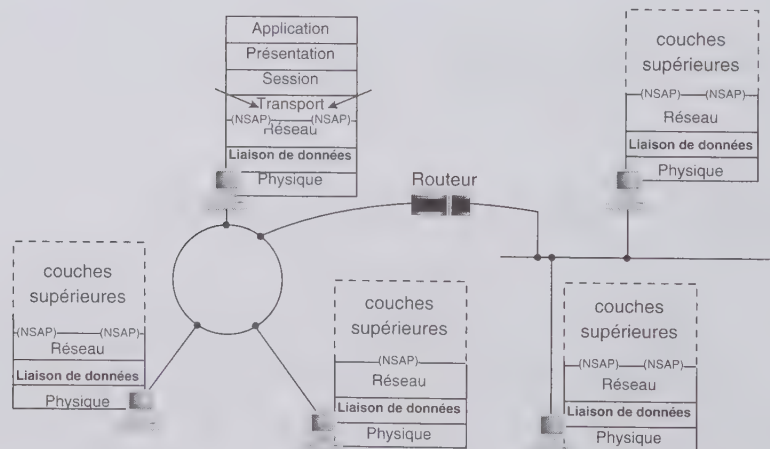


Figure 4.1

Les interréseaux doivent utiliser un mécanisme d'adressage logique cohérent.

**Figure 4.2**

Les SAP de réseau doivent être uniques.



• Valeurs d'adresses pour les NSAP

Au niveau de la couche réseau du modèle OSI, on préfère généralement utiliser pour les adresses NSAP une valeur numérique plutôt qu'un nom. En effet, les valeurs numériques sont manipulées plus facilement par les protocoles dans les couches basses du modèle OSI. On utilise des noms dans le cas des applications plus proches de l'utilisateur final (au niveau de la couche application), l'être humain manipulant, quant à lui, plus aisément les noms que les nombres. Les logiciels de protocole fonctionnent cependant plus efficacement avec une représentation numérique des adresses.

L'Internet Protocol (IP) opère au niveau de la couche réseau du modèle OSI. Cette couche traite des questions liées au routage. L'adresse IP doit être structurée de manière à assurer un routage efficace. Or, les calculs de routage sont bien plus simples en utilisant des nombres binaires plutôt que des noms symboliques. C'est là encore un argument en faveur des adresses numériques.

• Indépendance des adresses NSAP vis-à-vis du matériel du réseau

Comme nous avons pu le voir dans la section précédente, pour plus d'efficacité on utilise des valeurs numériques pour les adresses NSAP. Ces adresses sont nécessaires pour chaque connexion réseau effectuée sur le réseau physique. Si on trouve n connexions réseaux à partir d'un hôte, vous devez affecter n adresses NSAP. Le matériel du réseau, les cartes réseaux par exemple, permettent d'assurer la connexion physique au réseau, et les adresses réseaux matérielles consistent en des valeurs numériques. On pourrait être dès lors tenté d'utiliser ces adresses matérielles de réseau comme adresses NSAP. Il serait effectivement possible de mettre en place un mécanisme pour utiliser les adresses MAC comme des adresses NSAP, mais plusieurs problèmes se poseraient alors.

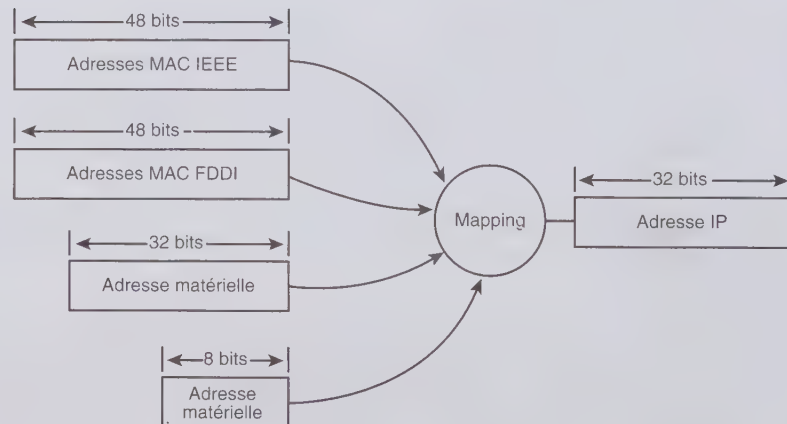
Ainsi, que se passerait-il si l'on changeait les cartes réseau (en cas de panne, par exemple) ? Dans le cas d'Ethernet, d'un Token Ring ou de FDDI, il s'ensuivrait une modification de l'adresse MAC, et donc une modification de l'adresse NSAP. Dans un tel mécanisme d'adressage, l'adresse NSAP est soumise aux changements survenant dans la configuration matérielle du réseau. Pour cette raison, les concepteurs de l'Internet firent le choix d'un mécanisme d'adresse de réseau logique pour les adresses IP, qui serait indépendant des adresses de la couche physique.



Au Chapitre 3, nous avons vu que TCP/IP s'exécute sur des types de matériels de réseau très divers. Ainsi, Ethernet, les anneaux à jetons et FDDI utilisent des adresses sur 48 bits. D'autres types de matériels de réseau utilisent eux des adresses sur 8, 16 ou 32 bits. En préférant les adresses logiques aux adresses physiques, IP n'a pas à se soucier des particularités matérielles du réseau sous-jacent. Les différentes adresses matérielles doivent être mappées en adresses de réseau logiques (voir Figure 4.3). Pour les adresse IP, des protocoles de résolution d'adresse spéciaux sont utilisés pour ce mapping. Les protocoles de résolution d'adresse sont traités au Chapitre 5.

Figure 4.3

Les adresses matérielles doivent être mappées en adresses logiques.



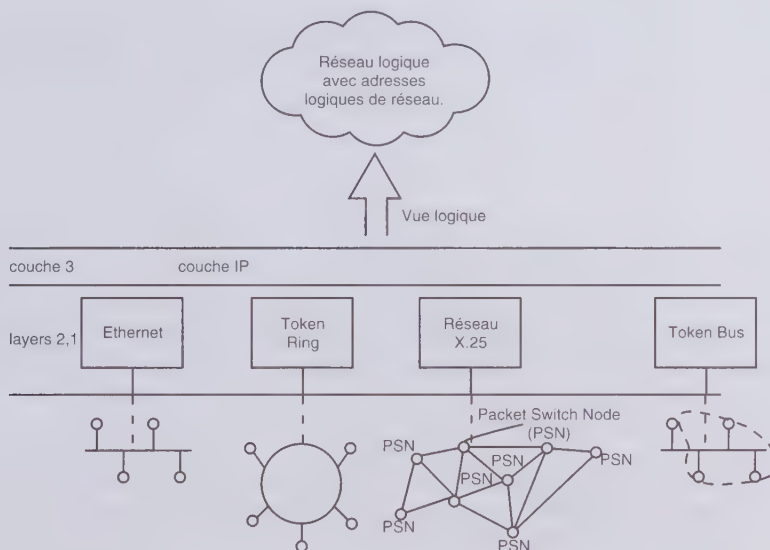
Comme les adresses IP ne dépendent pas des adresses physiques, vous pouvez modifier le matériel sous-jacent sans modifier l'adresse logique. Autrement dit, vous pouvez mettre à niveau votre réseau sans pour autant modifier les adresses logiques.

L'adresse IP

L'Internet a été conçu comme un réseau logique d'ordinateurs, dans lequel périphériques et logiciels sont connectés. Cette vision logique du réseau rend ce dernier indépendant de la technologie matérielle sous-jacente (voir Figure 4.4). Bien que le concept semble simple, il s'agit là d'une des caractéristiques qui assure la puissance et la souplesse de l'Internet. L'indépendance vis-à-vis du matériel a ainsi permis à l'Internet de se développer jusqu'à devenir le plus grand réseau du monde, contenant tous les matériels informatiques imaginables. Il est ainsi possible de connecter au réseau de nouveaux types de matériels sans en changer la configuration logique.

Figure 4.4

Un réseau logique est indépendant du matériel sous-jacent.



• Structure de l'adresse IP

La vision logique de l'Internet crée un réseau virtuel. Chaque connexion de réseau à ce réseau virtuel est identifiée de manière distincte, au moyen d'une adresse NSAP ou IP. Les concepteurs du réseau peuvent sélectionner la taille de l'adresse IP en fonction de la taille prévue pour le réseau virtuel. On a choisi un nombre à 32 bits pour l'adresse IP. Avec ces 32 bits, le nombre maximal de connexions au réseau est de 2^{32} , soit 4 294 967 296 connexions. Comme nous le verrons par la suite, le nombre maximal effectif de connexions est moindre, car certaines représentations binaires de l'adresse IP ont des significations réservées et ne peuvent être attribuées à des connexions de réseau individuelles.

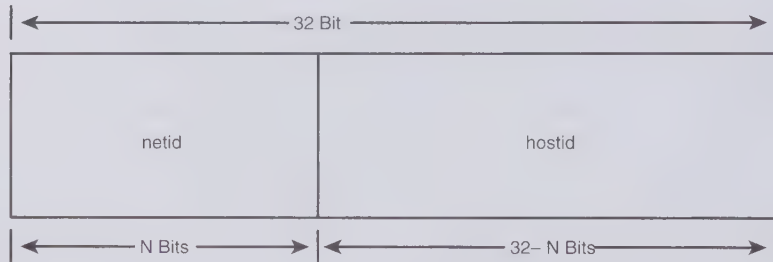
Le réseau virtuel est constitué des réseaux reliés entre eux par le biais de périphériques de connexion tels que les routeurs et les passerelles (voir Chapitre 3). Pour aiguiller des datagrammes IP, les routeurs doivent être en mesure de faire la distinction entre différents réseaux logiques. On a donc décidé de structurer l'adresse IP de façon qu'elle puisse refléter la distinction entre les différents réseaux logiques. Un certain nombre de bits dans l'adresse IP sont utilisés pour identifier le réseau individuel dans le réseau virtuel, et les bits suivants permettent d'identifier l'hôte au sein du réseau. La Figure 4.5 illustre la division de l'adresse IP.

P. 40, 1



Figure 4.5

L'adresse IP permet d'identifier le réseau et l'hôte.



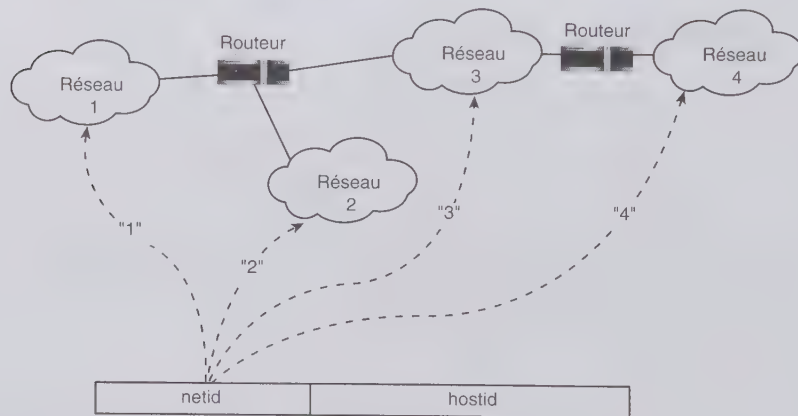
La portion de l'adresse concernant le réseau est appelé ID de réseau (*netid*), et la portion hôte, ID d'hôte (*hostid*). L'adresse IP peut donc être décrite sous la forme :

{*netid*, *hostid*}

Le *netid* décrit chaque réseau connecté et le *hostid* identifie l'hôte au sein de ce réseau. La Figure 4.6 montre comment des *netids* uniques peuvent être utilisés pour identifier les réseaux connectés.

Figure 4.6

Un exemple de réseau dans lequel les valeurs de *netids* permettent d'identifier les réseaux.



La méthode consistant à partager l'adresse IP entre un numéro de réseau et un numéro d'hôte au sein de ce réseau constitue un plan d'adressage hiérarchique. L'adressage hiérarchique rend plus efficace le routage. La plupart des plans hiérarchiques sont liés à une région géographique, mais l'adressage hiérarchique pour les adresses IP est logique et non pas géographique.

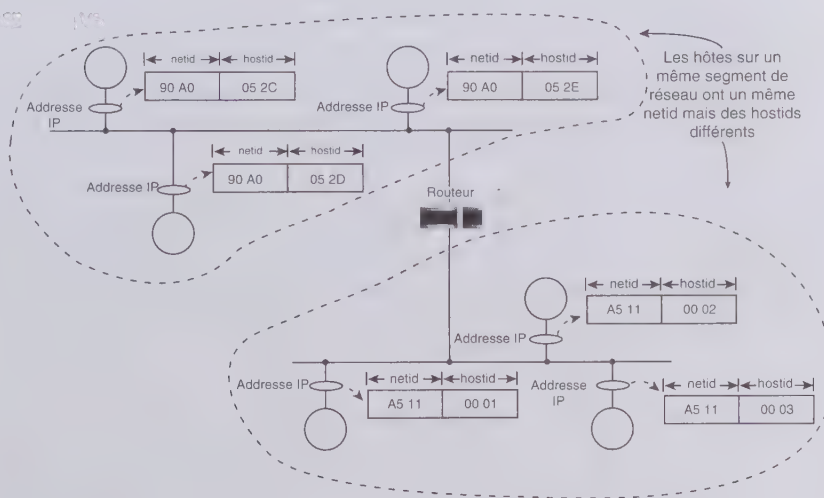
Le but premier des routeurs est d'envoyer un datagramme IP au réseau idoine. A cet effet, ils doivent stocker des informations concernant les *netids*, et pas les *hostids*. Le nombre de *netids* est inférieur à celui des *hostids*, ce qui maintient à

un niveau raisonnable la quantité d'informations qu'un routeur doit connaître. Si on n'avait pas établi de distinction entre le numéro de réseau et le numéro d'hôte – en choisissant un plan d'adressage plat plutôt qu'un plan d'adressage hiérarchique – les routeurs auraient dû stocker les quatre milliards d'adresses IP !

• Classes d'adresse IP

Tous les hôtes connectés au même réseau possèdent le même netid (voir Figure 4.7) et, évidemment, un hostid différent. Dans la Figure 4.7, les valeurs de netid et de hostid sont représentées en hexadécimal. Les nombres hexadécimaux permettent de représenter simplement des nombres binaires en groupes de 4 bits (si vous n'êtes pas familier des nombres hexadécimaux, reportez-vous au Tableau 4.2). Le netid identifie le réseau de manière distincte. Les réseaux interconnectés doivent avoir des netids distincts pour que le routage fonctionne correctement.

Figure 4.7
Partager un netid sur un réseau.



Les réseaux qui possèdent un même netid ont un préfixe commun représentant les adresses IP des hôtes présents au sein du réseau. La taille de ce préfixe est variable.



A partir du nombre 32 bits utilisé pour l'adresse IP, les concepteurs de l'Internet ont décidé qu'on utiliserait le premier, les deux premiers ou les trois premiers octets comme netid :

- **Premier octet (classe A).** Si le premier octet est utilisé comme netid, les trois octets restants servent pour le hostid. Ce type de format d'adresse IP est appelé une adresse de classe A.
- **Deux premiers octets (classe B).** Si les deux premiers octets sont utilisés pour le netid, les deux octets restant servent pour le hostid. Ce type de format d'adresse IP est appelé une adresse de classe B.
- **Trois premiers octets (classe C).** Si les trois premiers octets sont utilisés pour le netid, l'octet restant sert pour le hostid. Ce type de format d'adresse IP est appelé une adresse de classe C.

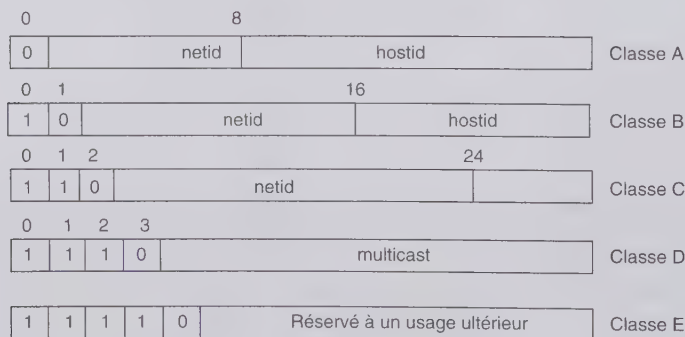
L'adresse IP est partitionnée en une paire netid/hostid, alignée à l'octet. On obtient ainsi trois classes d'adresse qui peuvent être affectées pour les connexions de réseau. Cet alignement à l'octet vient d'une volonté de simplicité, mais on aurait très bien pu partitionner l'adresse IP en s'alignant au bit, autrement dit affecter N bits pour le netid et $32-N$ bits pour le hostid.

Comment distinguer les différents formats d'adresse IP ? Les bits les plus significatifs permettent de déterminer le format de l'adresse IP, c'est-à-dire le nombre de bits utilisés pour le netid et le hostid.

En plus de ces trois classes A, B et C, deux autres classes sont définies (voir Figure 4.8). On peut affecter des adresses de classe A, B ou C, tandis que la classe D est réservée à la multidiffusion (*multicasting*) – une technique utilisée par des protocoles spéciaux pour transmettre simultanément des messages à un groupe donné de nœuds différents. La classe E est réservée à un usage ultérieur.

Figure 4.8

Classes d'adresses IP définies.



• Pourquoi utiliser des classes d'adresses spécifiques

Les différents types de classes d'adresses IP sont définis pour répondre aux besoins des réseaux de différentes tailles. Sur demande, l'autorité d'enregistrement du réseau affecte un numéro de réseau (le champ netid) à une organisation. Une fois ce numéro alloué à une organisation, il incombe à cette dernière d'affecter les numéros d'hôte (les champs hostids).

Le nombre d'hôtes qui peuvent être affectés pour un numéro de réseau donné dépend du nombre de bits dans le champ hostid, qui dépend lui-même de la classe d'adresses du réseau. Un réseau de classe A dispose du plus grand nombre de bits pour le hostid et, par conséquent, du plus grand nombre d'hôtes possible. Un réseau de classe C dispose du plus petit nombre d'hôtes possible. Le Tableau 4.1 indique les nombres de réseaux et de nœuds possibles pour chaque classe d'adresses.

Tableau 4.1 : Le nombre de réseaux et de nœuds possibles pour chaque classe d'adresses

Classe d'adresse	Nombre de réseaux	Nombre de nœuds
A	127	16 777 214
B	16 383	65 534
C	2 097 151	254

Les réseaux de classe A sont adaptés aux très grands réseaux, mais comme leur champ netid (voir Figure 4.8) ne fait que sept bits, il ne peut y avoir que 127 réseaux de classe A. L'ARPAnet originel est un exemple de réseau de classe A. Les réseaux de classe B sont des réseaux de classe moyenne, adaptés aux organisations moyennes ou plus grandes. Les réseaux de classe C sont adaptés aux petites organisations, dans lesquelles chaque réseau ne peut disposer de plus de 254 nœuds.

• Notation décimale à points pour les adresses IP

Pour simplifier les choses, on représente le nombre à 32 bits sous forme de quatre nombres décimaux correspondant à la valeur décimale des quatre octets qui composent l'adresse IP sur 32 bits. Les nombres décimaux sont séparés par des points (.). Cette notation pour les adresses IP est appelée la notation décimale à points. Voici une adresse IP sous sa forme binaire et sous sa forme décimale à points :

Adresse IP = 10010000 00010011 01001010 11001001

Adresse IP = 144.19.74.201



INFO



Comment déterminer que le nombre binaire 10010000 est égal au nombre décimal 144 ? Les nombres binaires utilisent la base 2 (et les nombres décimaux la base 10). Le nombre décimal 144 peut être décomposé en l'expression arithmétique suivante :

$$(1 \times 10^2) + (4 \times 10) + (4 \times 1) = 144$$

De même, le nombre binaire 10010000 peut être décomposé en l'expression suivante :

$$\begin{aligned} & (1 \times 2^7) + (0 \times 2^6) + (0 \times 2^5) + (1 \times 2^4) + (0 \times 2^3) + (0 \times 2) + (0 \times 1) \\ &= 2^7 + 2^4 \\ &= 144 \end{aligned}$$

Chaque bit (valant 1 ou 0) dans un nombre binaire est multiplié par 2 à la puissance du nombre de bits situé à sa droite. Le premier terme de l'expression précédente, 1×2^7 correspond au bit le plus à gauche dans 10010000. Le bit le plus à gauche est appelé le bit le plus significatif (MSB), et le bit le plus à droite, le bit le moins significatif (LSB). On compte sept bits à droite du bit le plus significatif, de sorte que sa valeur est 1×2^7 .

Le Chapitre 9 approfondira la question des nombres binaires dans la section intitulée "Conversion entre nombres décimaux et binaires".

La Figure 4.9 montre la relation existant entre la représentation binaire de l'adresse IP et la notation décimale à points. Sur la figure, vous pouvez constater que le premier groupe de 8 bits (10010000, dont la valeur décimale est 144) devient le premier nombre décimal de l'adresse IP à points. De même, les deuxième, troisième et quatrième groupes de 8 bits sont chacun représentés par leur valeur décimale.

Certaines implémentations de TCP/IP utilisent une notation hexadécimale à points. Les nombres hexadécimaux sont en base 16. Le Tableau 4.2 montre les symboles utilisés pour les nombres hexadécimaux, ainsi que les valeurs décimale et binaire correspondantes.

Tableau 4.2 : Valeurs hexadécimales

Code hexadécimal	Valeur décimale	Représentation binaire
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110

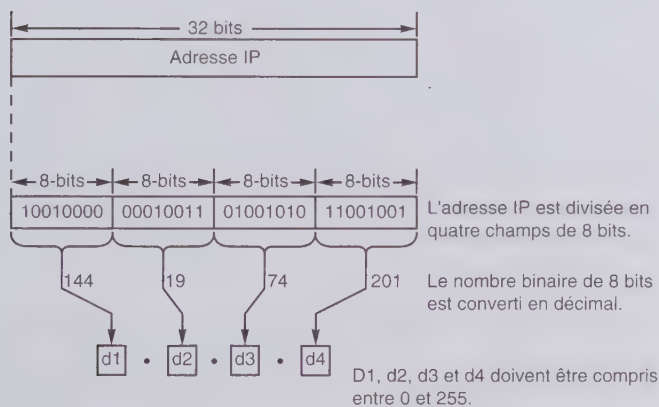
Tableau 4.2 : Valeurs hexadécimales (suite)

Code hexadécimal	Valeur décimale	Représentation binaire
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Figure 4.9

La notation décimale à points.

- Pour que le nombre binaire à 32 bits soit plus facilement lisible, on utilise la notation décimale à points.



Exemple : décimal à point 144.19.74.201



L'adresse IP 144.19.74.201 serait donc exprimée sous cette forme dans la notation hexadécimale à points :

0x90.0x13.0x4A.0xC9

Les chiffres qui suivent le préfixe 0x sont hexadécimaux.

Les zéros en préfixe peuvent être omis dans la notation décimale à points. La plupart des implémentations de TCP/IP ignorent les zéros en préfixe, toutefois certaines implémentations interprètent les zéros comme une indication que le nombre qui suit est en octal. Il est donc préférable de ne pas écrire l'adresse IP 129.12.33.61 sous la forme 129.012.033.061. En effet, dans le cas d'une implémentation interprétant les 0 en préfixe comme un symbole marquant un nombre octal, cette adresse serait interprétée comme étant 129.9.24.43.

• Calculer une classe d'adresses

A partir d'une adresse IP présentée sous forme de notation décimale à points, il est important de pouvoir déterminer la classe d'adresses qui lui correspond. La classe d'adresses IP détermine le nombre de bits affectés au champ hostid. La taille de ce champ limite le nombre d'hôtes pouvant être connectés au réseau. De plus, en connaissant la classe d'adresses on peut déterminer la façon de diviser un réseau en plusieurs réseaux plus petits appelés des sous-réseaux.

Une méthode permettant de déterminer la classe d'adresses IP consiste à convertir l'adresse IP dans sa forme binaire et à examiner les deux bits les plus significatifs (qui se trouvent le plus à gauche). Ces bits déterminent la classe d'adresses IP. Sur la Figure 4.8, vous pouvez constater que si le bit le plus significatif vaut 0, l'adresse est de classe A. Si les deux bits les plus significatifs sont 10, la classe est B, et si les trois bits les plus significatifs sont 110, la classe est C. Le Tableau 4.3 résume cette règle.

Tableau 4.3 : Classe d'adresse IP et bits les plus significatifs de l'adresse IP

Bits les plus significatifs	Classe d'adresse IP
0	Classe A
10	Classe B
110	Classe C
1110	Classe D

Prenons l'adresse IP 137.65.4.1. Si vous convertissez cette adresse IP sous forme binaire, vous obtenez :

10001001 01000001 00000100 00000001

Les deux bits les plus significatifs sont 10. L'adresse 137.65.4.1 est donc une adresse de classe B.

Prenons un autre exemple, l'adresse IP 199.245.180.10. La conversion de cette adresse sous forme binaire donne :

11000111 11110101 10110100 00001010

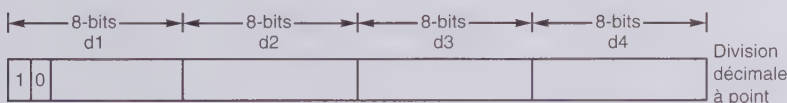
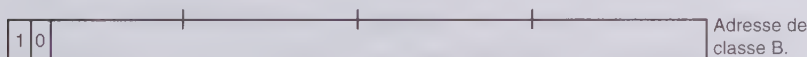
Les trois bits les plus significatifs sont 110. L'adresse 199.245.180.10 est donc de classe C.

Cette technique fonctionne bien, mais elle est laborieuse. Il existe heureusement une méthode plus simple. Prenons l'exemple de l'adresse de classe B de la Figure 4.10. Dans le cas d'une adresse de classe B, les deux bits les plus significatifs sont 10. La valeur minimale des huit premiers bits est atteinte lorsque les six autres bits valent 0, la valeur maximale lorsque les six autres bits valent 1.

En conséquence, la valeur minimale des huit premiers bits d'une adresse de classe B est 10000000 et la valeur maximale 10111111, soit 128 et 191 en décimal. Autrement dit, si le premier nombre décimal d'une adresse IP en notation décimale à points est compris entre 128 et 191 (inclus), l'adresse IP est une adresse de classe B. Dans l'exemple précédente de l'adresse 137.65.4.1, le nombre 137 est compris entre 128 et 191, et l'adresse correspondante est donc de classe B.

Figure 4.10
Déterminer la classe
d'une adresse IP.

- Le premier nombre décimal de l'adresse IP en notation décimale à point indique la classe d'adresse.



Valeur min de d1 = 1 0 0 0 0 0 0 0 = 128 (décimal)

Valeur max de d1 = 1 0 1 1 1 1 1 1 = 191 (décimal)

- Conclusion : pour une adresse de classe B, le premier nombre décimal doit être entre 128 et 191.



On peut procéder de même pour les classes A et C, ce qui donne le Tableau 4.4.

Tableau 4.4 : Déterminer la classe d'une adresse IP à partir du premier nombre décimal de l'adresse

Classe d'adresse IP	Minimum	Maximum
A	0	126
B	128	191
C	192	223
D	224	239
E	240	247

Adresses IP spéciales

Il existe certains cas particuliers d'adresses IP. La RFC 1700 décrit une notation particulière permettant d'exprimer des adresses IP spéciales :

Adresse-IP ::= { <Numero-reseau>, <Numero-hote> }

La ligne précédente définit une adresse IP comme étant composée d'un numéro de réseau et d'un numéro d'hôte. <Numero-reseau> correspond au netid et <Numero-hote> au hostid dont nous avons parlé précédemment.

La notation "-1" signifie que le champ contient tous les bits à 1, tandis que "0" signifie que tous les bits sont à 0.

On a défini les adresses IP spéciales suivantes :

- {<Numero-reseau>, 0} ;
- {<Numero-reseau>, -1} ;
- {-1, -1} ;
- {0, 0} ;
- {0, <Numero-hote>} ;
- {127, <quelconque>}.

Des adresses dont les bits du champ <Numero-hote> ne sont composés que de 0 ou de 1 ne peuvent être affectées comme adresses IP individuelles. Cela signifie que si le champ de numéro d'hôte contient n bits, vous ne pouvez affecter que $2^n - 2$ adresses IP.

Ces adresses spéciales sont détaillées dans les sections qui suivent. En plus des adresses spéciales que nous venons de citer, il existe d'autres adresses spéciales

destinées à la diffusion au sein de sous-réseaux. Les sous-réseaux seront traités au Chapitre 9.

• Adresses pour le réseau lui-même

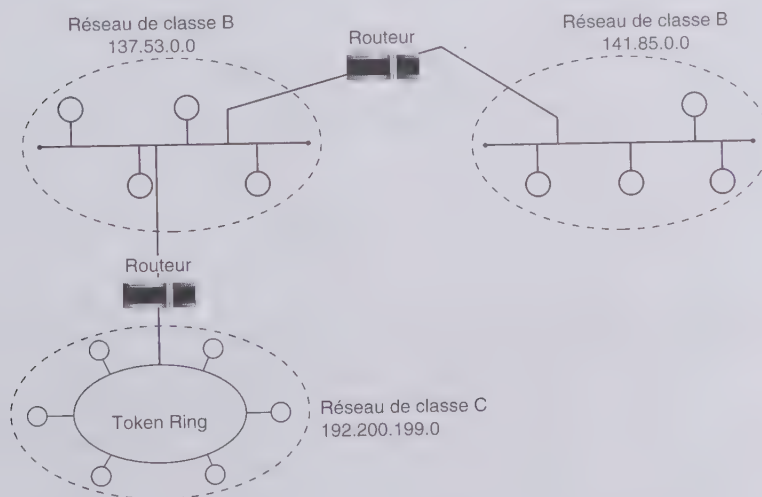
Une valeur d'hostid de 0 n'est jamais affectée à un hôte TCP/IP individuel. Une adresse IP dont l'hostid vaut 0 indique le réseau lui-même. Cette adresse spéciale est décrite de la manière suivante :

{<Numero-reseau>, 0}

Prenons l'adresse IP 137.53.0.0. Il s'agit d'un réseau de classe B. Comme nous l'avons vu, dans un réseau de classe B, <Numero-reseau> fait 16 bits (soit 2 octets, ou les deux premiers nombres décimaux dans l'adresse décimale à points) et <Numero-hote> correspond aux 16 autres bits de l'adresse IP. En conséquence, l'adresse spéciale 137.53.0.0 fait référence au réseau de classe B 137.53 lui-même. La Figure 4.11 montre plusieurs réseaux interconnectés qui utilisent ces adresses spéciales.

Les deux adresses de classe B 137.53.0.0 et 141.85.0.0 ont "0.0" comme hostid. Ces adresses indiquent tous les nœuds dans les réseaux. L'adresse {<Numero-reseau>, 0} ne peut être utilisée comme adresse IP de source ou de destination.

Figure 4.11
*Utilisation de l'adresse spéciale
{<Numero-reseau>, 0}.*



• Diffusion dirigée

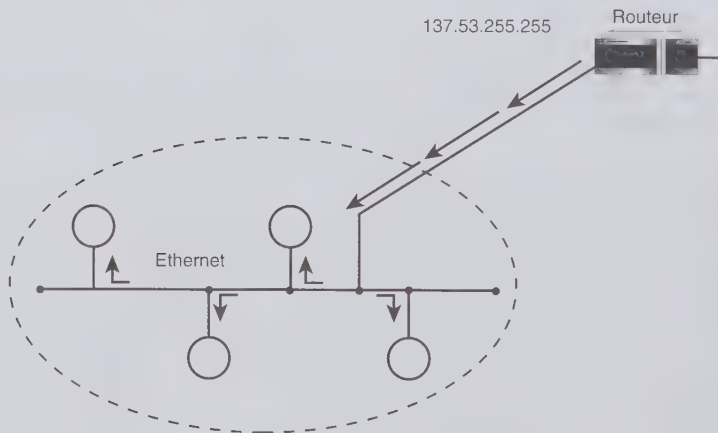
Une diffusion dirigée (ou *directed broadcast*) est une diffusion envoyée à tous les nœuds d'un réseau particulier. Si l'hostid ne contient que des bits à 1, il s'agit d'une adresse de diffusion dirigée. Une adresse de diffusion dirigée peut apparaître dans l'adresse IP de destination d'un datagramme IP, mais jamais comme source. Cette adresse spéciale s'exprime ainsi :

{<Numero-reseau>, -1 }

Une adresse de diffusion dirigée sera visible pour tous les nœuds du réseau (voir Figure 4.12). En conséquence, pour le réseau numéro 137.53, l'adresse de diffusion sera 137.153.255.255. Le numéro de réseau 137.53 est une adresse de classe B et comprend un hostid de 16 bits (soit 2 octets ou les deux premiers nombres décimaux de l'adresse décimale à points). Les 16 bits à 1 pour l'hostid correspondent à la valeur décimale 255.255.

Figure 4.12

Une diffusion IP avec prise en charge matérielle de la diffusion.



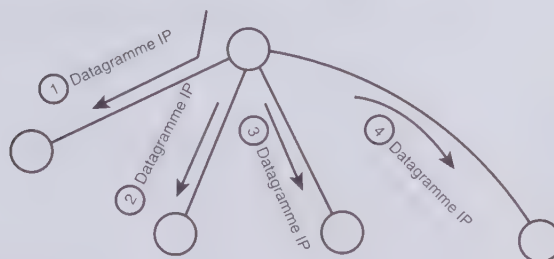
Dans l'exemple de la Figure 4.12, le matériel sous-jacent du réseau est Ethernet, qui prend en charge efficacement les diffusions. Sur d'autres réseaux, tels que les réseaux point à point (voir Figure 4.13) la diffusion au niveau matériel n'est pas toujours très efficacement prise en charge. Dans de tels cas, le diffusion est généralement assurée par le protocole IP qui réplique le datagramme vers toutes les adresses possibles.

Une diffusion dirigée est envoyée vers un réseau spécifique identifié par le champ <Numero-reseau>. Les routeurs du réseau configurés pour transmettre les diffusions dirigés envoient les datagrammes IP au routeur final qui assurera la connexion avec le réseau de destination <Numero-reseau>. Le routeur du réseau de destination

devra alors envoyer le datagramme IP à tous les nœuds du réseau. Si le réseau de destination peut assurer une diffusion dirigée matérielle, cette technique est utilisée pour la diffusion. La Figure 4.14 montre un exemple de diffusion dirigée dans lequel le mécanisme inhérent d'Ethernet est utilisé. Sous Ethernet, une adresse de destination dont tous les bits sont à 1 indique que la trame Ethernet doit être diffusée en diffusion.

Figure 4.13

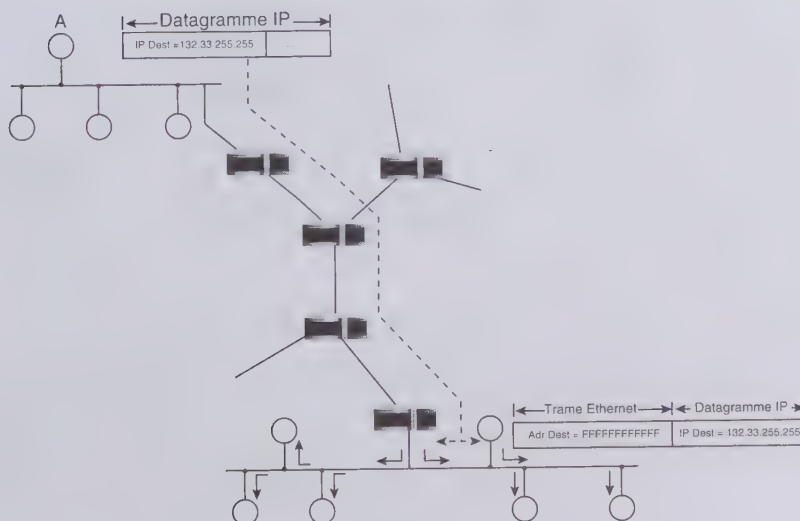
Une diffusion IP sans prise en charge matérielle de la diffusion.



Datagramme IP répliqué sur d'autres liaisons

Figure 4.14

Un exemple de diffusion dirigée.



Certains routeurs sont configurés pour ne pas transmettre les diffusions dirigées dans les cas où la diffusion vers d'autres réseaux n'est pas autorisée.



• Diffusion limitée

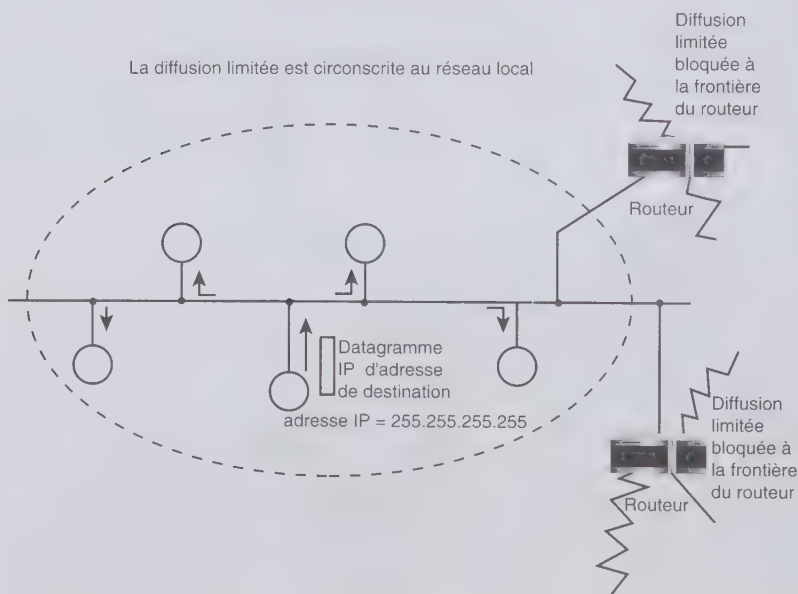
Une diffusion limitée (*limited broadcast*) est une diffusion envoyée à tous les nœuds d'un réseau, la source se trouvant dans le réseau lui-même. La diffusion limitée est représentée par l'adresse 255.255.255.255, ou par la notation :

$\{-1, -1\}$

La diffusion limitée peut être utilisée dans les réseaux locaux, pour lesquels une diffusion ne franchit jamais la frontière du routeur (voir Figure 4.15).

Figure 4.15

Utilisation de diffusions limitées dans des réseaux locaux.



Dans le cas d'une diffusion dirigée, l'émetteur doit fournir la valeur de <Numero-reseau>. Dans le cas d'une diffusion sur le réseau local, vous pouvez utiliser la diffusion limitée qui ne nécessite pas de connaître ce <Numero-reseau>.

Une adresse de diffusion limitée ne peut jamais apparaître comme adresse IP source, mais seulement comme adresse de destination.

• Adresse IP à zéro

Une adresse IP ne comportant que des 0 se note de la façon suivante :

$\{0, 0\}$

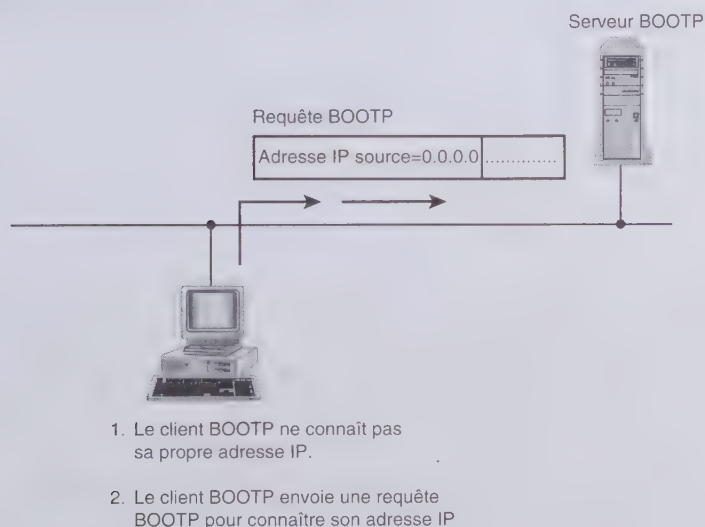
Le champ <Numero-reseau> vaut 0, ce qui signifie "le réseau d'ici". Le champ <Numero-hôte> est aussi nul, ce qui signifie "le nœud d'ici" sur le réseau. Cette

adresse est généralement utilisée lorsqu'un nœud IP essaye de déterminer sa propre adresse IP, comme le fait le protocole BOOTP qui permet aux nœuds d'un réseau de se voir affecter une adresse IP par un serveur BOOTP central.

Lorsque le nœud IP envoie une requête initiale au serveur BOOTP, le nœud IP ne connaît pas sa propre adresse IP (voir Figure 4.16). Il utilise la valeur 0.0.0.0 dans le champ d'adresse IP source pour indiquer "ce nœud-ci" sur "le réseau d'ici". Une fois que le nœud IP prend connaissance de son adresse IP grâce à la réponse de BOOTP, il n'utilise plus l'adresse 0.0.0.0.

Figure 4.16

Utilisation d'une adresse IP 0.0.0.0.



On utilise aussi l'adresse 0.0.0.0 dans les tables de routages pour indiquer l'entrée de réseau de l'adresse IP du routeur par défaut (souvent appelé "passerelle par défaut").

L'adresse IP 0.0.0.0 ne peut être utilisée que comme adresse IP source, jamais comme adresse IP de destination.



• Adresse IP sur "le réseau d'ici"

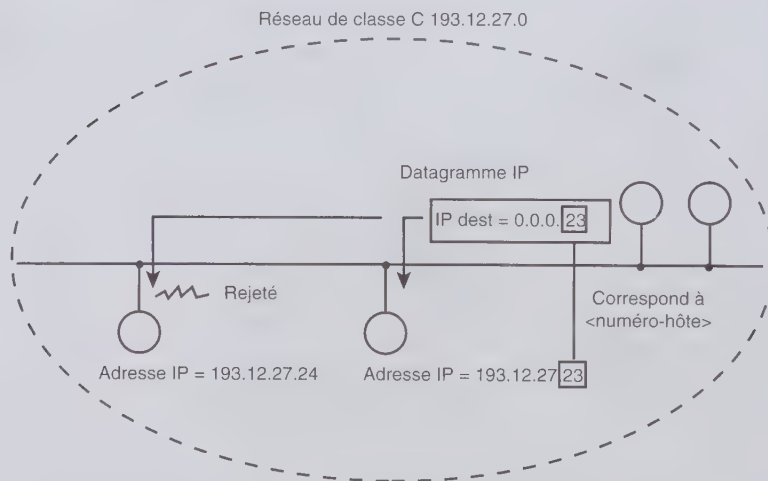
Une valeur nulle de <Numero-reseau> accompagnée d'une valeur non nulle de <Numero-hote> signifie le numéro d'hôte spécifié dans "le réseau d'ici".

{0, <Numero-hote>}

Si un nœud du réseau reçoit un paquet dont l'adresse IP de destination a un <Numero-reseau> nul, mais que le <Numero-hote> est égale au <Numero-hote> du destinataire, ce dernier acceptera le paquet. Le destinataire interprète le <Numero-reseau> nul comme signifiant "le réseau d'ici". La Figure 4.17 montre un exemple où un destinataire reçoit un paquet dont l'adresse IP de destination est 0.0.0.23.

Figure 4.17

Réception d'un datagramme IP pour "le réseau d'ici".



• Bouclage (loopback) logiciel

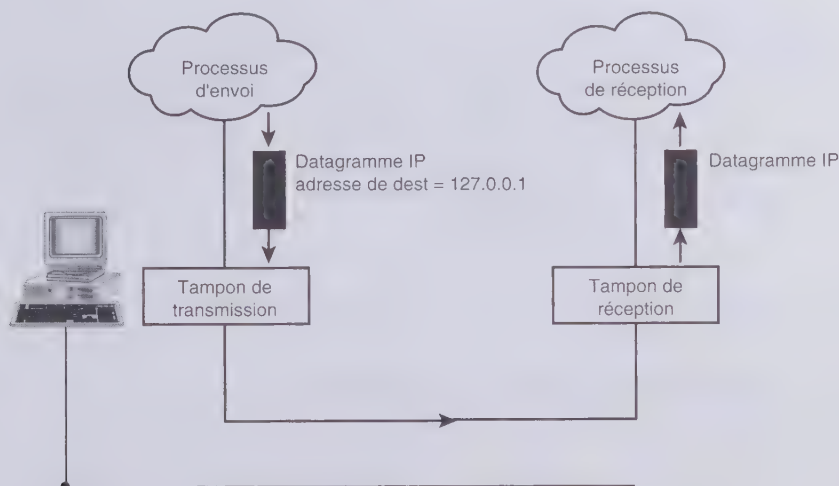
Si vous examinez le Tableau 4.4, vous remarquerez que le nombre 127, qui aurait dû faire partie de l'intervalle des valeurs correspondant à la classe A, n'y figure pas. Ce nombre est réservé à l'adresse de bouclage (*loopback*). L'adresse de bouclage est l'adresse spéciale notée :

{127, <quelconque>}

Tout paquet envoyé par une application TCP/IP vers une adresse de type 127.X.X.X, où X est un nombre entre 0 et 255, a pour conséquence le renvoi de ce paquet à l'application sans que le paquet n'atteigne le support du réseau. Le paquet est copié du tampon (*buffer*) de transmission au tampon de réception sur la machine elle-même (voir Figure 4.18).

Figure 4.18

Dans le cas d'un bouclage logiciel, le paquet n'atteint jamais le support physique du réseau.



Le datagramme IP n'atteint jamais le support physique.

L'adresse de bouclage logiciel permet de vérifier rapidement que le logiciel TCP/IP est bien configuré. On peut ainsi l'utiliser avec l'outil *ping* pour vérifier que la couche IP fonctionne bien. Ping envoie un paquet d'écho ICMP à une adresse IP de destination. La couche IP à cette adresse répond à un paquet d'écho ICMP par un paquet de réponse ICMP.

L'adresse de bouclage logiciel s'avère aussi utile lorsque un client et un service TCP/IP s'exécutent sur une même machine. Vous pouvez accéder au service présent sur la machine en utilisant l'adresse de bouclage. Ainsi, par exemple, vous pouvez utiliser le client Telnet ou FTP pour vous connecter au serveur Telnet ou FTP de votre machine, au moyen des commandes suivantes :

```
telnet 127.0.0.1
```

```
ftp 127.0.0.1
```

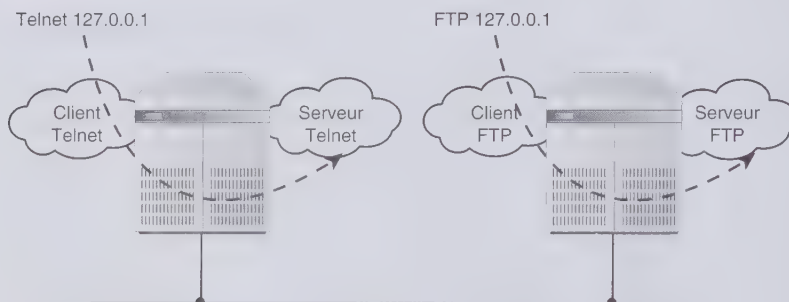
Ces commandes permettent de vérifier l'accès aux serveurs Telnet et FTP, comme le montre la Figure 4.19.

Bien que toute adresse de format 127.X.X.X indique une adresse de bouclage, les adresses de bouclage usuelles sont 127.0.0.1 et 127.1 dans la plupart des implémentations de TCP/IP.



Figure 4.19

Utilisation d'une adresse de bouclage pour accéder à des services locaux.



• Une exception dans l'adressage IP

Une exception notable pour la signification des adresses dont les bits du hostid sont tous à 1, se rencontre dans le cas de logiciels TCP/IP dérivés de l'Unix 4.2 BSD. De nombreuses implémentations commerciales en sont dérivées. Si vous utilisez un logiciel TCP/IP relativement ancien, vous risquez d'être confronté à cette exception.

En Unix BSD, par convention, un champ hostid à 0 signifiait une adresse de diffusion. A l'époque de l'écriture de 4.2 BSD, les RFC ne précisaient pas clairement les conventions utilisées pour les adresses de diffusion. L'ambiguïté fut levée par la suite, et il fut décidé que des hostids dont tous les bits étaient à 1 correspondraient à des adresses de diffusion. 4.3 BSD fut modifié pour se conformer aux nouvelles RFC. Cependant, la plupart des logiciels dérivant de 4.2 BSD utilisent toujours – sauf modification – la convention du diffusion à 0. Si des hôtes utilisant un diffusion à 0 sont placés sur le même réseau physique que des hôtes utilisant le diffusion à 1, le mécanisme de diffusion ne fonctionnera plus comme prévu. En conséquence, les applications TCP/IP sur ce réseau ne fonctionneront plus correctement.

• Unicast, diffusion et multidiffusion

Lorsqu'un datagramme IP est envoyé à une adresse IP individuelle, on dit qu'il s'agit d'un datagramme IP unicast. L'envoi de ce datagramme est appelé *unicasting*. On utilise l'unicasting lorsque deux nœuds IP communiquent ensemble.

Lorsqu'un datagramme IP est envoyé à tous les nœuds d'un réseau spécifique, il s'agit d'une diffusion (*broadcasting*). Dans des sections précédentes, nous avons vu des exemples de diffusions dirigées et limitées.

Il existe un troisième mode d'envoi de datagrammes IP : la multidiffusion (*multicasting*). Dans le cas de la multidiffusion, on utilise une adresse de classe D comme adresse de destination. Le datagramme IP est envoyé à un groupe de systèmes

identifié par une adresse de classe D. Les systèmes qui ont la même adresse de multidiffusion appartiennent à ce que l'on appelle un groupe de multidiffusion. Les membres du groupe de multidiffusion, qui se voient affecter une adresse de classe D, doivent aussi se voir affecter une adresse IP (de classe A, B ou C). Le groupe de multidiffusion peut recevoir un datagramme IP de deux manières :

- des datagrammes IP envoyés directement aux adresses IP individuelles (classes A, B ou C) ;
- des datagrammes IP envoyés à l'adresse de multidiffusion (classe D).

Les adresses de classe D sont celles dont le premier nombre décimal figurant dans leur adresse se trouve dans l'intervalle 224 à 239, bornes comprises (voir Tableau 4.4).

La RFC *Internet Assigned Numbers* ("numéros affectés pour l'Internet") définit des adresses de multidiffusion affectées de manière permanente (voir Tableau 4.5).

Tableau 4.5 : Affectations standards d'adresses de multidiffusion

Adresse de multidiffusion	Description
224.0.0.0	Adresse de base (réservée)
224.0.0.1	Tous les systèmes sur ce sous-réseau
224.0.0.2	Tous les routeurs sur ce sous-réseau
224.0.0.3	Non affecté
224.0.0.4	Routeurs DVMRP
224.0.0.5	Tous les routeurs OSPFIGP
224.0.0.6	Routeurs spécifiés OSPFIGP
224.0.0.7	Routeurs ST
224.0.0.8	Hôtes ST
224.0.0.9	Routeurs RIP2
224.0.0.10	IGRP
224.0.0.11	Agents mobiles
224.0.0.12–224.0.0.255	Non affectés
224.0.1.0	Groupe de gestionnaires VMTP



Tableau 4.5 : Affectations standards d'adresses de multidiffusion (*suite*)

Adresse de multidiffusion	Description
224.0.1.1	NTP (Network Time Protocol)
224.0.1.2	SGI-Dogfight
224.0.1.3	Rwhod
224.0.1.4	VNP
224.0.1.5	Artificial Horizons-Aviator
224.0.1.6	NSS – Name Service Server ("serveur de service de nom")
224.0.1.7	AUDIONEWS – multidiffusion Audio News
224.0.1.8	Service d'information SUN NIS+
224.0.1.9	MTP : Multicast Transport Protocol
224.0.1.10	IETF-1-LOW-AUDIO
224.0.1.11	IETF-1-AUDIO
224.0.1.12	IETF-1-VIDEO
224.0.1.13	IETF-2-LOW-AUDIO
224.0.1.14	IETF-2-AUDIO
224.0.1.15	IETF-2-VIDEO
224.0.1.16	MUSIC-SERVICE
224.0.1.17	SEANET-TELEMETRY
224.0.1.18	SEANET-IMAGE
224.0.1.19	MLOADD
224.0.1.20	Expérimentations privées
224.0.1.21	DVMRP on MOSPF
224.0.1.22	SVRLOC
224.0.1.23	XINGTV
224.0.1.24	microsoft-ds
224.0.1.25	nbc-pro
224.0.1.26	nbc-pfn
224.0.1.27–224.0.1.255	Non affectés
224.0.2.1	Groupe "rwho" (BSD) [non officiel]

Tableau 4.5 : Affectations standards d'adresses de multidiffusion (suite)

Adresse de multidiffusion	Description
224.0.2.2	SUN RPC PMAPPROC_CALLIT
224.0.3.000–224.0.3.255	Service générique RFE
224.0.4.000–224.0.4.255	Conférences individuelles RFE
224.0.5.000–224.0.5.127	Groupes CDPD
224.0.5.128–224.0.5.255	Non affectés
224.0.6.000–224.0.6.127	Projet ISIS de Cornell
224.0.6.128–224.0.6.255	Non affecté
224.1.0.0–224.1.255.255	Groupes multidiffusion ST
224.2.0.0–224.2.255.255	Appels conférences multimédias (<i>Multimedia Conference Calls</i>)
224.252.0.0–224.255.255.255	Groupes transient DIS
232.0.0.0–232.255.255.255	Groupes transient VMTP

Pour que la multidiffusion fonctionne, un hôte doit avoir la capacité de se joindre à un groupe de multidiffusion et de le quitter. Cette capacité est généralement implémentée par une ligne de commande, spécifique au système d'exploitation. Le logiciel de couche IP doit être en mesure de reconnaître les adresses de multidiffusion de datagrammes IP entrants ou sortants. Certaines anciennes implémentations d'IP n'ont pas cette faculté.

N'importe quel hôte sur un interréseau IP peut se joindre à un groupe de multidiffusion. Il n'est pas nécessaire que les hôtes du groupe appartiennent à un même réseau local ; ils peuvent être séparés par des routeurs. En conséquence, les routeurs doivent savoir comment transmettre des datagrammes de multidiffusion. Pour qu'ils le fassent de manière efficace, les routeurs doivent savoir s'il existe des hôtes sur un réseau connecté localement qui fassent partie d'un même groupe de multidiffusion. Comment font-ils pour le savoir ? Les routeurs échangent des informations entre eux afin de déterminer si des membres de groupes se trouvent sur des réseaux distants, afin de leur transmettre les datagrammes IP de multidiffusion.

Les hôtes qui se joignent à un groupe de multidiffusion ou qui le quittent utilisent l'*Internet Group Management Protocol* (IGMP ou "Protocole de gestion de groupe Internet") pour notifier leur appartenance à ce groupe auprès des routeurs voisins. Ces routeurs voisins reçoivent la notification et mettent à jour leurs tables



internes indiquant quels sont les hôtes qui sont membres d'un groupe de multidiffusion. Les routeurs peuvent périodiquement sonder les hôtes pour connaître leurs affiliations courantes. Ces sondages sont envoyés au moyen de l'adresse de multidiffusion spéciale 224.0.0.1 (voir Tableau 4.5) à tous les systèmes du sous-réseau concerné.

Lorsqu'un datagramme de multidiffusion est envoyé vers un réseau local, les capacités de multidiffusion matérielles de ce réseau locales sont utilisées. Nous avons vu au Chapitre 3 que les réseaux locaux Ethernet et IEEE 802 disposent d'un bit de multidiffusion qui peut être activé dans le bit le plus significatif de l'adresse MAC. Les 23 bits de poids faible de l'adresse IP de multidiffusion sont placés dans les 23 bits de poids faible de l'adresse de réseau de multidiffusion Ethernet ou IEEE 802.

L'adresse de multidiffusion de la couche physique utilisée adopte ce format :

```
10000000 00000000 01111001 xxxxxxxx xxxxxxxx xxxxxxxx
```

Les "x" représentent des valeurs quelconques. Le premier "x" est mis à zéro, de sorte que l'adresse devient :

```
10000000 00000000 01111001 0xxxxxxx xxxxxxxx xxxxxxxx
```

L'adresse IP de multidiffusion est une adresse de classe D dont les bits les plus significatifs sont 1110 (voir Tableau 4.3). La représentation binaire d'une adresse de multidiffusion quelconque est la suivante :

```
1110xxxx xxxxxxxx xxxxxxxx xxxxxxxx
```

Les "x" représentent des valeurs quelconques. Seuls les bits de poids faible de l'adresse IP de multidiffusion sont utilisés pour former l'adresse de multidiffusion physique. Ces bits de poids faible sont marqués par des "y" dans le gabarit d'adresse de multidiffusion ci-après :

```
1110xxxx yyyyyyyy yyyyyyyy yyyyyyyy
```

Les bits "x" dans l'adresse de multidiffusion de classe D précédente sont ignorés lors de la formation de l'adresse de multidiffusion physique, qui devient :

```
10000000 00000000 01111001 0yyyyyyy yyyyyyyy yyyyyyyy
```

Les conversions que nous venons de voir permettent de traduire simplement une adresse IP de multidiffusion en une adresse de multidiffusion physique sur un réseau local.

Il convient de remarquer que, comme seuls les 23 bits de poids faible de l'adresse de multidiffusion sont utilisés dans l'adresse de multidiffusion physique, de nombreuses adresses IP de multidiffusion correspondront au final à la même adresse de multidiffusion physique. Prenons l'exemple de l'adresse IP de multidiffusion suivante :

```
1110xxxx yyyyyyyy yyyyyyyy yyyyyyyy
```

Quelles que soient les valeurs des bits "x", l'adresse IP de multidiffusion correspondra à l'adresse de multidiffusion physique unique suivante :

10000000 00000000 01111001 0yyyyyyy yyyyyyyy yyyyyyyy

Comme plusieurs adresses IP de multidiffusion correspondent à une seule adresse de multidiffusion physique, il est possible que l'hôte récupère des datagrammes IP de multidiffusion dont il n'a pas l'usage. C'est pourquoi les bits "x" de l'adresse IP de multidiffusion sont généralement mis à zéro. Dans ce cas, le format de l'adresse IP de multidiffusion est le suivant :

11100000 0yyyyyyy yyyyyyyy yyyyyyyy

En notation décimal à points, cela correspond à des adresses IP dans l'intervalle :

224.0.0.0 à 224.127.255.255

Affectation des adresses IP

Si votre réseau prévoit se connecter à d'autres réseaux tels que l'Internet, vous devez vous adresser à une autorité centrale pour obtenir un netid (numéro de réseau) distinct qui n'est utilisé par personne d'autre. Une fois ce numéro obtenu, il vous incombe d'affecter les numéros d'hôte à partir de votre numéro de réseau. C'est l'*Internet Address Network Authority* (IANA) qui définit les procédures et constitue l'autorité suprême sur les numéros affectés. L'IANA fait partie de l'Internet Registry, lui-même placé sous l'autorité de l'IAB. Dans la pratique, les numéros de réseau sont obtenus auprès de l'InterNIC.

INFO



En France, vous faites appel au NIC ou à un organisme conventionné par ce dernier pour obtenir un nom de domaine. Cependant, les adresses IP se devant d'être uniques sur le réseau Internet, c'est en réalité l'IANA qui fournit des plages d'adresses IP au NIC. Pour plus d'informations, consultez le site du NIC France (<http://www.nic.fr>)

• Adresses de réseau privé

Pour réduire le besoin en nouvelles adresses IP la RFC 1597, sortie en mars 1994, concerne l'allocation d'adresses pour les réseaux privés. Les réseaux privés sont des réseaux qui ne sont pas connectés à d'autres réseaux, ou dont les hôtes et les services ont une interaction limitée avec l'Internet.



Les hôtes sur un réseau peuvent être répartis dans les catégories suivantes :

- Les hôtes qui ne nécessitent pas un accès à des hôtes dans d'autres entreprises ou sur Internet en général.
- Les hôtes qui ont besoin d'accéder à un ensemble limité de services extérieurs tels que l'e-mail, FTP, netnews, la connexion à distance, etc., qui peuvent être traités par une passerelle de couche d'application (voir Chapitre 2).
- Les hôtes qui ont besoin d'accéder à la couche réseau hors de l'entreprise, ce qui requiert une connectivité IP et des périphériques de routage ou des hôtes qui soient visibles de l'extérieur.

Les hôtes de la première catégorie peuvent utiliser des adresses IP distinctes au sein du réseau privé, mais pas forcément sur l'Internet. Comme aucun échange de paquet n'aura lieu entre ces hôtes et le monde extérieur au réseau privé, le problème des doublons d'adresses IP ne se posera jamais.

Les hôtes de la deuxième catégorie, qui sont isolés de l'Internet par une passerelle de couche d'application n'ont pas besoin d'avoir des adresses IP distinctes sur l'Internet. En effet, la passerelle de couche d'Application cache les adresses IP de ces hôtes du point de vue du réseau extérieur.

Les hôtes appartenant à la troisième catégorie nécessitent des adresses IP qui soient globalement distinctes sur l'Internet.

En maintenant correctement une documentation sur votre réseau, vous pouvez facilement identifier les hôtes appartenant aux catégories 1 et 2.

La RFC 1597 donne les exemples suivants pour les hôtes des catégories 1 et 2 :

- Un grand aéroport disposant de nombreux panneaux d'affichage des arrivées et départs, adressables individuellement via TCP/IP. Il est fort peu probable qu'il soit nécessaire d'accéder à ces panneaux directement à partir d'autres réseaux.
- Des organismes de grande taille tels que les banques et des chaînes de magasins, passent à TCP/IP pour leur système de communication interne. Des stations de travail telles que les caisses enregistreuses, les distributeurs de monnaie et l'équipement des employés ont rarement besoin d'une connectivité de ce type.

Pour des raisons de sécurité, de nombreuses entreprises utilisent des passerelles de couche d'Application (tels des *firewalls*) pour connecter leur réseau local à l'Internet. Le réseau interne n'a généralement pas un accès direct à l'Internet et seuls un ou plusieurs hôtes firewall sont visibles depuis l'Internet. Dans ce cas, le réseau interne peut utiliser des numéros IP non distincts. Certaines sociétés proposent des périphériques de NAT (*Network Address Translation* : "traduction d'adresses de réseau") qui peuvent faire correspondre un ensemble d'adresses IP privées non distinctes et un ensemble d'adresses IP distinctes.

Si deux entreprises communiquent par le biais de leur propre liaison privée, seule une portion limitée d'hôte est généralement accessible depuis l'entreprise distante par cette liaison. Seuls ces hôtes ont besoin de numéros IP globalement distincts.

Les interfaces de routeur dans un réseau interne n'ont généralement pas besoin d'être directement accessibles depuis l'extérieur de l'entreprise.

En conformité avec la demande des auteurs de la RFC 1597, l'IANA a réservé pour les réseaux privés les trois blocs suivants de l'espace d'adressage IP. Les adresses correspondantes sont affectées aux hôtes des catégories 1 et 2.

10.0.0.0 à 10.255.255.255	1 réseau de classe A
172.16.0.0 à 172.31.255.255	16 réseaux de classe B
192.168.0.0 à 192.168.255.255	256 réseaux de classe C

Une entreprise qui décide d'utiliser des adresses IP provenant de l'espace d'adressage ci-avant peut le faire sans devoir en référer à l'IANA ou à toute autre organisation de ce type. Ce même espace d'adressage peut donc être utilisé par de nombreuses entreprises ce qui réduit le nombre de numéros de réseaux à attribuer. Les adresses au sein de cet espace d'adressage privé sont distinctes au sein de l'entreprise, mais pas sur le réseau tout entier.

Si une entreprise a besoin d'un espace d'adressage globalement distinct pour des applications et des périphériques connectés sur la couche IP, elle devra se le procurer auprès d'un Internet Registry. L'IANA n'affecte pas d'adresses IP appartenant à l'espace d'adressage IP privé.

Comme les adresses privées n'ont pas de signification globale, le routage des informations concernant les réseaux privés ne doit pas être propagé au-delà de l'entreprise. Les routeurs des réseaux n'utilisant pas l'espace d'adressage privé, tels ceux des fournisseurs d'accès Internet (FAI), doivent avoir des filtres leur permettant de rejeter des informations de routage concernant des réseaux privés. La mise au rebut de ces informations ne doit pas être rapportée comme une erreur du protocole de routage.

Pour utiliser l'espace d'adressage privé, une entreprise doit déterminer quels sont ses hôtes qui n'ont pas besoin de bénéficier d'une connectivité de couche de Réseau. Si la configuration des hôtes change et qu'une connectivité de couche de réseau vers des réseaux extérieurs devient nécessaire, il faudra alors affecter des adresses globalement distinctes. Autrement dit, l'adresse IP de ces hôtes devra changer pour passer dans l'espace d'adressage public. Si l'entreprise ne suit pas soigneusement l'évolution des besoins de ses hôtes, elle risque de se retrouver



confrontée à des problèmes de doublons d'adresses IP. C'est sans doute une des raisons pour lesquelles la RFC 1627 a vu le jour en juillet 1994.

Les auteurs de la RFC 1627, intitulée "Network 10 considered harmful (some practices should'nt be codified)" (Network 10 est estimé dangereux (on ne devrait pas codifier certaines pratiques)), considèrent que les recommandations de la RFC 1597 réduisent la fiabilité et la sécurité du réseau, et peuvent entraîner des problèmes graves. Ainsi, si deux organisations utilisant des espaces d'adressage privés devaient se connecter l'une à l'autre directement, au moins une d'entre elles se verrait obligée de modifier ses affectations d'adresses IP. La RFC 1627 affirme que "la RFC 1597 donne l'illusion de remédier à un problème en offrant un cadre formel à une pratique informelle en usage depuis longtemps. En fait, ce cadre nous cache la nécessité de résoudre ces problèmes bien réels et ne fournit pas même une aide à court terme". De plus, la RFC 1627 affirme que l'IANA a outre-passé son mandat en recommandant la RFC 1597 sans passer par l'examen public usuel ni l'approbation de l'IETF ou de l'IAB.

Et vous, le concepteur de réseau, quelle position allez-vous prendre dans cette polémique sur les adresses privées ? Si vous avez l'intention de suivre les recommandations de la RFC 1597, procédez avec la plus grande circonspection. Veuillez à bien comprendre et à analyser votre réseau pour savoir quels services sont visibles ou non de l'extérieur, et lisez soigneusement la RFC 1627 pour comprendre les problèmes qui pourraient se poser.

• Allocation d'adresses de classe C

Les adresses de classe C, qui vont de 192.0.0 à 223.255.255, ont été divisées en huit blocs. Des autorités régionales sont responsables de l'allocation de ces blocs. Le Tableau 4.6 indique la répartition de ces adresses.

Tableau 4.6 : Allocation des adresses de classe C

Intervalle d'adresses	Région
192.0.0–193.255.255	Plusieurs régions. Comprend des adresses allouées avant la répartition régionale des adresses.
194.0.0–195.255.255	Europe.
196.0.0–197.255.255	Utilisées lorsqu'il est nécessaire d'affecter des adresses IP qui ne sont pas basées sur la région.
198.0.0–199.255.255	Amérique du Nord.
200.0.0–201.255.255	Amérique centrale et Amérique du Sud.
202.0.0–203.255.255	Zone Pacifique

Tableau 4.6 : Allocation des adresses de classe C (suite)

Intervalle d'adresses	Région
204.0.0–205.255.255	Utilisées lorsqu'il est nécessaire d'affecter des adresses IP qui ne sont pas basées sur la région.
206.0.0–207.255.255	Utilisées lorsqu'il est nécessaire d'affecter des adresses IP qui ne sont pas basées sur la région.
208.0.0–223.255.255	Disponibles pour l'affectation.

Les informations d'allocation d'adresses IP qui figurent au Tableau 4.6 peuvent s'avérer utiles si vous souhaitez tracer l'origine de paquets reçus dans votre réseau.

Configuration des adresses IP

Les hôtes d'un réseau TCP/IP doivent être configurés avec des adresses IP correctes. La procédure de configuration elle-même dépend du système d'exploitation. Les divers types de procédures sont les suivants :

- en ligne de commande ;
- interface de menus ;
- interface graphique utilisateur ;
- configuration dynamique lorsque les hôtes démarrent en se référant à un serveur central.

Le Tableau 4.7 montre des exemples de systèmes d'exploitation entrant dans ces différentes catégories.

Tableau 4.7 : Configuration des adresses IP et des paramètres divers pour les hôtes

Méthode	Système d'exploitation/protocole
Ligne de commande	Unix, VMS, périphériques de routage et MS-DOS
Interface de menus	Périphériques de routage, Unix, serveurs Netware et MS-DOS
Interface graphique utilisateur	Produits Unix et Microsoft Windows
Affectation dynamique	Protocoles DHCP et BOOTP, disponibles sur la quasi-totalité des principaux systèmes d'exploitation



De nombreux systèmes proposent une commande qui permet de modifier l'adresse IP. Dans ce cas, la ligne de commande est souvent placée dans le script de démarrage pour le système d'exploitation. L'interface de menus est construite à l'aide de jeux de caractères de dessin étendus. On vous demande d'entrer l'adresse IP et d'autres paramètres IP. L'interface graphique utilisateur existe sur des systèmes tels que X-Window et Microsoft Windows qui fonctionnent sous un affichage graphique au niveau du pixel. L'affectation dynamique d'adresse IP se fait par le biais de protocoles tels que BOOTP ou DHCP. Lors du démarrage, un périphérique demande son adresse IP et d'autres paramètres afférents à un serveur central qui peut fournir ces renseignements en utilisant BOOTP ou DHCP.

Les informations d'adresse IP pour l'hôte sont enregistrées en plusieurs endroits. Lorsque ces informations sont entrées, elles sont mises en antémémoire où elles sont disponibles pour le logiciel TCP/IP. Ces informations peuvent aussi être enregistrées dans un système de fichiers du système d'exploitation. On peut découvrir les adresses IP d'autres systèmes en consultant un fichier spécial, généralement appelé le fichier "hosts", ou en utilisant le protocole DNS. Le service DNS permet de déterminer l'adresse IP d'un hôte à partir de son nom DNS symbolique (voir Chapitre 1). Sur certains systèmes, des protocoles propriétaires permettent de découvrir une adresse IP sur un réseau. C'est le cas, par exemple, du service WINS utilisé dans les systèmes d'exploitation Microsoft.

Vous devez vous reporter aux manuels de votre système d'exploitation pour plus de précisions sur la configuration des adresses IP. L'encadré qui suit constitue un exemple de configuration des adresses IP sur la plupart des implémentations Unix, et sur le système d'exploitation Windows NT.

Supposons que vous configuriez une interface de réseau avec l'adresse IP 144.19.74.102, le masque de sous-réseau 255.255.0.0 et l'adresse de diffusion dirigée 144.19.255.255 (les masques de sous-réseaux sont traités au Chapitre 9).

Pour une configuration Unix :

1. Loguez-vous sous root.
2. Exécutez la commande suivante :

```
ifconfig eth0 144.19.74.102 netmask 255.255.0.0 diffusion  
144.19.255.255
```
3. Remplacez `eth0` par le nom logique Unix de l'interface de réseau.

Pour Windows NT :

1. Loguez-vous sous Administrator.

2. Sélectionnez les options suivantes :

- Démarrage ;
- Paramètres ;
- Panneau de configuration ;
- Réseau ;
- Sélectionnez le protocole TCP/IP ;
- Sélectionnez Propriétés.

3. Vous verrez apparaître alors une boîte de dialogue dans laquelle vous pouvez entrer l'adresse IP et le masque de sous-réseau.

Par défaut, Windows NT utilise l'adresse de diffusion dirigée appropriée 144.19.255.255, en fonction du masque de sous-réseau que vous spécifiez.

Adresses d'IP version 6

Comme nous l'avons vu dans ce chapitre, l'espace d'adressage à 32 bits des adresses IP se révèle trop étroit à mesure du développement de l'Internet. Malgré le RFC 1597, chaque connexion de réseau sur l'Internet nécessite une adresse IP distincte. Certains périphériques disposent de plusieurs connexions réseau, ce qui a pour conséquence un épuisement rapide des adresses IP pouvant encore être attribuées.

On estime que les adresses IP à 32 bits (les adresses IP version 4) peuvent satisfaire plus de 2 100 000 réseaux pour un total de plus de 3 720 millions d'hôtes. Cependant, le plan d'allocation de l'espace d'adressage IP n'est pas très efficace, car il est souvent nécessaire d'allouer de nouveaux numéros de réseau pour la connexion de réseaux par le biais de périphériques de relais de couche de Réseau tels que les routeurs.

Au Chapitre 9, nous verrons les systèmes de sous-réseaux et de surréseaux qui permettent d'utiliser plus efficacement l'espace d'adressage IP. Cependant, même avec l'utilisation de sous-réseaux, le plan d'allocation actuel reste peu efficace. Ainsi, si l'on vous affecte un réseau de classe B, vous pouvez disposer d'un maximum de 65 534 hôtes. Cependant, une organisation de taille moyenne qui ne contient que 15 000 hôtes sur son réseau laissera les 50 534 autres adresses inutilisées, adresses qui ne pourront pas être affectées ailleurs.



On connaîtra bientôt une pénurie d'adresses IP. Pour résoudre ce problème, l'IETF a étudié des méthodes permettant de sortir du cadre de l'espace d'adressage actuel et d'améliorer le protocole existant pour qu'il s'exécute plus efficacement sur les dernières technologies de réseau. Après examen de différentes propositions d'amélioration du protocole IP existant (appelé IP version 4 – ou IPv4), l'IETF a arrêté son choix sur le protocole "IP next generation", aussi appelé IPng ou IP version 6 (IPv6). Le cahier des charges de IPv6 spécifiait que ce protocole devait pouvoir prendre en charge un minimum d'un milliard de réseaux. A cet effet, IPv6 utilise des adresses à 128 bits, soit quatre fois la taille des adresses IPv4. IPv6 conserve le principe des numéros de réseau et des numéros d'hôte, en l'étendant de plusieurs façons. L'adressage hiérarchique de IPv6 permet un routage plus efficace. L'adresse IPv6 peut contenir une adresse IPv4 en plaçant cette adresse (32 bits) dans les bits de poids faible de l'adresse IPv6, et en ajoutant un préfixe fixé de 96 bits. Ce préfixe est constitué de 80 bits à 0 suivis de 16 bits à zéro ou à un.

On peut utiliser en IPv6 des adresses IPv4 non enregistrées en ajoutant un modificateur distinct enregistré, de manière à rendre distincte l'adresse IPv6 dans son ensemble.

IPv6 est conçu pour interopérer avec les systèmes IPv4. Cela assure une transition douce entre les deux systèmes IP. L'objectif à terme est de remplacer tous les systèmes IPv4 par des systèmes IPv6. Les routeurs prenant en charge à la fois IPv4 et IPv6 peuvent être utilisés pour relayer des informations entre des réseaux exécutant le protocole IPv4 et d'autres utilisant déjà IPv6.

IPv6 prend en charge les systèmes portables. Ainsi, les utilisateurs d'ordinateurs et de périphériques portables peuvent se connecter de n'importe où dans le réseau sans devoir effectuer de configuration manuelle. Les systèmes mobiles seront automatiquement connectés lorsqu'ils se joindront au réseau en un nouveau point. Par ailleurs, les stations de travail classiques sont maintenant autoconfigurantes. IPv6 prend en charge le chiffrement au niveau de la couche Internet et prend mieux en charge le trafic en temps réel. Le trafic en temps réel exige une garantie sur le délai maximal de transmission de datagrammes sur le réseau.

Comme les adresses IPv6 font 128 bits, la notation décimale à point devient encombrante. Si on la conservait, il faudrait utiliser une chaîne de 16 nombres décimaux séparés par des points.

Prenons l'exemple de la représentation binaire suivante d'une adresse IPv6 :

```
01011000 00000000 00000000 11000101
11100011 11000011 11110001 10101010
01001000 11100011 11011001 00100111
11010100 10010101 10101010 11111110
```


Comme on le voit, l'écriture binaire n'est pas très pratique. Dans la notation décimale à point, cette adresse aurait la forme suivante :

88.0.0.197.227.195.241.170.72.227.217.39.212.149.170.254

Cette notation n'est pas non plus vraiment compacte. Les concepteurs d'IPv6 ont donc décidé d'utiliser une notation hexadécimale à deux points (:). Les valeurs hexadécimales sont écrites par groupe de 16 bits séparés par le caractère deux points (:). Notre adresse devient alors :

5800:00C5:E3C3:F1AA:48E3:D927:D495:AAFE

La notation hexadécimale à deux points nécessite moins de chiffres et de caractères de séparation. Cette notation propose deux techniques permettant de réduire le nombre de caractères à écrire.

La première technique consiste à omettre les zéros de tête. Prenons l'adresse IPv6 suivante :

48A6:0000:0000:0000:0DA3:003F:0001

En omettant les zéros de tête, cette adresse peut s'écrire :

48A6:0:0:0:DA3:3F:1

Ainsi, 0000 est remplacé par 0, 0DA3 par DA3 et 0001 par 1.

La deuxième technique est la compression des zéros. Cette technique consiste à remplacer une chaîne de zéros qui se répètent en un double deux-points (::). L'adresse IPv6 précédente peut donc s'écrire :

48A6::DA3:3F:1

Pour développer une adresse IPv6 à zéros compressés, il faut aligner tout ce qui se trouve à la gauche du double deux-points avec les mots de 16 bits situés à gauche de l'adresse IPv6. Ensuite, on aligne tout ce qui se trouve à la droite du double deux-points avec les mots de 16 bits de droite de l'adresse. On remplit les mots de 16 bits restant avec des zéros. Prenons, par exemple, l'adresse IPv6 à zéros compressés suivante :

5400:FD45::FFFF:3AFE

Dans sa forme développé, l'adresse devient :

5400:FD45:0:0:0:0:FFFF:3AFE

Vous ne pouvez utiliser le double deux-points qu'une seule fois dans une adresse, sinon l'interprétation peut devenir ambiguë. La notation IPv6 ci-après n'est pas valide, car le nombre de zéros compressés est ambigu.

5400::45A1::23A6



La troisième technique consiste à utiliser les doubles deux points comme préfixe ou suffixe. Prenons la représentation IPv6 de l'adresse IPv4 170.1.1.1 :

0:0:0:0:0:AA01:101

Cette adresse peut aussi s'écrire :

::AA01:101

Récapitulatif

L'affectation d'adresses Internet aux nœuds d'un réseau est une tâche très courante lors de la construction d'un réseau TCP/IP. Il existe deux types d'adresses IP : IP version 4 et IP version 6. Dans ce chapitre nous avons vu les formats de ces deux types. Comme IPv4 est le protocole le plus courant sur l'Internet, le terme "adresse IP" renvoie à une adresse IPv4.

Les adresses IP doivent être distinctes sur un réseau. Dans des cas particuliers, il est possible d'utiliser des adresses non distinctes, appelées des adresses privées, en suivant les principes de la RFC 1597.

Une adresse IP a une structure bien définie : une partie constitue le numéro de réseau (netid) et l'autre un numéro d'hôte (hostid). Toutes les adresses IP ne peuvent pas être affectées comme identifiant distinct de connexions réseau. Seules les adresses A, B et C sont affectables à des connexions réseau individuelles. La classe D est utilisée pour la multidiffusion IP. De plus, il existe plusieurs adresses spéciales utilisées pour la diffusion, le bouclage et d'autres usages particuliers.

Testez vos connaissances

1. Si un hôte possède cinq connexions de réseau sur lesquelles se déroulent des transmissions IP, combien d'adresses IP sont-elles nécessaires ?
2. Quel est le terme OSI pour une adresse IP ?
3. Comparez les avantages et inconvénients de l'utilisation de l'adresse matérielle MAC comme valeur pour l'adresse NSAP.
4. Quel est l'avantage d'utiliser une valeur d'adresse logique pour les adresses IP ?
5. Combien de classes d'adresses existent sous IP ? Nommez les classes d'adresses que l'on peut utiliser comme adresses IP individuelles.
6. A quoi servent les classes d'adresse IP D et E ?
7. Dessinez le format de bits de chaque classe d'adresses IP. Montrez la valeur des bits les plus significatifs et expliquez leur importance.
8. Quel est l'avantage/inconvénient de la division de l'adresse IP en un netid et un hostid ?
9. Prouvez qu'une adresse IP en notation décimale à points dont le premier nombre est compris entre 1 et 126 est une adresse de classe A.
10. Prouvez qu'une adresse IP en notation décimale à points dont le premier nombre est compris entre 128 et 191 est une adresse de classe B.
11. Prouvez qu'une adresse IP en notation décimale à points dont le premier nombre est compris entre 192 et 223 est une adresse de classe C.
12. Prouvez qu'une adresse IP en notation décimale à points dont le premier nombre est compris entre 224 et 239 est une adresse de classe D.
13. Expliquez la différence entre une adresse de diffusion dirigée et une adresse de diffusion limitée. Donnez des exemples de chaque type d'adresse.
14. Qu'est-ce qu'une adresse de bouclage logiciel ? Décrivez son format. Combien existent-il de représentations de l'adresse de bouclage ?
15. Que signifie l'adresse IP 0.0.0.0 ? Donnez des exemples d'utilisation.
16. A quoi sert l'adresse IP 255.255.255.255 ?
17. Sur un réseau de classe B 135.23.0.0, que signifie l'adresse 0.0.45.23 ?



18. Etant donnée une adresse de réseau de classe C valant 193.234.55.0, combien d'hôtes est-il possible de placer sur ce réseau ? Explicitez votre réponse.
19. Etant donnée une adresse de réseau de classe B valant 167.45.0.0, combien d'hôtes est-il possible de placer sur ce réseau ? Explicitez votre réponse.
20. Etant donnée une adresse de réseau de classe A valant 9.0.0.0, combien d'hôtes est-il possible de placer sur ce réseau ? Explicitez votre réponse.
21. Quelle est la différence entre une adresse IP unicast, de diffusion et de multidiffusion ?
22. A quoi sert l'adresse IP 224.0.0.1 ?
23. Etant donnée une adresse IP de multidiffusion 224.0.0.1, donnez la valeur de l'adresse de multidiffusion physique pour un réseau IEEE 802.5.
24. Qu'est-ce qu'une adresse privée ? Donnez l'intervalle de valeurs pour les adresses IP privées. Quelle RFC décrit son utilisation ? Quels sont les problèmes posés par les adresses privées ?
25. Dans quelles circonstances une organisation a-t-elle intérêt à utiliser des adresses privées ?
26. A quel type d'hôte peut-on affecter une adresse IP privée ?
27. Peut-on affecter une adresse IP privée à un hôte ou à un routeur visible du monde extérieur ? Justifiez votre réponse.
28. Quelles autorités affectent des adresses IP en Europe et en Asie ?
29. Déterminez si les adresses IP suivantes sont des adresses spéciales, des adresses IP unicast, des adresses IP de multidiffusion ou des adresses invalides. Spécifiez aussi, le cas échéant, à quelle classe appartiennent ces adresses IP.

33.0.0.45	212.44.45.56	128.0.35.0
212.0.0.0	100.78.189.1	100.0.0.0
0.0.0.0	190.34.0.0	127.33.45.255
255.255.255.255	10.255.255.255	137.256.34.0
127.1	33.55.260.35	127.0.0.1
0.0.22.45 dans le réseau 156.23.0.0		
0.67.43.90 dans le réseau 7.0.0.0		
0.0.0.252 dans le réseau 99.25.66.0		

30. Quelle est la limite théorique pour le nombre d'adresses IP de multidiffusion ?

31. Si seuls les 23 bits de poids faible d'une adresse IP de multidiffusion sont mappés sur une adresse de multidiffusion physique, combien d'adresses IP de multidiffusion peuvent potentiellement être mappées sur la même adresse de multidiffusion physique ? Cela peut-il poser des problèmes ? Si oui, quelles restrictions doivent s'appliquer à l'adresse IP de multidiffusion pour résoudre les problèmes ?
32. Etant donné le mécanisme décrit dans ce chapitre pour mapper une adresse IP de multidiffusion sur une adresse de multidiffusion physique, combien d'adresses de multidiffusion physiques sont-elles possibles ?
33. Décrivez les méthodes permettant d'entrer les informations concernant l'adresse IP sur un hôte.
34. Exprimez en notation hexadécimale à deux points les adresses IPv6 suivantes :
- 01011000 10101011 01111001 10111101
10100011 11011011 11110011 11110010
11001000 10000011 10011001 10101111
01010100 10111101 11101110 10000001
35. Convertissez les adresses IPv4 suivantes en adresses IPv6. Indiquez à la fois les notations hexadécimales à deux points compressées et développées.
- | | |
|-----------------|----------------|
| 127.0.0.1 | 199.245.180.10 |
| 255.255.255.255 | 167.99.44.33 |



5

Protocoles de résolution d'adresse

Le plan d'adressage IP est un plan logique conçu pour créer l'apparence d'un réseau virtuel. Toutes les interfaces du réseau sont modélisées par un unique identificateur à 32 bits appelé l'adresse IP. Toutefois, la transmission des datagrammes IP sur le réseau physique nécessite que ces datagrammes soient encapsulés dans des trames de couche de liaison de données (couche OSI 2). Les trames de couche de liaison de données, telles que les trames Ethernet ou d'anneaux à jetons, doivent contenir des adresses matérielles. Ce chapitre traite des techniques utilisées dans les réseaux TCP/IP pour créer l'association entre adresses matérielles et adresses IP. Les protocoles permettant cette association sont appelés les protocoles de résolution d'adresse.

Nécessité de la résolution d'adresse

Le service DNS est utilisé pour fournir une association entre les noms symboliques attribués à l'hôte et l'adresse IP. En connaissant le nom symbolique, un hôte peut découvrir l'adresse IP correspondante. Pour transmettre un message à un hôte d'un réseau doté de la capacité d'effectuer des diffusion, comme c'est le cas pour Ethernet et les anneaux à jetons, l'émetteur doit connaître l'adresse matérielle de l'hôte de destination. L'adresse matérielle, aussi appelée adresse MAC (*Media Access Control*, "couche d'accès au support"), doit figurer dans l'en-tête MAC du paquet utilisé pour envoyer le message. Le logiciel hôte connaît l'adresse IP de destination en utilisant DNS ou par consultation de table. Le problème consistant à déterminer l'adresse matérielle de l'hôte de destination est illustré à la Figure 5.1.

Une méthode pour résoudre le problème consistait à utiliser une table d'associations entre adresses IP et adresses MAC, similaire au fichier *hosts* vu précédemment. Le problème de cette méthode est que si une carte réseau est remplacée sur un hôte, l'adresse MAC change et la table doit être mise à jour. Lorsqu'un technicien change la carte réseau, il est rare qu'il en informe immédiatement l'administrateur réseau.

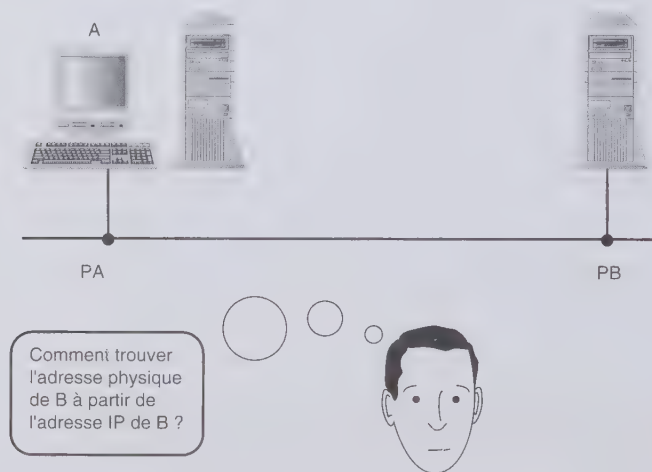
La configuration manuelle des correspondances entre adresses IP et adresses MAC fut une pratique courante sur les systèmes les plus anciens. Dans le cas de systèmes modernes, un mécanisme plus souple est nécessaire pour déterminer



dynamiquement l'adresse MAC à partir de l'adresse IP d'un hôte. Ce mécanisme dynamique est implémenté sous forme d'un protocole distinct, appelé ARP (*Address Resolution Protocol*, "protocole de résolution d'adresse").

Figure 5.1

Nécessité de la résolution d'adresse (source : *Learning Tree*)



Address Resolution Protocol (ARP)

La Figure 5.2 montre un schéma simplifié du fonctionnement d'ARP. Dans cette figure, l'hôte A souhaite déterminer l'adresse matérielle de la destination B avant d'envoyer son message.

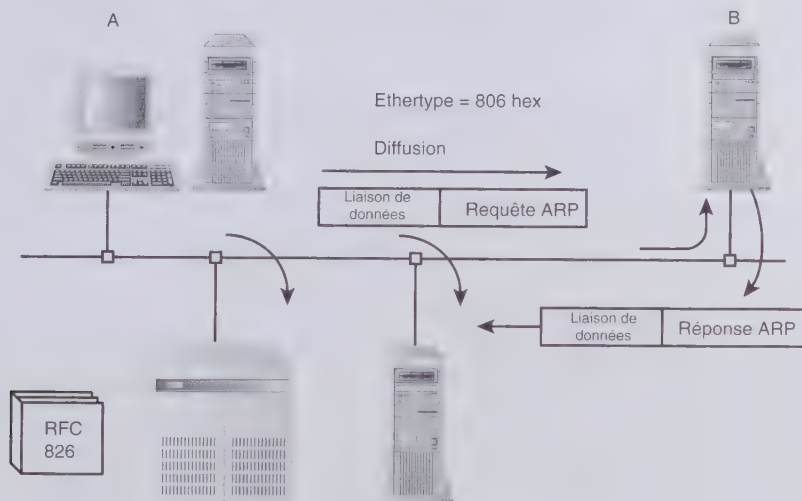
L'hôte A envoie sur le réseau une trame MAC de diffusion appelée la trame ARP de requête. La "trame ARP de requête" contient les adresses IP et MAC de l'hôte A émetteur, ainsi que l'adresse IP de la destination B. La trame de requête ARP inclut un champ destiné à contenir l'adresse matérielle de B. Tous les nœuds du réseau physique reçoivent la trame de requête ARP de diffusion. Ces nœuds comparent leur adresse IP aux adresses IP figurant dans la requête ARP. Seul l'hôte dont l'adresse IP correspond à l'adresse requise dans la trame de requête ARP répond.

Si l'hôte B existe sur le réseau, il répond en encodant sa propre adresse matérielle dans une trame de réponse ARP. L'hôte A initialise alors sa table de cache ARP (conservée en mémoire) en utilisant la réponse fournie par la réponse ARP. Les entrées de cache ARP expirent après une temporisation donnée qui peut être

configurée dans certaines implémentations de TCP/IP. Généralement, la temporisation du cache ARP est de 15 minutes. Une fois qu'une entrée de cache ARP pour un hôte donné a expiré, une nouvelle trame de requête ARP est envoyée pour découvrir l'adresse matérielle de l'hôte. Le cache ARP est consulté par un hôte juste avant l'envoi d'une requête ARP. Si la réponse se trouve dans le cache, l'hôte peut ainsi se passer d'envoyer la requête.

Figure 5.2

Fonctionnement d'ARP
(source : Learning Tree).



La requête ARP est envoyée avec une adresse matérielle de diffusion, car l'adresse matérielle du nœud cible n'est pas connue. La réponse ARP envoyée par le nœud cible *n'est pas* une trame de diffusion. En effet, le nœud cible prend connaissance de l'adresse matérielle de l'émetteur en examinant le paquet ARP. La réponse ARP est, par conséquent, directement envoyée au nœud qui a émis la requête ARP.

Le protocole ARP part du principe que le réseau physique sous-jacent prend en charge la diffusion. C'est le cas des réseaux locaux tels qu'Ethernet, les anneaux à jetons, FDDI et ARCnet.

Les sections qui suivent détaillent le format ARP et le fonctionnement du protocole.



• Le format ARP

La Figure 5.3 illustre le format des paquets de requêtes et de réponses ARP. A l'origine, le protocole ARP a été conçu pour la version DIX d'Ethernet (voir Chapitre 2). Par la suite, il a été étendu à d'autres réseaux locaux tels que les réseaux IEEE 802. Le paquet ARP est encapsulé dans la trame de couche de liaison de données du réseau. La Figure 5.4 montre un exemple de paquet ARP incorporé dans une trame Ethernet. La valeur hexadécimale 806 d'EtherType est réservée pour les trames ARP.

Figure 5.3

Format du paquet ARP
(source : Learning Tree).

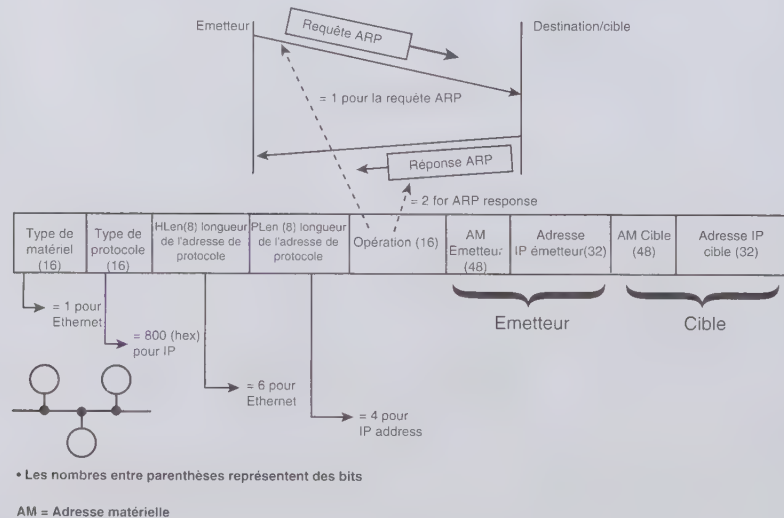
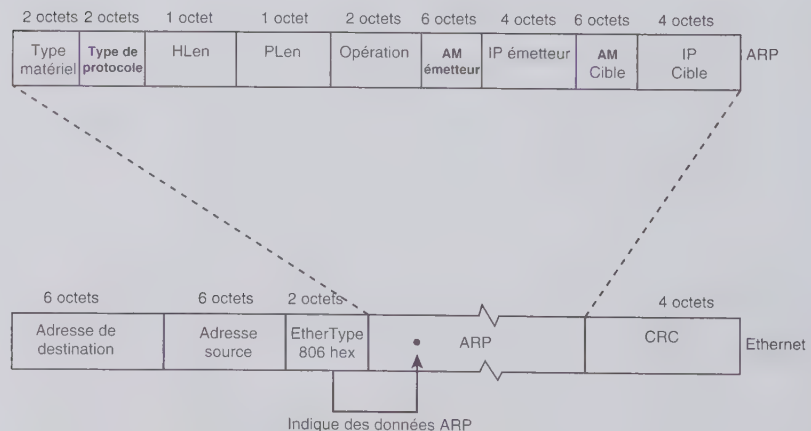


Figure 5.4

Un paquet ARP encapsulé
dans une trame Ethernet.



Cette section définit le format ARP en détaillant chacun des champs du paquet ARP (sur la Figure 5.3, les nombres entre parenthèses représentent la taille en bits des champs ARP).

Le champ type de matériel est le premier champ du paquet ARP (voir Figure 5.3). Sa longueur est de deux octets.

La valeur dans ce champ type de matériel détermine la configuration du champ HLen. Une valeur de 1 dans le champ type de matériel indique un réseau Ethernet. Le Tableau 5.1 dresse la liste des codes de types de matériel pour différents réseaux.

Tableau 5.1 : Valeurs pour le champ ARP type de matériel

Valeur de type de matériel	Description du réseau
1	Ethernet (10 Mbit/s)
2	Ethernet expérimental (3 Mbit/s)
3	Radio amateur AX.25
4	Anneau à jetons Proteon ProNET
5	Chaos Net
6	Réseau IEEE 802
7	ARCNET
8	Hyperchannel
9	Lanstar
10	Autonet Short Address
11	LocalTalk
12	LocalNet
13	Ultra link
14	SMDS
15	Frame Relay (relais de trames)
16	Asynchronous Transmission Mode (ATM)
17	HDLC



Tableau 5.1 : Valeurs pour le champ ARP type de matériel (*suite*)

Valeur de type de matériel	Description du réseau
18	Fibre Channel
19	Asynchronous Transmission Mode (ATM)
20	Ligne série
21	Asynchronous Transmission Mode (ATM)

Le Type de protocole est l'adresse de protocole de couche supérieure pour laquelle l'adresse matérielle est désirée. Ce champ fait deux octets. Dans le cas des réseaux TCP/IP, l'adresse de protocole est l'adresse IP pour laquelle on souhaite connaître l'adresse matérielle. La valeur du type de protocole est la même que la valeur EtherType pour le protocole. Le Tableau 2.2 du Chapitre 2 dresse une liste des valeurs EtherType usuelles pour divers protocoles. Pour le protocole IP, la valeur hexadécimale 800 est utilisée pour le champ Type de protocole.

Le champ HLen correspond à la longueur en octets de l'adresse matérielle. Ce champ fait un octet. La valeur de HLen est déterminée par le champ Type de matériel. Si le type de matériel est 1, cela indique un réseau Ethernet dont l'adresse matérielle fait 8 octets. En conséquence, la valeur du champ HLen est 6 dans le cas de réseaux Ethernet. Le champ HLen détermine la taille des champs "AM émetteur" (*Sender HA*) et "AM destination" (*destination HA*) [*HA*, *Hardware Address* – et *AM* – est une abréviation usuelle pour les adresses matérielles dans les protocoles TCP/IP]. Dans le cas de réseaux Ethernet, ces champs font 6 octets de long.

Le champ PLen est la longueur de l'adresse de protocole en octets. Ce champ fait 1 octet. La valeur de PLen est déterminée par la valeur du champ Type de protocole. Si le Type de protocole est 800 en hexadécimal, cela indique le protocole IP dont l'adresse IP fait 4 octets. En conséquence, la valeur du champ PLen est 4 pour les réseaux IP. Le champ PLen détermine la taille des champs IP émetteur et IP destination. Dans le cas de réseaux IP, ces champs font 4 octets.

Le champ Opération fait deux octets et indique si le paquet contient une requête ARP ou une réponse ARP. Le Tableau 5.2 indique les différentes valeurs possibles dans ce champ. Certaines des valeurs figurant dans le tableau sont réservées pour des protocoles expérimentaux ou des protocoles de résolution d'adresse encore en projet, et vous avez peu de chance de les voir sur des réseaux existants.

Le champ AM émetteur contient l'adresse matérielle du nœud envoyant la requête ARP. L'émetteur connaît sa propre adresse matérielle en la lisant sur la carte réseau et remplit ce champ avec cette adresse.

Tableau 5.2 : Valeurs du champ Opération d'un paquet ARP

Valeur du champ	Type d'opération
1	Requête ARP
2	Réponse ARP
3	Requête RARP
4	Réponse RARP
5	Requête DRARP
6	Réponse DRARP
7	Erreur DRARP
8	Requête InARP
9	Réponse InARP
10	NAK ARP

L'adresse IP émetteur contient l'adresse IP du nœud envoyant la requête ARP. L'émetteur connaît son adresse IP en la lisant dans un fichier de configuration ou dans son cache mémoire qui contient des informations de configuration de réseau.

Le champ AM cible contient l'adresse matérielle de la cible. Cette valeur n'est pas connue par l'émetteur dans le cas d'une requête ARP, puisque c'est précisément pour connaître cette valeur qu'on envoie une requête ARP. Ce champ est souvent entièrement rempli de zéros ou de uns. STD37 (RFC826 aussi) déclare, concernant les protocoles de résolution d'adresse, que le champ AM cible n'a pas besoin de recevoir une valeur particulière. Ce champ peut recevoir l'adresse de diffusion pour le matériel (tout à 1 dans le cas d'Ethernet), si cela simplifie les choses du point de vue de l'implémentation.

Le champ IP cible de la requête ARP contient l'adresse IP du nœud dont on souhaite déterminer l'adresse matérielle. Cette valeur est fournie par l'émetteur de la requête ARP.

Le paquet de réponse ARP utilise le même format que le paquet de requête ARP, mais le champ Opération reçoit la valeur 2 pour indiquer qu'il s'agit d'une réponse (voir Tableau 5.2).

L'utilisation d'un même format pour les requêtes et les réponses ARP permet à l'implémentation de réutiliser le buffer de paquet de requête pour la réponse ARP.



La réponse ARP est de même longueur que la requête ARP, et plusieurs des champs sont identiques.

Lorsqu'un nœud IP envoie une réponse ARP, il place son adresse matérielle dans le champ AM émetteur. Notons que la valeur de ce champ AM émetteur du paquet de réponse ARP contient la réponse à la question posée, c'est-à-dire l'adresse matérielle du nœud cible. Les champs du paquet de réponse ARP sont remplis ainsi :

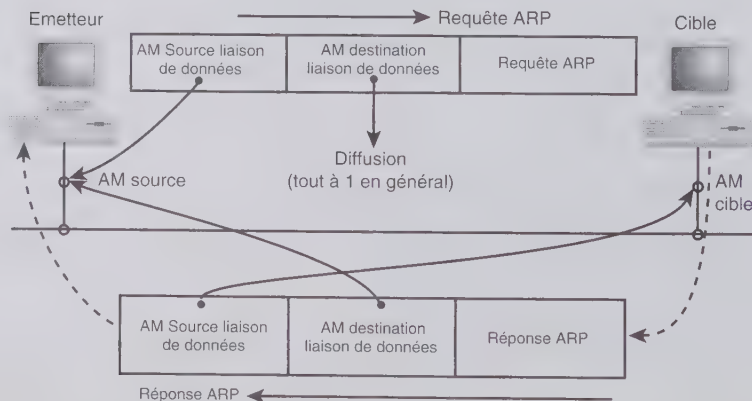
- **AM émetteur.** Contient l'adresse matérielle du nœud cible (l'information désirée par la requête).
- **IP émetteur.** Contient l'adresse IP du nœud cible.
- **AM cible.** Contient l'adresse matérielle de l'émetteur de la requête ARP. Le nœud cible connaît cette adresse en examinant le champ AM émetteur de la requête ARP.
- **IP cible.** Contient l'adresse IP de l'émetteur de la requête ARP. Le nœud cible connaît cette adresse en examinant le champ IP émetteur de la requête ARP.

Dans les trames de liaison de données qui servent à envoyer la requête ARP et la réponse ARP, les champs d'adresses sont remplis comme suit (voir Figure 5.5) :

1. Paquet de requête ARP :
 - **AM source liaison de données.** Valeur lue sur la carte réseau de l'émetteur de la requête ARP.
 - **AM destination liaison de données.** Diffusion. Généralement tout à 1.
2. Paquet de réponse ARP :
 - **AM source liaison de données.** Valeur lue sur la carte réseau du nœud cible qui génère la réponse ARP.
 - **AM destination liaison de données.** Valeur lue dans le champ AM émetteur de la requête ARP.

Figure 5.5

Adresses de liaison de données pour les requêtes et réponses ARP.

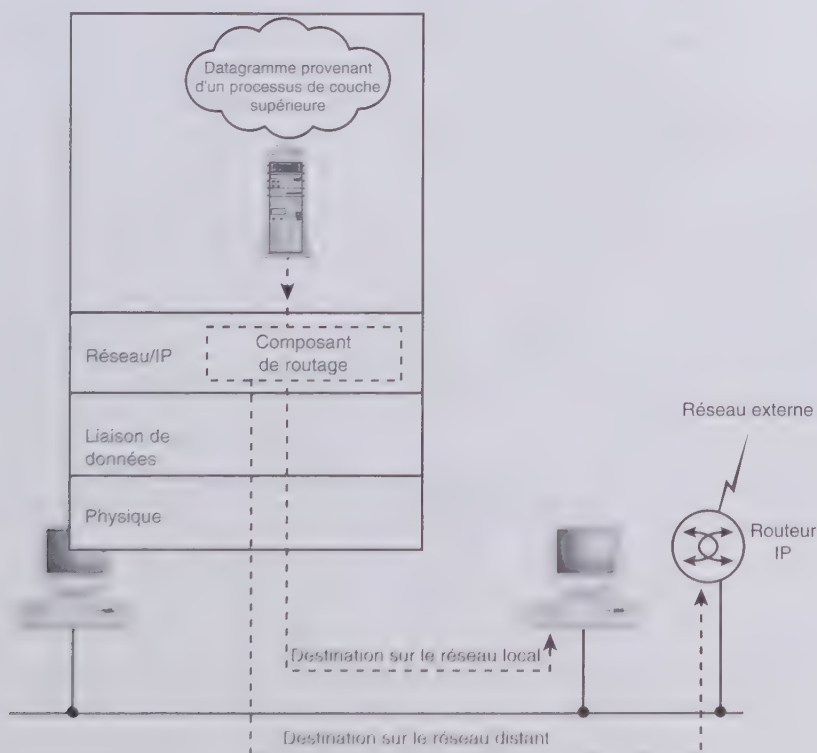


Notons que les paquets de requêtes et de réponses ARP ont des tailles variables, car la taille des champs d'adresse matérielle et d'adresse de protocole dépend du type de matériel et de protocole présents sur le réseau.

• Fonctionnement d'ARP

Lorsqu'un paquet IP est envoyé à travers les couches de réseau des nœuds d'un interrésseau TCP/IP, la composante de routage de la couche de réseau détermine l'adresse IP du routeur suivant (voir Figure 5.6). La composante de routage de la couche de réseau détermine si le datagramme IP doit être envoyé à un hôte du réseau local ou à un hôte situé sur un autre segment de réseau. L'algorithme de routage employé par un hôte émetteur est détaillé au Chapitre 7.

Figure 5.6
*La composante de routage
de la couche de réseau de l'hôte
émetteur.*



Si l'hôte émetteur détermine que la destination se trouve sur le réseau local, il doit trouver l'adresse matérielle de la destination. Si la destination se trouve sur un réseau distant (différent donc du réseau local), l'émetteur doit trouver l'adresse matérielle du port du routeur auquel le datagramme IP sera transmis.

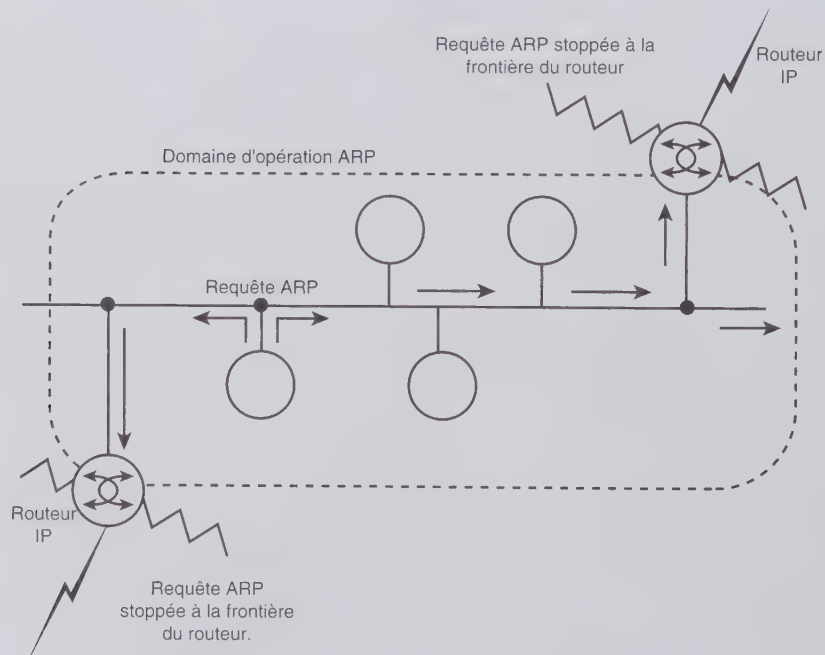
Pour les réseaux dotés de la diffusion, on utilise ARP pour déterminer l'adresse matérielle du nœud cible, qui est soit le nœud sur le réseau local, soit le port du routeur sur le même réseau local.

ARP est implémenté au sein du module IP dans les réseaux qui ont besoin de la résolution d'adresse, tels que les réseaux locaux à diffusion (Ethernet, anneaux à jetons, etc.). Le protocole ARP est directement encapsulé par le protocole de couche de liaison, et pas par le protocole IP (voir Figure 5.7). Cela signifie que le protocole ARP ne peut être routé – c'est-à-dire qu'il ne peut franchir un routeur –, car la transmission d'un paquet par un routeur IP nécessite une encapsulation IP.

Avant d'envoyer la requête ARP, le module ARP essaye de trouver l'adresse cible dans son cache mémoire local, aussi appelé "table de cache ARP". La table de cache ARP conserve une liste de paires d'adresses IP et d'adresses matérielles correspondantes (voir Figure 5.8).

Figure 5.7

La limitation d'ARP au segment de réseau local.



Si l'adresse IP cible est trouvée dans la table de cache ARP, on prend connaissance de l'adresse matérielle correspondante, adresse qui est alors renvoyée au module ARP. Le module ARP renvoie l'adresse matérielle au pilote de réseau qui a formulé la requête pour connaître l'adresse matérielle du nœud cible. Le pilote de réseau transmet alors une trame de couche de liaison de données, contenant le datagramme IP avec dans le champ d'adresse matérielle de destination l'adresse matérielle du nœud cible. Dans le cas de figure que nous venons de voir, une requête ARP n'est jamais générée, car l'adresse matérielle du nœud cible figure déjà dans le cache mémoire local.

Que se passe-t-il si le cache ARP ne contient pas l'adresse matérielle du nœud cible ? Dans ce cas, le module ARP génère une trame de couche de liaison de données contenant la requête ARP permettant de déterminer cette adresse. La requête ARP est envoyée en diffusion à tous les nœuds du segment de réseau local. Comme nous l'avons vu précédemment, les requêtes ou réponses ARP sont confinées au segment de réseau local et ne franchissent pas les routeurs.

Figure 5.8

La table de cache ARP.

Type de protocole (IP)	Adresse de protocole (adresse IP)	Adresse matérielle	Numéro d'interface	Time Stamp



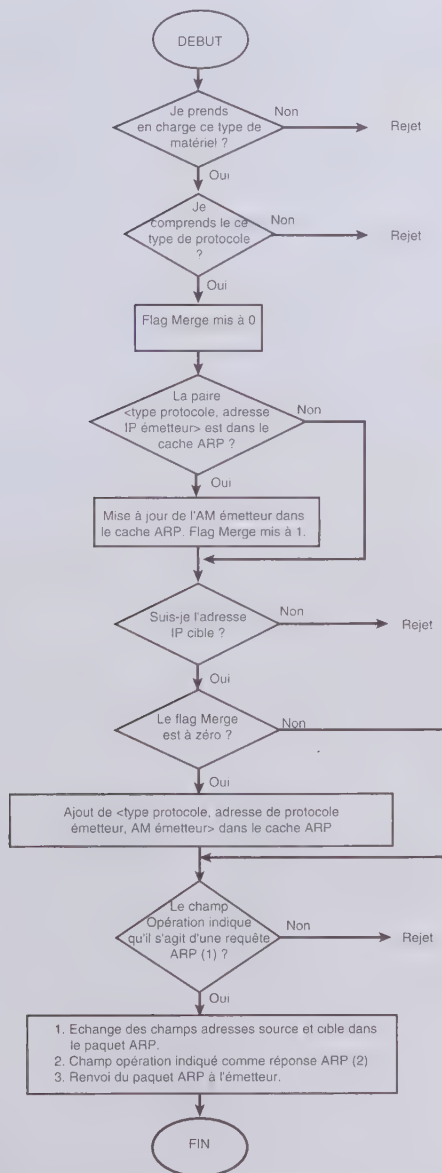
Lorsqu'une requête ARP est reçue, la couche de liaison de données au niveau du nœud cible donne le paquet au module de résolution d'adresse. La requête ARP est envoyée en diffusion au niveau de la couche de liaison de données, de sorte que tous les nœuds du réseau local la reçoivent. Toutefois, seul le nœud dont l'adresse IP correspond à l'adresse IP cible répond par une réponse ARP.

Les étapes suivies par le module ARP au niveau du nœud cible sont les suivantes (voir aussi l'organigramme à la Figure 5.9) :

1. Est-ce que je reconnais le type de matériel qui figure dans le champ Type de matériel du paquet de requête ARP ?
2. Si la réponse de l'étape 1 est "oui", on vérifie éventuellement le champ HLen dans le paquet de requête ARP.
3. Est-ce que je sais comment traiter le protocole figurant dans le champ Type de protocole du paquet de requête ARP ?
4. Si la réponse est "oui" à l'étape 3, on vérifie éventuellement le champ PLen dans le paquet de requête ARP.
5. On définit Merge-flag comme *false* pour contrôler le traitement ARP dans la suite du processus.
6. Si la paire <type de protocole, adresse de protocole de l'émetteur> se trouve déjà dans ma table de traduction, on met à jour le champ d'adresse matérielle émetteur de l'entrée avec les nouvelles informations présentes dans le paquet et on affecte *true* à Merge-flag.
7. L'adresse de protocole cible me correspond-elle ? (Cela signifie que mon adresse IP est la même que celle du champ Adresse IP cible.)
8. Si la réponse est oui à l'étape 7, on poursuit en regardant si Merge-flag est *false*. Si c'est le cas, on ajoute le triplet <type de protocole, adresse de protocole de l'émetteur, adresse matérielle de l'émetteur> à la table de traduction.
9. La valeur du champ Opération correspond-elle à celle d'une requête ?
10. Si la réponse est oui à l'étape 9, on poursuit en inversant les champs matériel et protocole, en plaçant les adresses matérielle et de protocole dans les champs émetteurs. On indique dans le champ Opération qu'il s'agit d'une réponse ARP. On envoie le paquet à la nouvelle adresse matérielle cible à partir de l'adresse matérielle qui a reçu le paquet de requête.

Dans l'algorithme ci-avant, le triplet <type de protocole, adresse de protocole émetteur, adresse matérielle de l'émetteur> est intégré à la table de cache ARP avant que le champ Opération ne soit examiné lors de l'étape 6. Cette étape ne survient que si le nœud dispose dans sa table de cache ARP d'une entrée pour l'adresse IP émetteur.

Figure 5.9
*Organigramme de réception
 d'un paquet ARP.*



La Figure 5.10 montre comment une table de cache ARP est modifiée par une unique requête ARP. Seule la table de cache ARP pour les nœuds disposant d'une entrée pour l'adresse IP émetteur, est modifiée.

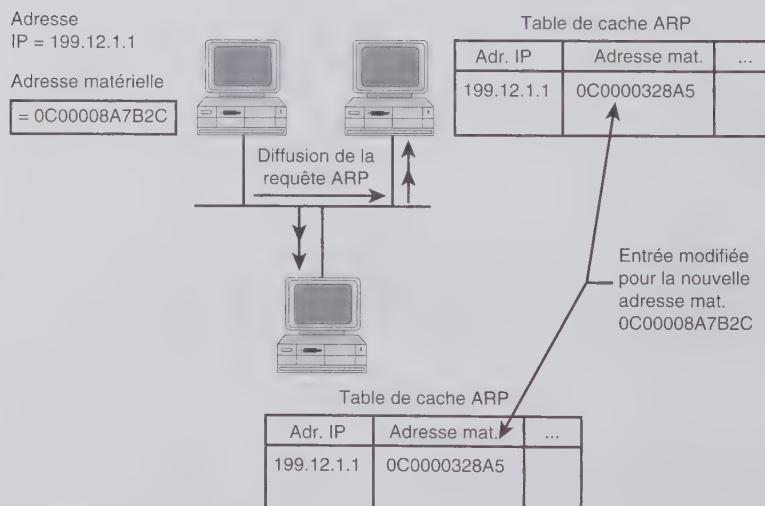
La mise à jour ou l'ajout d'entrées de cache ARP se fait parce que les couches supérieures du nœud cible vont probablement répondre à l'émetteur de la requête ARP. Lorsque la cible génère une trame de couche de liaison de données, elle consulte son cache ARP et reconnaît l'adresse matérielle de l'émetteur. Il est donc inutile d'envoyer une requête ARP pour la connaître.

Notez que si le nœud cible possède déjà une entrée dans sa table de cache ARP pour la paire <type de protocole, adresse de protocole de l'émetteur>, la nouvelle adresse matérielle remplace l'ancienne. Cela peut survenir dans les situations suivantes :

1. Un nouveau matériel de réseau a été ajouté sur l'émetteur de la requête ARP pour remplacer un matériel existant. Dans ce cas, l'adresse matérielle du nœud émetteur change.
2. Les adresses IP ont été réaffectées. L'émetteur de la requête ARP est un autre nœud qui s'est vu affecter l'adresse IP se trouvant dans la table de cache ARP de la cible. L'adresse matérielle de l'émetteur ARP est différente parce qu'il s'agit d'un autre nœud.
3. Il y a un problème de doublon d'adresses. Un autre nœud affirme posséder la même adresse IP que celle de la table de cache ARP du nœud cible.

Figure 5.10

Une requête ARP provoquant des modifications dans la table de cache ARP, conformément à la RFC 826.



Si les deux premières situations sont normales, la troisième – le doublon d'adresses – est une condition d'erreur. On la traite en détail dans la section "Doublons d'adresses IP" plus loin dans ce chapitre.

• Commentaires sur la conception du protocole ARP

En théorie, les champs de longueur HLen et PLen dans le paquet ARP sont redondants, car la longueur de l'adresse matérielle et celle de l'adresse de protocole peuvent être déterminées à l'aide de la valeur des champs Type de matériel et Type de protocole. Ainsi, si le champ Type de matériel vaut 1, cela indique une trame Ethernet (voir Tableau 5.1), ce qui implique une valeur de six octets pour HLen, car les adresses Ethernet font six octets. De même, si le champ Type de protocole indique IP, PLen vaut quatre octets, car les adresses IP font quatre octets.

Les champs redondants HLen et PLen sont inclus pour une vérification facultative de la cohérence et pour l'utilisation d'outils de surveillance de réseau.

Le champ Opération permet de déterminer si le paquet est une requête ou une réponse à une requête antérieure. L'utilisation de deux octets pour ce champ peut sembler excessive étant donné le nombre de codes d'opération définie jusqu'à présent (voir Tableau 5.1). Un champ d'un octet aurait été amplement suffisant.

Les champs AM émetteur et IP émetteur sont nécessaires, car ils sont susceptibles d'être enregistrés dans les tables de cache ARP des nœuds récepteurs. Dans le cas de la requête ARP, le champ AM cible n'a pas de signification, car sa valeur n'est pas encore connue. La réponse est en fait renvoyée dans le champ AM émetteur de la requête ARP. Le champ AM émetteur est inclus par souci d'exhaustivité et pour la surveillance du réseau.

Le champ IP cible est nécessaire dans la réponse ARP pour permettre à une machine de déterminer si elle est ou non le nœud cible et doit en conséquence envoyer une réponse. Le champ IP cible n'est pas forcément utile dans la réponse ARP si l'on considère qu'une réponse n'est provoquée que par une requête. Ce champ peut servir pour surveiller le réseau et simplifier l'algorithme de traitement des requêtes/réponses ARP.

• Surveillance du réseau avec ARP

ARP peut être utilisé par un périphérique de surveillance pour obtenir des informations sur l'activité des protocoles de plus haut niveau, par un simple examen des en-têtes des trames ARP. Ainsi, le périphérique de surveillance peut déterminer



quels sont les protocoles utilisés en examinant la valeur du champ Type de protocole. Cette section montre comment concevoir un tel dispositif de surveillance.

Lorsque le périphérique de surveillance reçoit un paquet ARP, il peut entrer dans une table le type de protocole, l'adresse de protocole de l'émetteur et l'adresse matérielle de l'émetteur. De plus, il peut déterminer la longueur des adresses matérielle et de protocole en regardant les champs HLen et PLen du paquet ARP. Notez que les requêtes ARP sont envoyées en diffusion au niveau de la couche de liaison de données, et qu'un périphérique de surveillance reçoit donc toutes les requêtes ARP.

Si le champ Opération indique qu'il s'agit d'un paquet de réponse ARP, et que l'adresse de protocole cible correspond à l'adresse de protocole du périphérique de surveillance, ce dernier envoie une réponse ARP comme le ferait tout autre nœud. Le périphérique de surveillance ne reçoit aucune autre réponse ARP, car une réponse ARP est directement envoyée à l'hôte qui a fait la requête (pas de diffusion).

• Temporisations dans la table de cache ARP

Les tables de cache ARP sont souvent implémentées avec un mécanisme de temporisation (*timeout*). Les entrées du cache ARP se voient ainsi associées à des marques de temps (*timestamps*). Dans cette section, nous aborderons les différentes questions liées à l'utilisation des temporisations.

Si un nœud est déplacé, il est généralement fermé avant son déplacement. La fermeture d'un nœud provoque l'effacement de sa table de cache ARP afin que des entrées obsolètes ne soient pas prises en compte. Les autres nœuds ne sont généralement pas avertis qu'un nœud s'est déplacé ou a été fermé.

Si le nœud a été déplacé au sein du même segment de réseau, les informations de cache ARP concernant ce nœud restent valides, car l'adresse IP et l'adresse matérielle du nœud déplacé n'ont pas changé (voir Figure 5.11).

Si le nœud est fermé, les autres nœuds ne peuvent plus l'atteindre. Ils utilisent les informations de leur cache ARP pour y accéder, mais la connexion est impossible. Une implémentation pourrait profiter de l'impossibilité d'initier une connexion avec un nœud pour supprimer les informations concernant ce nœud dans la table de cache ARP (voir Figure 5.12). Elle pourrait aussi essayer de nouveau d'accéder au nœud en envoyant quelques requêtes ARP supplémentaires, avant d'abandonner.

Si le nœud a été déplacé jusqu'à un segment de réseau différent (voir Figure 5.13), il se trouve au-delà d'un routeur. Comme les adresses IP contiennent des informations concernant le segment de réseau auquel leur hôte est attaché (voir Chapitre 4), l'adresse IP du nœud déplacé doit changer, et ce même si son adresse matérielle reste la même. Souvenez-vous que les transmissions ARP ne franchissent pas les

routeurs. Du point de vue pratique, on peut considérer que le nouveau nœud est fermé ou non disponible. Cette situation est traitée dans le paragraphe précédent et à la Figure 5.12.

Figure 5.11

Les informations de cache ARP concernant un nœud déplacé dans un même segment de réseau ne changent pas.

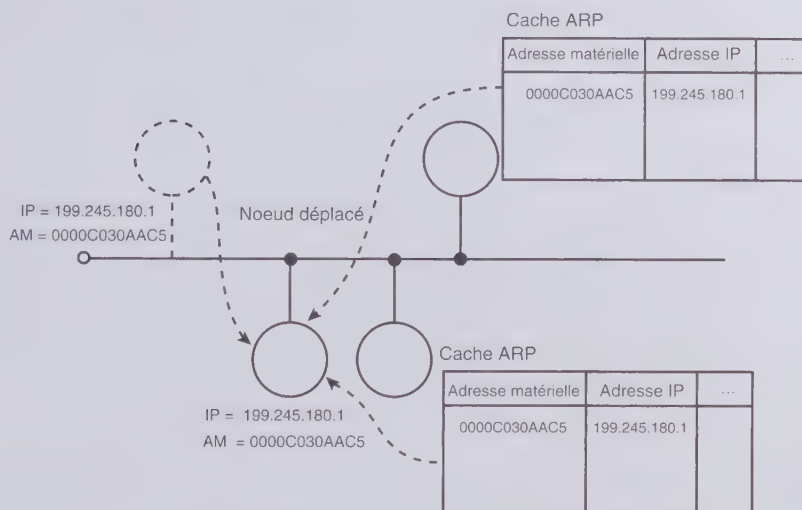
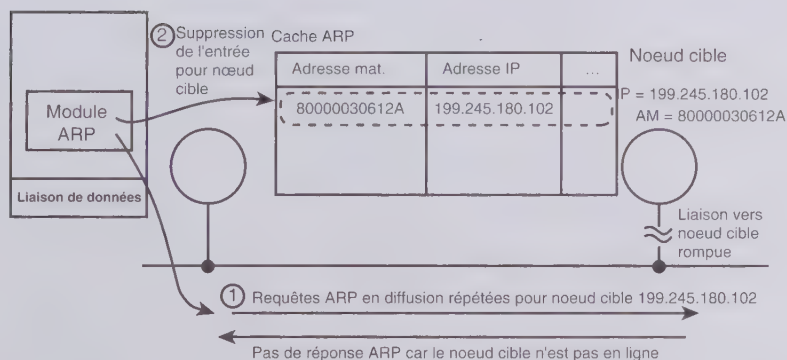


Figure 5.12

La mise à jour du cache ARP lorsqu'un nœud devient inaccessible.

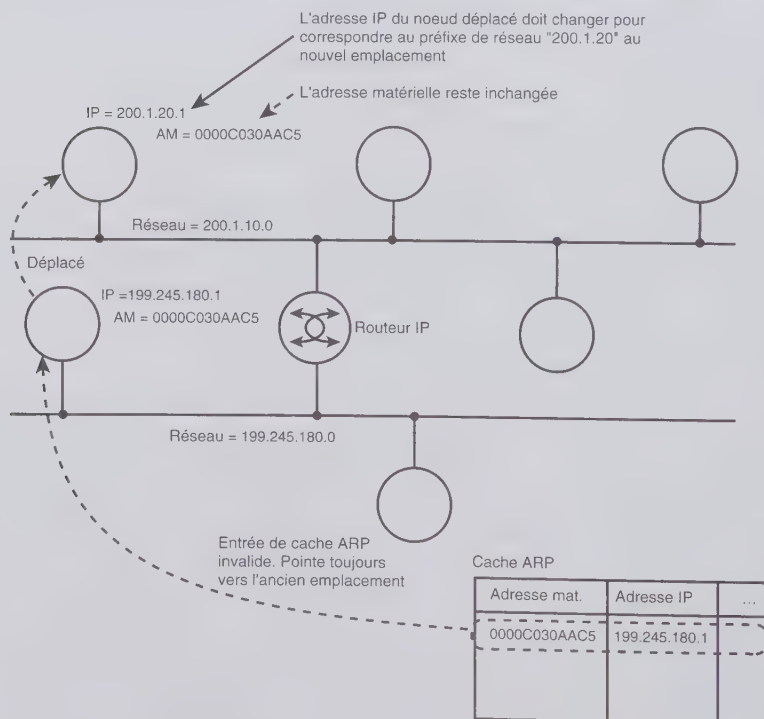


Une implémentation d'ARP met généralement à jour les marques de temps lorsque l'entrée de résolution d'adresse est utilisée pour transmettre des paquets à un nœud. Les marques de temps sont aussi mises à jour lorsque des requêtes ARP sont reçues en provenance d'un nœud possédant une entrée dans la table de cache ARP. Si aucun paquet n'est reçu en provenance d'un nœud au bout d'un certain temps (la valeur de la temporisation ARP), l'entrée de résolution d'adresse est écartée.

Une implémentation ARP peut aussi disposer d'un processus indépendant – aussi appelé *daemon* – pour consulter périodiquement la table de cache ARP et faire expirer les entrées obsolètes. Sous TCP/IP, la temporisation par défaut est de 15 minutes ; certaines implémentations laissent la possibilité à l'administrateur de modifier cette valeur. Pour raffiner, le *daemon* ARP peut aussi commencer par envoyer directement une requête ARP au nœud concerné. Si aucune réponse ARP n'est reçue après plusieurs retransmissions, l'entrée ARP correspondante est écartée. Dans ce cas de figure, la requête ARP est directement envoyée *sans diffusion*, car l'adresse matérielle du nœud est déjà connue grâce à la table de cache ARP.

Figure 5.13

Une table de cache ARP pour un nœud déplacé jusqu'à un segment de réseau différent.



La plupart des implémentations de TCP/IP permettent d'ajouter manuellement des entrées dans la table de cache ARP. En principe, c'est inutile : l'opération ARP dynamique détermine les correspondances entre adresses IP et matérielles. Les entrées ARP ajoutées à la main ne sont pas concernées par la temporisation et permettent de résoudre des problèmes liés à la présence de doublons d'adresses IP ou d'un mauvais fonctionnement du logiciel.

L'outil permettant de visualiser ou de modifier la table ARP dépend du système d'exploitation utilisé. Il peut s'agir d'un outil à ligne de commande ou d'un outil graphique.

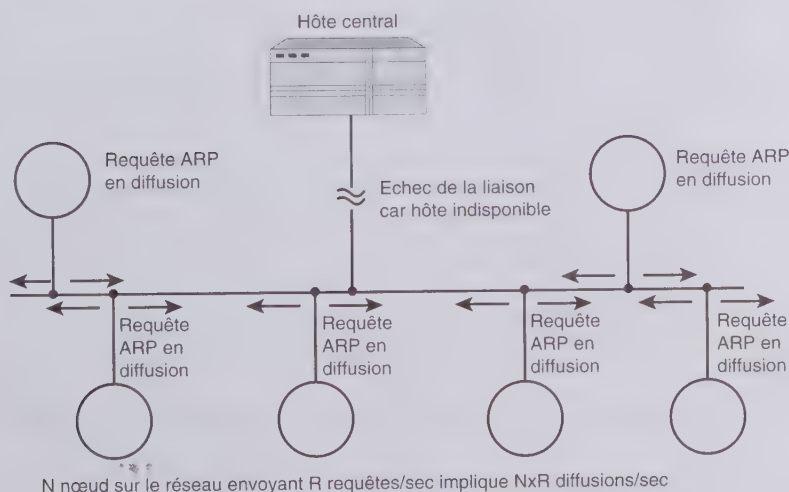
• ARP dans les réseaux à ponts

Lorsqu'une connexion TCP/IP échoue, le protocole TCP, qui est en charge de fournir des échanges fiables, essaye de corriger l'échec en retransmettant les paquets. Si l'échec est une conséquence d'une panne de la liaison physique et que d'autres chemins vers la destination sont disponibles, les datagrammes IP sont routés pour contourner la liaison défectueuse. Si TCP est dans l'impossibilité de corriger l'échec, la correction se poursuit dans les couches les plus basses. Dans ces couches, le module ARP tente de corriger la situation en diffusant une requête ARP.

Si le nœud cible se trouve sur le segment du réseau local, cette technique est bien adaptée pour corriger les échecs. Toutefois, les choses se compliquent lorsque plusieurs nœuds commencent à envoyer des diffusions de requêtes ARP. Ce cas de figure peut survenir lorsqu'un hôte central n'est plus disponible suite à une panne d'un hôte ou d'une liaison (voir Figure 5.14).

Figure 5.14

Le trafic de diffusion ARP après une panne.



Si cinq cents stations sont connectées à un hôte central sur le réseau, et que la connexion à cet hôte tombe en panne, les cinq cents stations commencent à transmettre simultanément des diffusions ARP. Sur certains systèmes, de nouvelles tentatives d'envoi de requêtes ARP sont effectuées toutes les secondes. Ce trafic de diffusion augmente considérablement la charge du réseau qui risque d'affecter la disponibilité du réseau pour d'autres nœuds du réseau.

Le trafic de diffusion peut même ralentir des nœuds qui n'accèdent pas présentement au réseau. En effet, la carte réseau et le pilote logiciel associé doivent examiner chaque paquet envoyé en diffusion. Dans le cas des requêtes ARP, le module ARP est aussi exécuté pour chaque requête en diffusion. Le traitement d'une cinquantaine de requêtes ARP par seconde suffit à occuper une bonne partie de la puissance de capture ayant pour conséquence un ralentissement notable sur le nœud. En conséquence, tous les nœuds du réseau semblent ralentir.

La répétition de diffusions ARP pose des problèmes supplémentaires dans le cas d'un réseau à ponts (voir Figure 5.15). Les segments de réseaux séparés par des ponts sont considérés comme faisant partie du même domaine MAC. Les requêtes/réponses ARP doivent être transmises par le biais de ces ponts. La propagation du trafic des diffusions par-delà les ponts peut faire boule de neige et ralentir encore plus le trafic du réseau. La mise en place d'une topologie de réseau étendue ou trop complexe, utilisant des ponts n'est donc pas désirable dans ces cas de figure.

• Doublons d'adresses et ARP

Les adresses IP doivent être distinctes sur un réseau. Comment s'assurer qu'elles le sont ? En documentant et en affectant correctement les adresses IP. Du fait de l'implication du facteur humain dans l'affectation des adresses IP, il arrive que des nœuds d'un réseau reçoivent la même adresse IP. C'est alors une porte ouverte au chaos et à la confusion. Les applications en réseau fonctionneront par intermittence ou cesseront mystérieusement toute activité.

Pour mieux comprendre l'effet de doublons d'adresses IP, les sections qui suivent détaillent deux scénarios :

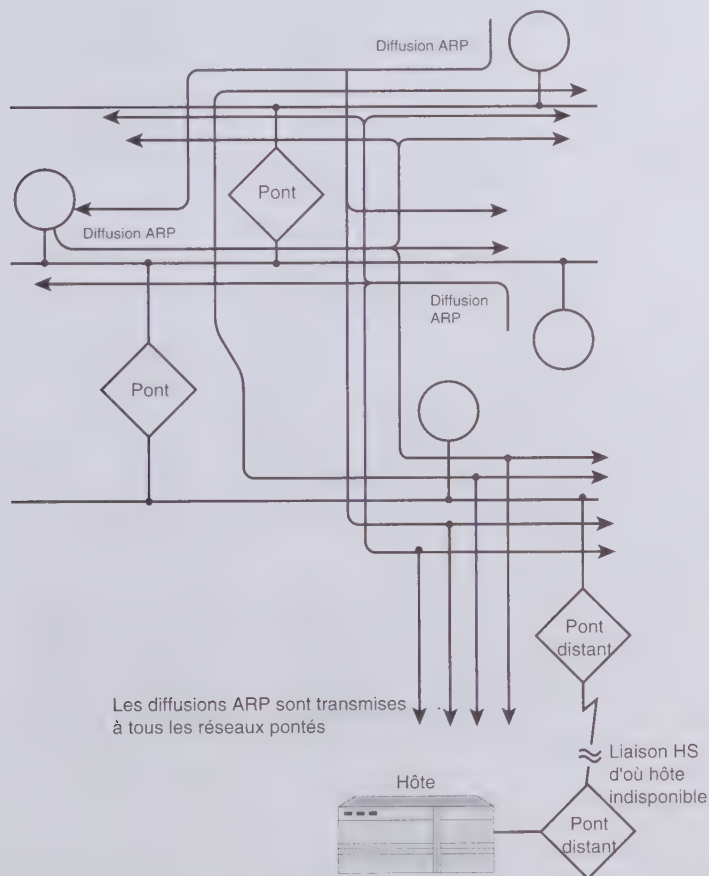
- des doublons d'adresses au niveau des clients TCP/IP ;
- des doublons d'adresses au niveau des serveurs TCP/IP.

Doublons d'adresses IP au niveau des clients TCP/IP

Considérez la situation présentée à la Figure 5.16. On y voit deux clients TCP/IP, appelés "station de travail 1" et "station de travail 2", qui accèdent à un serveur TCP/IP central avec l'adresse 144.19.74.102. L'adresse matérielle du serveur est 080020021545.

Figure 5.15

Le trafic ARP sur un réseau à ponts.



Les stations de travail 1 et 2 utilisent la même adresse IP 144.19.74.1. Leurs adresses matérielles sont respectivement 0000C0085121 et 0000C0075106.

La station de travail 1 commence une session FTP avec le serveur TCP/IP. Avant le début de la session FTP, la station de travail 1 envoie une requête ARP en diffusion pour déterminer l'adresse matérielle du serveur TCP/IP. A la réception de la requête ARP, le serveur TCP/IP ajoute une entrée pour l'émetteur de la requête (la station de travail 1) dans sa table de cache ARP, et envoie une réponse ARP à l'émetteur. A la réception de la réponse, la station de travail 1 ajoute une entrée

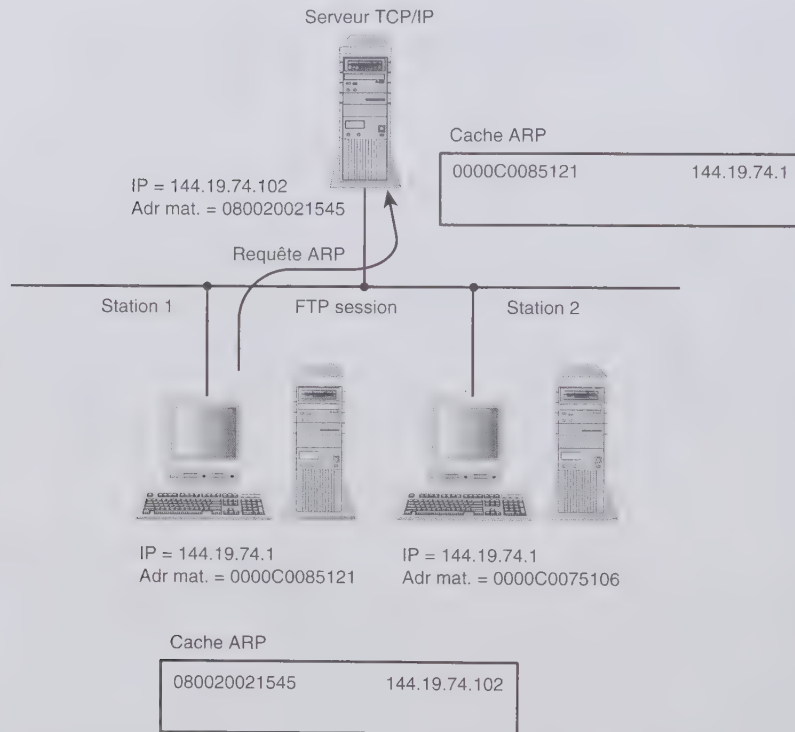


pour le serveur TCP/IP dans sa table de cache ARP locale. A cet instant, la table de cache ARP du serveur TCP/IP est dans l'état suivant :

Adresse IP	144.19.74.1
Adresse matérielle	000C0085121

Figure 5.16

Doublons d'adresses IP au niveau des clients TCP/IP, première phase (source : Learning Tree).



• **Etape 1**

- La station ouvre la session FTP auprès du serveur

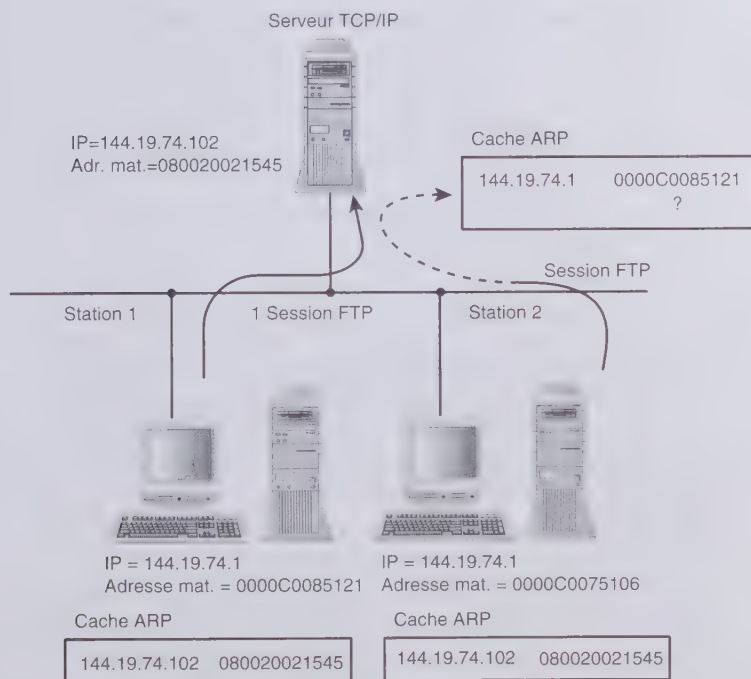
La table de cache ARP de la station de travail 1 est dans l'état suivant :

Adresse IP	144.19.74.102
Adresse matérielle	080020021545

Dans un deuxième temps, la station de travail 2 tente d'établir une connexion TCP/IP (une autre session FTP par exemple) avec le même serveur TCP/IP (voir Figure 5.17)

Figure 5.17

Doublons d'adresses IP au niveau des clients TCP/IP, seconde phase (source : Learning Tree).



• **Etape 2**

- La 2ème station avec une adresse IP en doublon ouvre la session FTP auprès du serveur.

Avant de procéder à la connexion, la station de travail 2 doit découvrir l'adresse matérielle du serveur TCP/IP. Il envoie à cet effet une requête ARP. Après réception de la requête ARP, le serveur TCP/IP souhaite ajouter une entrée pour l'émetteur de requête ARP (la station de travail 2) dans sa table de cache ARP, et envoie une réponse ARP à l'émetteur. Le serveur TCP/IP découvre qu'il existe déjà une entrée pour 144.19.74.1 dans sa table de cache ARP. Conformément à la RFC 826, le serveur TCP/IP doit remplacer l'entrée existante par la nouvelle.

La plupart des implémentations de TCP/IP effectuent cette substitution silencieusement, sans avertir de la possibilité d'un problème de doublons d'adresses IP. Certaines implémentations le font toutefois. Le serveur TCP/IP répond à la requête ARP provenant de la station de travail 2. A la réception de la réponse ARP, la station de travail 2 ajoute une entrée pour le serveur TCP/IP dans sa propre



table de cache ARP. A ce moment, la table de cache pour le serveur TCP/IP est la suivante :

Adresse IP	144.19.74.1
Adresse matérielle	0000C0075106

La table de cache ARP pour la station de travail 2 est la suivante :

Adresse IP	144.19.74.102
Adresse matérielle	080020021545

Lorsque le serveur TCP/IP souhaite envoyer des données à la station de travail 1, il recherche l'adresse matérielle de la station de travail 1 correspondant à l'adresse IP 144.19.74.1 dans sa table de cache ARP, et envoie les données à cette adresse matérielle. Malheureusement, il s'agit en fait de l'adresse matérielle de la station de travail 2, vers laquelle sont donc envoyées les données. La station de travail 1 ne recevant pas les données attendues tente de renvoyer sa dernière transmission, et le serveur TCP/IP continue d'envoyer la réponse à la station 2. La station 1 se retrouve en conséquence bloquée en attente d'une réponse. Elle finit par renoncer après expiration d'un temporisateur où il deviendra nécessaire de redémarrer la station pour la rétablir. Dans tous les cas de figure, une application qui jusqu'ici fonctionnait correctement cessera immédiatement de le faire, à la plus grande consternation de l'utilisateur et de l'administrateur système.

Lorsque la station 2 reçoit des données destinées en fait à la station 1, elle les rejette sans doute, génère éventuellement un message d'erreur et poursuit sa session avec le serveur TCP/IP. Il se peut aussi que, déroutée par cet afflux de données non souhaitées, la station 2 se bloque elle aussi.

Si l'utilisateur de la station 1 redémarre son poste et tente de se reconnecter au serveur TCP/IP, l'entrée de cache ARP du serveur TCP/IP pour la station 2 sera remplacée par celle correspondant en fait à la station 1. En conséquence, la station 2 se bloquera et la situation risque de se perpétuer jusqu'à ce que l'utilisateur de désespoir abandonne.

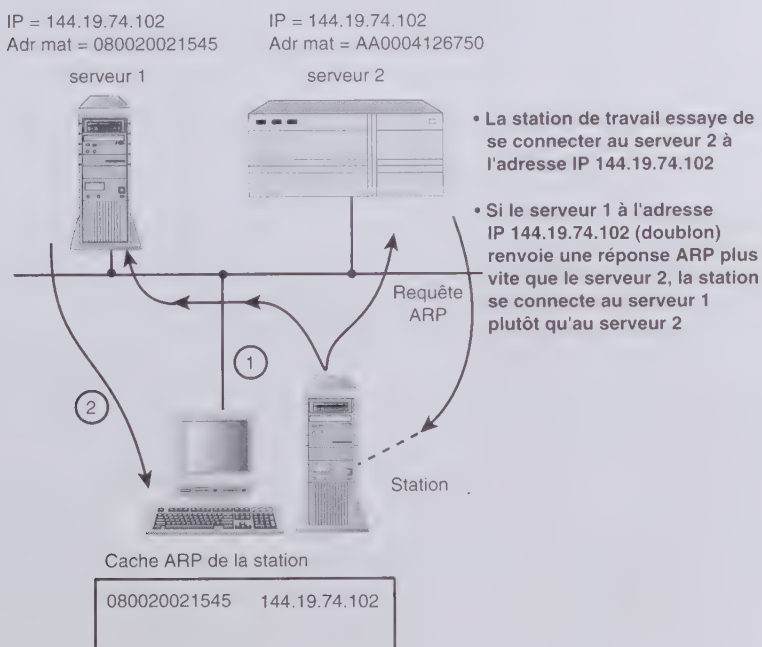
Si les utilisateurs des stations 1 et 2 ne communiquent pas, ils continueront à constater des blocages inattendus lors de l'accès au serveur TCP/IP. Si les utilisateurs n'accèdent pas au serveur en même temps, le problème évoqué ici ne se posera jamais et le doublon d'adresse IP ne sera pas détecté. S'il arrivait aux utilisateurs d'être confrontés de temps à temps à ce problème, ils le mettraient sur le compte des sautes d'humeur passagères du serveur et la cause resterait là encore inconnue.

Doublons d'adresses IP au niveau des serveurs TCP/IP

Prenons l'exemple de la situation décrite à la Figure 5.18, montrant deux serveurs TCP/IP appelés "serveur 1" et "serveur 2". Ils ont tous deux la même adresse IP, 144.19.74.102. Les adresses matérielles des serveurs sont 080020021545 et AA0004126750 respectivement.

Figure 5.18

Doublons d'adresses IP au niveau des serveurs (source : Learning Tree).



Une station de travail dont l'adresse IP est 144.19.74.1 tente d'accéder au serveur TCP/IP à 144.19.74.102. L'adresse matérielle de la station de travail est 0000C0085121.

La station tente de se connecter au serveur 2. Avant d'effectuer la connexion TCP/IP, elle envoie une requête ARP en diffusion pour prendre connaissance de l'adresse matérielle du serveur TCP/IP. Deux serveurs TCP/IP ont la même adresse IP cible 144.19.74.102. A la réception de la requête ARP, les serveurs 1 et 2 ajoutent chacun dans leur table de cache ARP une entrée pour l'émetteur de la requête (la station) et envoient une réponse ARP à l'émetteur. Deux réponses sont



donc envoyées sur le réseau. La station de travail accepte la première réponse ARP et ignore silencieusement (c'est-à-dire sans notifier l'anomalie) la seconde. Si la réponse ARP du serveur 1 atteint la station en premier, la station ajoute une entrée pour le serveur 2 dans sa table de cache ARP locale. A ce moment, les tables de cache ARP des serveurs et de la station sont les suivantes :

Adresse IP du serveur 1	144.19.74.1
Adresse matérielle	0000C0085121
Adresse IP du serveur 2	144.19.74.1
Adresse matérielle	0000C0085121
Adresse IP de la station	144.19.74.102
Adresse matérielle	080020021545

La station continue alors d'interagir avec le serveur 1 au lieu du serveur 2, comme c'était son intention à l'origine. Si le serveur 1 ne propose pas le service attendu, la connexion est un échec et l'utilisateur reçoit un message d'erreur indiquant que le service n'est pas pris en charge sur le serveur. L'utilisateur en reste perplexe, à plus forte raison s'il avait l'habitude d'accéder sans problème aux services du serveur 2.

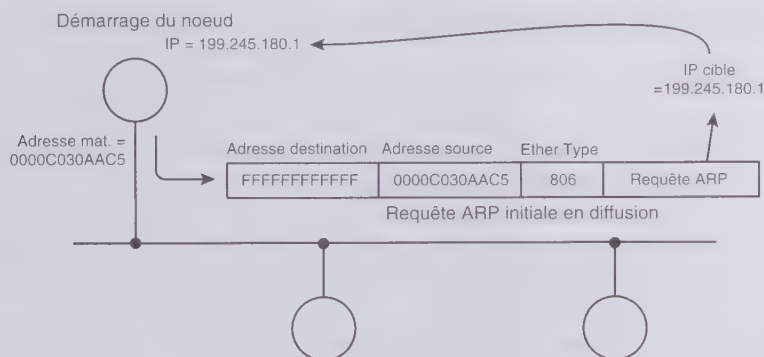
Si le serveur 1 dispose du service attendu par la station, l'utilisateur tente de se loguer à ce service. Si le service est Telnet ou FTP, un nom d'utilisateur et un mot de passe sont demandés. Si l'utilisateur n'est pas vigilant, il peut ne pas remarquer que le nom/version du serveur FTP ou Telnet apparaissant alors à l'écran ne correspond pas à celui attendu. Si l'utilisateur ne dispose pas d'un compte nom/mot de passe similaire sur le serveur 1, l'accès lui est refusé. Si le nom/mot de passe est similaire sur les deux serveurs, l'utilisateur se logue sur le mauvais serveur et risque de ne pas trouver les fichiers ou les données attendus.

Si le serveur est un hôte multiconnecté (c'est-à-dire un serveur disposant de plusieurs connexions réseau) tenant lieu de routeur, les choses peuvent encore empirer puisque les paquets destinés à des réseaux extérieurs risquent de ne pas suivre la route escomptée ou de n'être tout simplement pas transmis parce qu'un serveur a, à tort, accepté les paquets.

Le test ARP de doublons d'adresses

De nombreuses implémentations de TCP/IP envoient une requête ARP en diffusion lors du démarrage. La requête ARP est envoyée avec un champ IP cible contenant l'adresse du nœud qui démarre (voir Figure 15.19).

Figure 5.19
La requête ARP initiale
lors du démarrage.



La finalité de cette requête ARP initiale est de découvrir s'il existe des nœuds de même adresse IP. Si une réponse ARP est reçue, cela signifie qu'un autre nœud (celui qui a généré la réponse) a la même adresse IP que le nœud qui vient de démarrer. Le nœud qui démarre doit alors notifier la présence de ce doublon d'une manière appropriée. Le nœud répondant à la requête ARP peut même effectuer une vérification supplémentaire sur le champ IP émetteur du paquet de la requête ARP pour le comparer avec sa propre adresse IP. Si les champs IP émetteur et IP cible sont les mêmes, le nœud cible peut générer un message d'alerte approprié avertissant du problème rencontré.

La trame ARP initiale ne peut découvrir des problèmes de doublons d'adresses IP que sur son propre segment de réseau. Comme les paquets ARP ne franchissent pas les routeurs, cette technique ne permet pas de détecter des doublons d'adresses sur un réseau comprenant plusieurs segments connectés par des routeurs.

Une autre fonction de la diffusion ARP de démarrage est de faire que les nœuds possédant une entrée pour le nœud qui démarre mettent à jour leur table de cache ARP. Si le nœud qui démarre a une adresse matérielle différente (parce que l'on a changé sa carte réseau), tous les nœuds recevant la diffusion ARP écartent l'ancienne adresse matérielle et mettent à jour leur table avec la nouvelle adresse.



Trace de protocole pour ARP

Un "analyseur de protocole" est un périphérique permettant d'intercepter le trafic d'un réseau. Le trafic ainsi intercepté est appelé une "trace de protocole". Le pilote de réseau de l'analyseur de protocole place la connexion de réseau dans un mode d'opération "libéral" (*promiscuous*). Dans ce mode, la carte réseau accepte tous les paquets transitant par le segment de réseau, y compris les paquets à destination d'autres nœuds. L'analyseur de protocole intercepte le trafic du réseau dans des tampons (*buffers*) internes et les décode à fin d'analyses. L'utilisation de l'analyseur de protocole sur un réseau pose un problème de sécurité, en particulier si des données sensibles telles que des noms d'utilisateur et des mots de passe, sont envoyées en clair (sans chiffrement).

L'analyseur de protocole peut s'avérer très utile pour comprendre les protocoles de réseau et en corriger les dysfonctionnements. Nous allons voir des traces de protocoles provenant d'un vrai réseau. Nous analyserons ces traces pour mieux comprendre les mécanismes internes qui sont en œuvre.

La Figure 5.20 montre une séquence typique de requête/réponse ARP, interceptée à l'aide d'un analyseur de protocole. La trace de protocole est analysée plus en détails dans les Figures 5.21, 5.22 et 5.23.

Figure 5.20

Résumé de l'échange de paquets ARP.

No.	Source	Destination	Couche	Taille	Résumé
1	0000C024282D	FFFFFFFFFFFF	arp	0064	Demandé par 144.19.74.44 pour 144.19.74.44
2	0000C024282D	FFFFFFFFFFFF	arp	0064	Demandé par 144.19.74.44 pour 144.19.74.201
2	0000C0A20F8E	0000C024282D	arp	0064	Réponse 144.19.74.201 = 0000C0A20F8E

No. de paquet

Adresse MAC source

Adresse MAC destination

Paquet ARP vu par le champ EtherType

Taille de trame Ethernet en octets

Résumé du décodage du paquet ARP

Le résumé de la trace des paquets ARP tel qu'il apparaît à la Figure 5.20 montre trois paquets ARP. Le paquet 1 est une requête ARP initiale envoyée lors du démarrage d'une station d'adresse IP 144.19.74.44. Le paquet 2 est la requête ARP pour une adresse IP cible spécifiée, et le paquet 3 est la réponse ARP correspondante.

• Diffusion ARP initiale

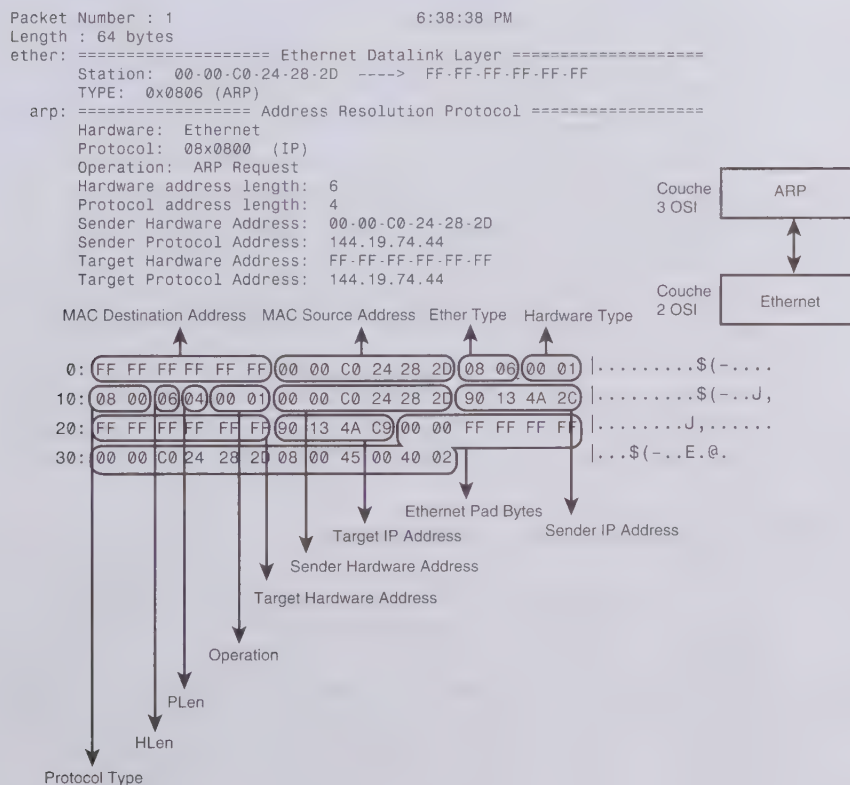
Lorsque la station d'adresse IP 144.19.74.44 a démarré, elle a envoyé une requête ARP à sa propre adresse IP, et cela pour deux raisons :

1. Pour détecter un éventuel doublon d'adresse sur le segment du réseau ;
2. Pour mettre à jour les tables de cache ARP de nœuds possédant déjà des informations concernant 144.19.74.44.

La Figure 5.21 décrit en détail le paquet de diffusion initial.

Figure 5.21

Broadcast ARP initial.



D'après l'examen de la couche de liaison utilisée pour la transmission de la requête ARP, vous pouvez constater qu'il s'agit d'une trame Ethernet dont les champs ont les valeurs suivantes (en hexadécimal) :

Adresse de destination	= FFFFFFFF
Adresse source	= 0000C024282D
EtherType	= 0806

L'adresse de destination Ethernet FFFFFFFF correspond à tous les bits à 1 et constitue donc une adresse de diffusion. La station qui envoie la diffusion a pour adresse Ethernet 0000C024282D. Le champ EtherType vaut 806 (hexadécimal), qui est la valeur EtherType réservée pour les paquets ARP. Cela signifie que le paquet ARP est encapsulé dans une trame Ethernet.

Le champ Opération du paquet ARP a pour valeur 1, ce qui indique qu'il s'agit d'une requête ARP. Les valeurs des champs AM émetteur et IP cible ont pour valeur FFFFFFFF et 144.19.74.44 respectivement. Le champ AM cible n'est pas connu, et la plupart des implémentations mettent tous les bits du champ à 0 ou à 1. Ici, tout est à 1.

L'adresse IP cible 144.19.74.44 est la même que celle de l'émetteur. On en déduit que ce paquet de requête ARP est une diffusion ARP initiale envoyée lors du démarrage de la pile de protocoles TCP/IP. Ce paquet permet de détecter l'existence de doublons d'adresses IP sur le segment de réseau et de provoquer la mise à jour dans les tables de cache ARP d'autres nœuds des informations existantes concernant la station qui démarre.

• Bourrage des trames Ethernet

Dans la représentation hexadécimale du paquet ARP (voir Figure 5.21), plusieurs octets sont marqués de la mention *Ethernet Pad Bytes*. La taille des paquets Ethernet est au minimum de 64 octets (voir Chapitre 2). Si la taille du paquet à envoyer est inférieure à cette taille minimale, Ethernet remplit les octets manquants pour obtenir une taille de trame de 64 octets. La plupart des pilotes Ethernet remplissent avec n'importe quelles données se trouvant dans le buffer de transmission. Vous pouvez le constater dans les décodages de paquets (voir Figures 5.21, 5.22 et 5.23).

Si vous comptez le nombre d'octets apparaissant dans le dump hexadécimal, vous n'en trouvez en réalité que 60, alors même qu'une longueur de 64 octets est indiquée au début de ces décodages. Qu'est-il arrivé aux 4 octets manquants ? Ils constituent en fait le total de contrôle (*checksum*) CRC utilisé par Ethernet. Bien que le total de contrôle CRC fasse partie de la trame Ethernet, la plupart des analyseurs de protocoles n'en donnent pas la valeur.

Les champs d'en-tête Ethernet sont les mêmes que ceux de la Figure 5.21. En effet, la trame a été envoyée par la même station, 144.19.74.44. Vous pouvez remarquer là aussi que le champ EtherType vaut 806 (hexadécimal), qui est la valeur EtherType réservée pour les paquets ARP. Cela signifie qu'un paquet ARP est encapsulé dans une trame Ethernet.

Le champ Opération du paquet ARP vaut 1, il s'agit donc d'une requête ARP. Les valeurs des champs AM émetteur et IP cible ont pour valeur 0000C024282D et 144.19.74.44 respectivement. Les champs AM cible et IP cible ont pour valeur FFFFFFFF et 144.19.74.201 respectivement. Le champ AM cible n'est pas connu et la plupart des implémentations mettent tous les bits du champ à 0 ou à 1. Ici, tout est à 1.

L'adresse IP cible 144.19.74.201 est celle du nœud cible dont on cherche l'adresse matérielle.

• Paquet de réponses ARP

La Figure 5.23 détaille le paquet de réponse ARP qui a été envoyé par le nœud cible en 144.19.74.201, en réponse au paquet de requête ARP de la Figure 5.22.

Ethernet est la couche de liaison de données utilisée pour la transmission de la requête ARP. Le décodage du paquet montre une trame Ethernet dont les champs ont les valeurs suivantes (en hexadécimal) :

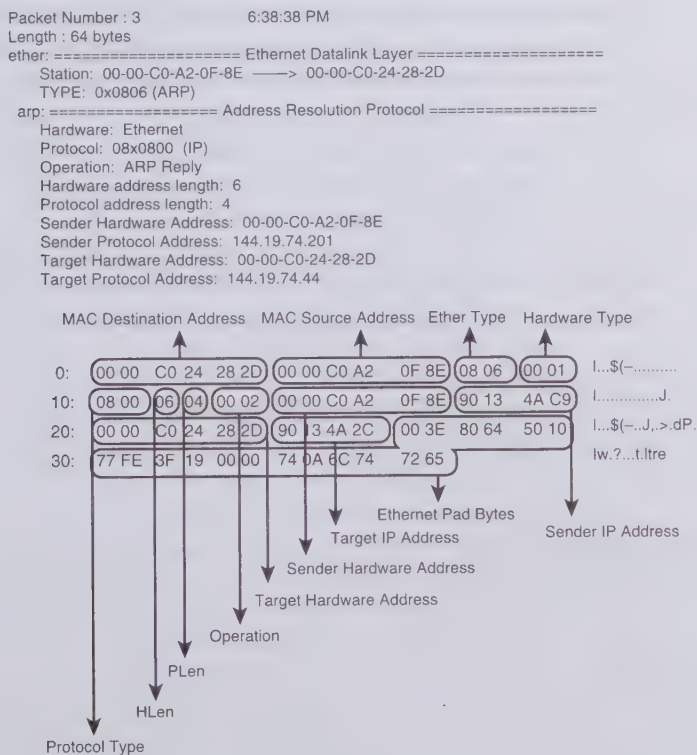
Adresse de destination	= 0000C024282D
Adresse source	= 000C0A20F8E
EtherType	= 806

Les champs Ethernet indiquent que cette trame Ethernet est envoyée au nœud d'adresse MAC 0000C024282D, correspondant à l'émetteur de la requête ARP. Vous pouvez remarquer que la réponse ARP n'est pas envoyée en diffusion, mais directement à l'émetteur de la requête ARP. Le champ EtherType vaut 806 (hexadécimal), ce qui confirme qu'il s'agit là d'un paquet ARP encapsulé dans une trame Ethernet.

Le champ Opération du paquet ARP a pour valeur 2, ce qui signifie qu'on a là une requête ARP. Les champs AM émetteur et IP émetteur ont pour valeur 0000C0A20F8E et 144.19.74.201 respectivement. Cette valeur du champ AM émetteur est la réponse qui a motivé la requête ARP. Il s'agit de l'adresse matérielle de la cible 144.19.74.201. Les champs AM cible et IP cible ont pour valeur 0000C024282D et 144.19.74.44 respectivement. Le champ AM cible est l'adresse matérielle de l'émetteur de la requête ARP, et le champ IP cible son adresse IP.

Figure 5.23

Paquet de réponse ARP.



Résolution inverse d'adresses

Dans les exemples d'utilisation d'ARP que nous avons pu voir, on supposait qu'un nœud connaissait sa propre adresse IP. Mais comment fait-il pour la connaître ? Il peut l'obtenir dans son stockage local, généralement à l'intérieur d'un fichier dont le nom et l'emplacement diffèrent selon l'implémentation de TCP/IP et le système d'exploitation. Cependant, une catégorie d'ordinateurs ne dispose pas de stockage local. Toutes les informations système de ces ordinateurs se trouvent dans un serveur distant. Ces ordinateurs sont appelés des "stations sans disque" (*diskless stations*).

Les stations sans disque modernes peuvent néanmoins disposer d'un disque local. Mais ce disque sert à accélérer le fonctionnement du système d'exploitation, et



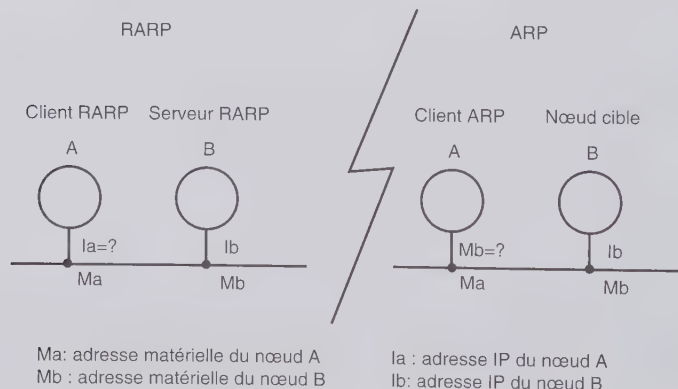
non à stocker des paramètres concernant TCP/IP. Ces paramètres se trouvent sur un système distant. Par exemple, le disque local peut uniquement servir pour contenir une zone d'échange locale implémentant le mécanisme de mémoire virtuelle du système d'exploitation.

Les stations sans disque disposent aussi sur des serveurs distants d'une copie de l'image du système d'exploitation. Au démarrage, la station sans disque télécharge une copie de cette image, du serveur distant à sa mémoire. Pour effectuer ce téléchargement, la station a besoin d'avoir une adresse IP. Cette adresse IP ne peut être attribuée aléatoirement. Elle doit être distincte et posséder le même préfixe que les adresses IP des autres nœuds du même segment de réseau. Si les autres stations du réseau ont des adresses IP de type 199.245.180.1, 199.245.180.5 et 199.245.180.7, le préfixe commun au segment est donc "199.245.180". L'adresse IP de la station sans disque devra commencer par ce préfixe. Ce préfixe commun est nécessaire, car les informations sur le routage dans un segment de réseau sont incorporées sous la forme du préfixe même.

La station sans disque obtient généralement son adresse IP en envoyant une requête aux serveurs se trouvant sur son segment de réseau. Comme elle ne connaît pas les informations concernant le serveur distant, la requête est envoyée sous forme de diffusion de couche liaison de données. Un des mécanismes permettant d'obtenir son adresse IP auprès d'un serveur distant est RARP (*Reverse Address Resolution Protocol*, "Protocole de résolution inverse d'adresses"). Le terme "inverse" figure ici, car les informations glanées par ce protocole sont inverses de celles obtenues par ARP. La Figure 5.24 montre le type d'informations recherchées par le client RARP comparées aux informations recherchées par un client ARP. Le client RARP, A, connaît son adresse matérielle (Ma), mais pas son adresse IP (Ia). Dans le cas d'ARP, le client ARP connaît son adresse matérielle et son adresse IP, mais ne connaît pas l'adresse matérielle de B (Mb).

Figure 5.24

Comparaison de RARP et ARP.



Les stations sans disque permettent de diminuer le coût du matériel, de simplifier la configuration et la mise à niveau des nœuds grâce à la conservation des informations critiques en un lieu centralisé ; elles réduisent les risques d'infection par des virus parce que l'utilisateur ne peut facilement introduire des programmes directement dans une telle station.

Les sections qui suivent détaillent le protocole RARP.

• Fonctionnement de RARP

Dans le cas de RARP, le nœud qui veut connaître sa propre adresse IP (le client RARP) envoie en diffusion une requête appelée requête RARP. La "requête RARP" est envoyée en diffusion au niveau de la couche de liaison de données, car le client RARP ne connaît pas l'adresse (matérielle ou IP) du serveur distant. Le serveur distant qui traite la requête RARP est appelé le serveur RARP. Tous les nœuds sur le segment de réseau reçoivent la requête RARP en diffusion, mais seuls les nœuds tenant lieu de serveurs RARP y répondent.

S'il existe plusieurs serveurs RARP, chacun d'eux tente de traiter la requête RARP. Généralement, le client RARP accepte la première réponse reçue et ignore silencieusement les suivantes.

Le serveur RARP tient à jour une table des adresses IP des nœuds du segment. Cette table est indexée par un identificateur distinct, spécifique à chaque machine. Cet identificateur distinct est envoyé dans la requête RARP. Pour les stations sans disque, cet identificateur doit être un paramètre lié au matériel et aisément lisible par la station. L'identificateur ne peut être une valeur stockée dans le stockage local, car l'éventuel stockage local d'une station sans disque n'a pas vocation de contenir ce genre d'informations. Les concepteurs de RARP (voir STD 38, RFC 903) ont décidé d'utiliser les adresses matérielles comme identificateurs distincts, car ces adresses sont bien sur un segment de réseau et peuvent aisément être lues par les pilotes de réseau.

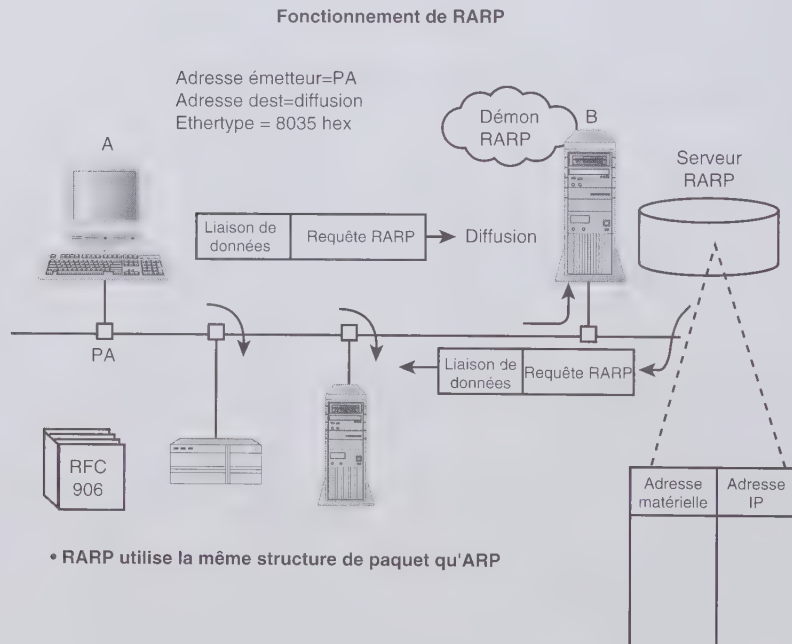
Lorsqu'un serveur RARP reçoit une requête RARP, il consulte sa table de correspondances entre adresses IP et adresses matérielles (voir Figure 5.25). S'il trouve une entrée dans la table RARP correspondant à l'adresse matérielle figurant dans la requête RARP, il renvoie l'adresse IP correspondante dans une réponse RARP. La réponse RARP n'est pas envoyée en diffusion, car le serveur RARP connaît l'adresse matérielle du client RARP grâce à sa table.



Les paquets de requête et de réponse RARP utilisent le même format que les paquet ARP (voir Figure 5.3). La différence entre ARP et RARP tient dans le placement des valeurs dans les champs. Lorsque des paquets RARP sont dans une trame Ethernet, on utilise une valeur de 8035 (en hexadécimal) pour l'EtherType.

Figure 5.25

Fonctionnement de RARP
(source : Learning Tree).



Les champs de la requête RARP sont remplis comme suit (les références aux noms de champs renvoient à ceux du paquet ARP de la Figure 5.3) :

• **Requête RARP :**

- adresse matérielle de destination pour la couche de liaison de données = Diffusion ;
- adresse matérielle source pour la couche de liaison de données = AM client RARP ;
- EtherType = 8035 (hexadécimal) ;
- opération = 3 (requête RARP) ;
- AM émetteur = AM client RARP ;
- IP émetteur = non défini. Généralement on utilise 0.0.0.0 ;
- AM cible = AM client RARP ;
- IP cible = non défini.

• Réponse RARP :

- adresse matérielle de destination pour la couche de liaison de données = AM client RARP ;
- adresse matérielle source pour la couche de liaison de données = AM serveur RARP ;
- EtherType = 8035 (hexadécimal) ;
- opération = 4 (réponse RARP) ;
- AM émetteur = AM serveur RARP ;
- IP émetteur = adresse IP serveur RARP ;
- AM cible = AM client RARP ;
- IP cible = adresse IP du client RARP (la réponse recherchée).

Le champ AM cible dans la requête RARP correspond à l'adresse matérielle du client RARP, afin de faciliter la tâche du serveur RARP qui n'a plus besoin de modifier ce champ, puisque ce dernier contient l'adresse matérielle client dans la requête RARP.

Le pilote de réseau examine le champ EtherType et détermine qu'il s'agit d'un paquet RARP (EtherType=8035 hexadécimal). Il envoie la requête au module RARP. Les codes du champ Opération sont 3 pour une requête RARP et 4 pour une réponse RARP. Ces champs peuvent aussi permettre de faire la distinction entre un paquet ARP et un paquet RARP. Mais cette manière de faire n'est pas très efficace, car le module ARP devrait alors commencer par examiner le champ Opération et, dans le cas où il trouverait indiqué un paquet ARP, il devrait l'envoyer au module RARP. En utilisant le champ EtherType, le démultiplexage (voir Chapitre 2) peut être effectué efficacement à une couche plus basse.

Dans la plupart des implémentations, RARP n'est pas automatiquement fourni par le module ARP ou IP. Il doit être exécuté sous forme de processus séparé, un daemon par exemple, sur l'ordinateur qui tiendra lieu de serveur RARP.

• Tempêtes RARP

Si le serveur RARP est indisponible à cause d'une panne ou d'un dysfonctionnement de la liaison, les clients RARP sont dans l'impossibilité de démarrer. Les clients envoient continuellement des requêtes RARP en diffusion. Si de nombreux clients RARP envoient simultanément leurs requêtes, la charge du trafic réseau est considérablement alourdie. On appelle ce cas de figure une "tempête RARP".



La non-réception d'une réponse RARP a des conséquences plus graves que la non-réception d'une réponse ARP. L'absence d'une réponse ARP implique qu'un hôte particulier et ses services sont indisponibles. Le client ARP peut toujours essayer d'accéder à d'autres nœuds du réseau. En revanche, en l'absence d'une réponse RARP, le client RARP ne peut démarrer. La plupart des clients RARP continuent indéfiniment à envoyer des requêtes RARP dans l'espoir que le serveur RARP redevienne disponible. Cela a pour conséquence une charge accrue pour le réseau.

• Serveurs RARP principaux et auxiliaires

Pour mieux garantir la disponibilité des services RARP, on peut utiliser plusieurs serveurs RARP. Dans ce cas, tous les serveurs tentent de répondre simultanément à la requête RARP envoyée en diffusion. Une seule des réponses RARP est utilisée par le client RARP. Toutes les autres réponses sont écartées et consomment de la bande passante. Une technique permettant d'empêcher la multiplication des réponses RARP consiste à désigner un des serveurs comme serveur principal et les autres comme serveurs auxiliaires.

Lors de la réception d'une requête RARP, le serveur RARP principal répond à la requête. Les serveurs auxiliaires ne répondent pas, mais enregistrent l'heure d'arrivée de la requête. Si le serveur principal ne répond pas dans un intervalle de temps donné, le serveur RARP auxiliaire considère que le serveur principal est en panne et répond à la requête RARP.

S'il existe plusieurs serveurs RARP auxiliaires, tous les serveurs auxiliaires tentent de répondre en même temps. On peut raffiner la technique en faisant que les serveurs RARP secondaires attendent un intervalle de temps aléatoire avant de répondre. Si le serveur principal fonctionne normalement, l'envoi de la réponse RARP n'est pas retardé. S'il est en panne, un des serveurs RARP répondra passé un court intervalle de temps.

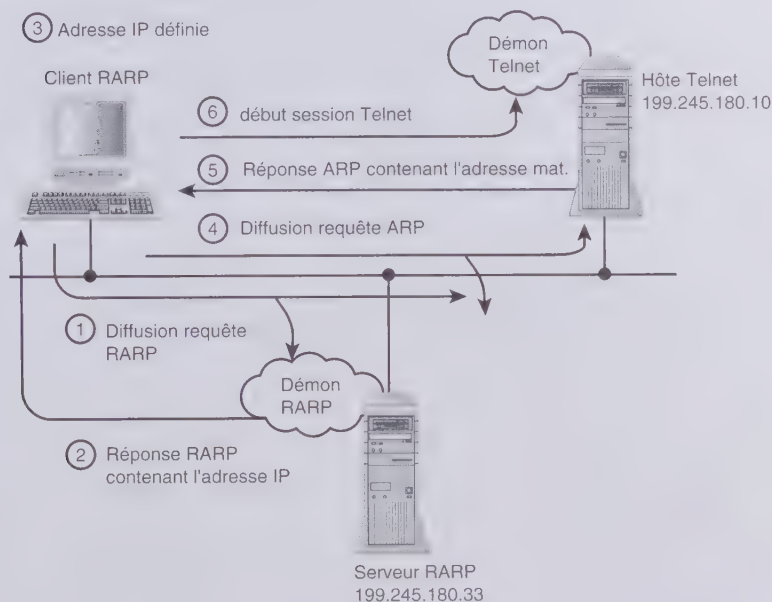
Trace de paquets RARP

Cette section analyse la trace de paquets produite par un client RARP envoyant une requête RARP pour s'initialiser lui-même. Le client RARP tente de démarrer et établit une session Telnet avec un hôte à l'adresse IP 199.245.180.10. Un serveur RARP a été configuré à l'adresse IP 199.245.180.33. Cette situation est illustrée à la Figure 5.26, où un client RARP envoie une requête en diffusion pour obtenir son adresse IP. Lorsque le serveur RARP répond avec l'adresse IP du client RARP, ce dernier tente d'établir une session Telnet avec l'hôte d'adresse IP 199.245.180.10. Avant de pouvoir établir la session, le client Telnet doit obtenir

l'adresse matérielle de l'hôte. A cet effet, il envoie une requête ARP. L'hôte renvoie son adresse matérielle dans la réponse ARP. Le client Telnet établit alors la session.

Figure 5.26

Exemple de fonctionnement de RARP.



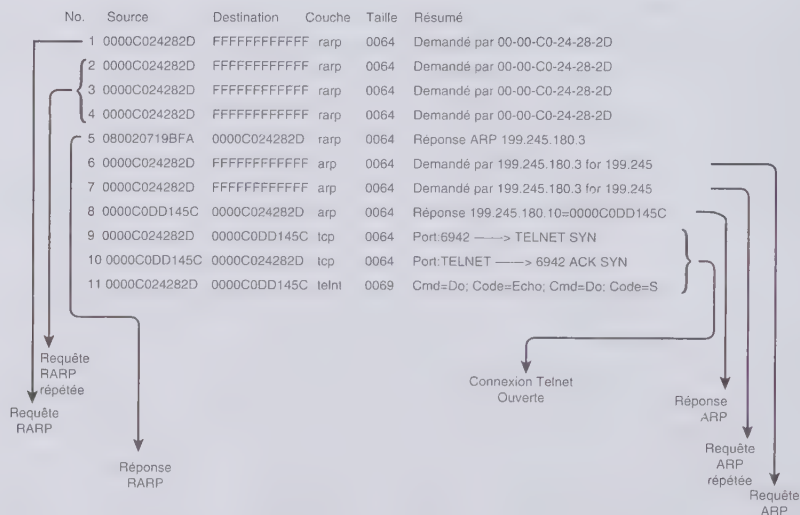
La Figure 5.27 montre la trace de paquets produite sur un vrai réseau, pour l'exemple de la Figure 5.26.

Remarquez que les paquets RARP 2 à 4 sont répétés. C'est imputable à la lenteur du serveur RARP dans ses réponses. Cela illustre un des dangers des tempêtes de diffusions RARP, qui surviennent lorsque le serveur RARP ne répond pas assez vite. La requête RARP du paquet 4 obtient une réponse avec la réponse RARP du paquet 5. A la réception de cette réponse, le client RARP s'affecte comme adresse IP la valeur 199.245.180.3 (valeur renvoyée dans la réponse RARP). Le client RARP essaie alors d'obtenir l'adresse matérielle de l'hôte en envoyant cette fois une requête ARP en diffusion. Les requêtes ARP sont envoyées dans les paquets 6 et 7. Le paquet 7 est une répétition de la requête ARP du paquet 6, car l'hôte Telnet ne répond pas assez vite à la première requête. L'hôte Telnet répond dans le paquet 8, dans lequel il fait figurer son adresse matérielle. Le client RARP ouvre alors une session Telnet dans les paquets 9 à 11.



Figure 5.27

Trace de protocole des paquets
RARP/ARP et Telnet.



Les paquets de requête et de réponse RARP sont analysés dans les deux sections suivantes.

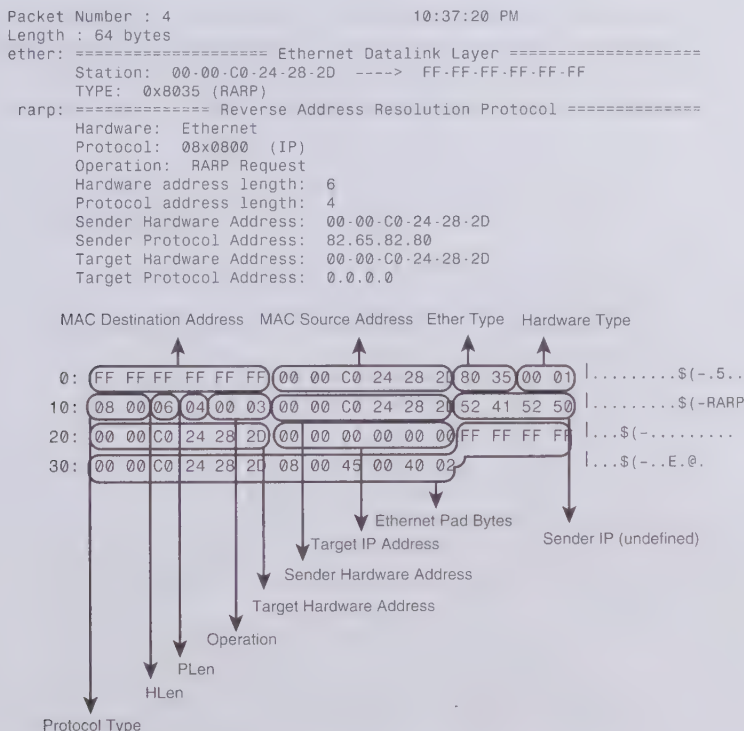
• Requête RARP

La Figure 5.28 montre le décodage détaillé du paquet de requête RARP en diffusion. Ethernet est la couche de liaison de données utilisée pour la transmission de la requête RARP. Le décodage du paquet montre une trame Ethernet dont les champs ont les valeurs suivantes (en hexadécimal) :

Adresse de destination =FFFFFFFFFFFF
 Adresse source =0000C024282D
 EtherType =8035

Le champ EtherType a pour valeur 8035 (hexadécimal), qui est la valeur réservée pour les paquets RARP. Cela signifie donc qu'il s'agit d'un paquet RARP encapsulé dans une trame Ethernet.

Le champ Opération du paquet RARP a pour valeur 3, ce qui indique une requête RARP. Les champs AM émetteur et IP émetteur ont pour valeur 0000C024282D et 82.65.82.80 respectivement. Le champ IP émetteur n'est pas défini, puisqu'il est censé contenir l'adresse IP dont la recherche est l'objet même de la requête.

Figure 5.28*Paquet de requête RARP.*

L'implémentation peut faire figurer dans ce champ une valeur quelconque. La plupart des implémentations utilisent la valeur 0.0.0.0. Assez bizarrement, dans notre exemple tiré d'une situation réelle, c'est la valeur 82.65.82.80 qui est utilisée. La valeur de ce champ n'a de toute façon aucune incidence sur le processus.

Les champs AM cible et IP cible ont pour valeur 0000C024282D et 0.0.0.0 respectivement. Dans l'implémentation TCP/IP de notre exemple, l'adresse IP cible n'est pas définie et la valeur en est fixée arbitrairement à 0.0.0.0.

• Réponse RARP

La Figure 5.29 décrit en détail le paquet de réponse RARP qui a été envoyé par le serveur RARP à l'adresse 144.19.74.33 en réponse au paquet de requête RARP de la Figure 5.28. Ethernet est la couche de liaison de données utilisée pour la

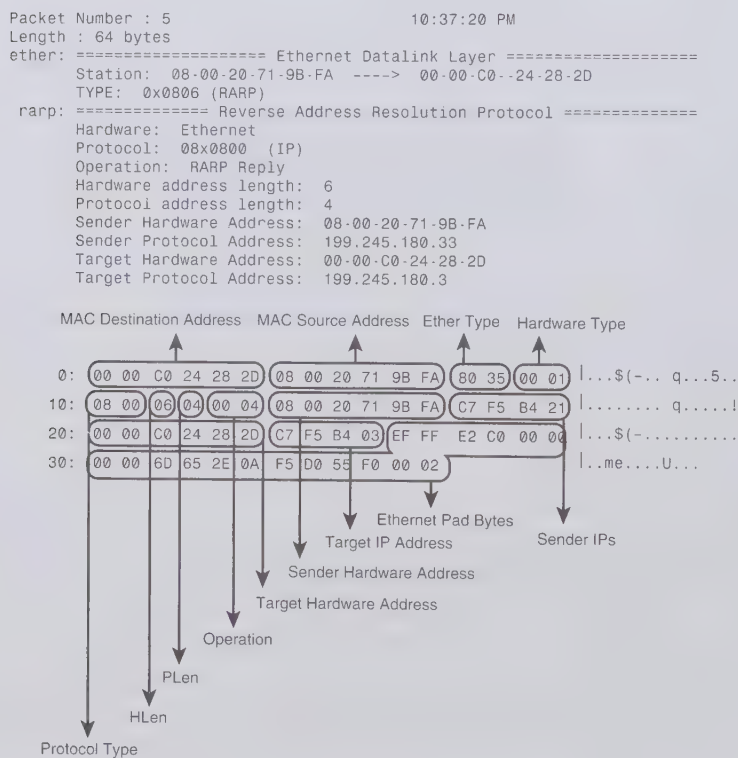


transmission de la réponse RARP. Le décodage du paquet montre une trame Ethernet dont les champs ont les valeurs suivantes (en hexadécimal) :

Adresse de destination = 0000C024282D
 Adresse source = 080020719BFA
 EtherType = 8035

Figure 5.29

Paquet de réponse RARP.



Le champ EtherType indique que cette trame Ethernet est envoyée au client RARP d'adresse MAC 0000C024282D. Notez que la réponse RARP n'est pas envoyée en diffusion, mais directement au client RARP. Le champ EtherType a pour valeur 8035 (hexadécimal). Cela confirme qu'il s'agit d'un paquet RARP encapsulé dans une trame Ethernet.

Le champ Opération du paquet RARP a pour valeur 4, ce qui indique une réponse RARP. Les champs AM émetteur et IP émetteur ont pour valeur 080020719BFA et 199.245.180.33 respectivement. Ce sont les paramètres du serveur RARP.

Les champs AM cible et IP cible ont respectivement pour valeur 0000C024282D et 199.245.180.3. Le champ AM cible est l'adresse matérielle du client RARP. L'adresse IP cible 199.245.180.33 est la réponse recherchée par la requête RARP. C'est cette adresse qui sera affectée au client RARP.

Affichage du cache ARP

En temps normal, l'administrateur de réseau n'ajoute pas d'entrées dans la table de cache ARP d'un hôte, car le protocole ARP est utilisé par le module IP pour ajouter et supprimer des entrées dans cette table. De nombreuses implémentations TCP/IP fournissent néanmoins un outil permettant d'afficher, d'ajouter, de supprimer ou de modifier des entrées de table de cache ARP. Cet outil diffère selon chaque système.

Ainsi, Unix et Windows NT fournissent un outil en ligne de commande appelé `arp`, qui permet la maintenance manuelle de la table de cache ARP. Cet outil permet d'apporter des corrections temporaires. Pour des détails sur la syntaxe de cette commande, reportez-vous à la documentation de votre système d'exploitation.

Autres protocoles de résolution d'adresse

Il existe d'autres protocoles de résolution d'adresse, conçus pour des usages spécifiques. La RFC 1293 décrit le protocole IARP (*Inverse ARP*), utilisé dans les connexions à relais de trames pour découvrir l'adresse IP de la machine située à l'autre bout de la connexion. Proxy ARP peut être employé dans des réseaux utilisant de vieux logiciels TCP/IP incapables de manipuler la notion de sous-réseau. Proxy ARP est décrit dans la RFC 1027. La RFC 1577 décrit le mécanisme d'ARP sur des réseaux ATM. La RFC 1390 décrit la transmission de IP et ARP sur des réseaux FDDI et la RFC 1374 décrit ARP sur des réseaux HIPPI.



Récapitulatif

Dans les réseaux à diffusion, tels que la plupart des réseaux locaux, il est nécessaire de disposer d'une méthode pour découvrir dynamiquement les adresses matérielles et IP. Ces méthodes sont appelées des protocoles de résolution d'adresse. ARP est conçu pour découvrir l'adresse matérielle du nœud cible à partir de son adresse IP. Typiquement, on utilise le protocole IP au début d'une connexion de réseau à un service résidant sur un réseau local.

Le protocole RARP est utilisé par des stations pour découvrir leur propre adresse IP. Les adresses IP sont stockées en un endroit central, tel qu'un serveur RARP. RARP est très souvent utilisé par les stations de travail sans disque. Le protocole RARP est utilisé pour affecter une adresse IP à une station sans disque. Puis, on utilise un protocole de transfert de fichier, tel que le *Trivial File Protocol* (TFTP), pour charger dans la station l'image du système d'exploitation. Le protocole TFTP est de taille assez petite pour être implémenté dans la ROM des stations sans disque.

Testez vos connaissances

1. Quels types de réseaux ont besoin d'un protocole de résolution d'adresse ?
2. Quelle est la taille de l'en-tête de requête/réponse ARP ?
3. Dessinez un schéma représentant les relations entre ARP, IP et Ethernet au niveau de la couche de protocole et de la couche d'implémentation.
4. Pourquoi les requêtes ARP sont-elles envoyées sous forme de trames de diffusion ?
5. Dans quelles circonstances une requête ARP sera envoyée sous forme de trame dirigée plutôt que de trame de diffusion ?
6. En examinant la structure des paquets ARP et RARP envoyés dans des trames Ethernet, on remarque que les champs AM émetteur et AM cible semblent redondants, car ces informations se trouvent aussi dans l'en-tête Ethernet. Pourquoi est-il nécessaire de conserver ces informations redondantes ?
7. Pourquoi le trafic de diffusion, tel que celui produit par des envois répétés de requêtes ARP, ralentit les nœuds n'accédant pas au réseau à ce moment ?
8. A quelle couche du modèle OSI correspond le protocole ARP ? Justifiez votre réponse.

9. Quels problèmes seraient susceptibles de survenir si l'on configurait statiquement un hôte en indiquant des adresses MAC et les adresses IP correspondantes ?
10. Pourquoi la requête ARP initiale est-elle envoyée en diffusion ?
11. Pourquoi la réponse ARP n'est-elle pas envoyée en diffusion ?
12. Certaines implémentations d'ARP envoient une requête ARP en diffusion initiale contenant l'adresse IP locale dans le champ IP cible. Pourquoi ?
13. Représentez la structure d'un paquet ARP et expliquez l'utilisation des différents champs. Représentez aussi la structure d'un paquet RARP et montrez en quoi il diffère du précédent.
14. ARP et RARP sont démultiplexés au niveau de la couche de liaison de données. Comment cela est-il réalisé ?
15. Un hôte d'un réseau local démarre et essaie de se connecter à un serveur sur le même réseau. En supposant qu'il ne s'agit pas d'une station sans disque, une requête ARP sera-t-elle générée par l'hôte ? Si l'hôte est une station sans disque et qu'on utilise RARP pour découvrir l'adresse IP de cet hôte, quel type de paquets sera généré pour établir la connexion ? La station sans disque générera-t-elle des paquets ARP ?
16. Dans quelles circonstances un hôte n'a-t-il pas besoin d'envoyer une requête ARP ?
17. Comment s'effectue un test ARP de doublons d'adresses IP ?
18. Lorsqu'une requête/réponse ARP est transmise sur un réseau Ethernet, la trame Ethernet nécessite-t-elle d'être remplie ? Si c'est le cas, combien d'octets doivent être ajoutés ? Pourquoi est-ce nécessaire ?
19. Quel est l'impact des tempêtes ARP/RARP sur les réseaux à ponts ?
20. Sur un réseau, deux stations possédant la même adresse : 198.24.54.1. Elles tentent d'établir une session Telnet avec un hôte central dont l'adresse IP est 198.24.54.33. Les adresses MAC des deux stations sont 0000C003A216B et 08000000DFCA90. Expliquez ce qui va se passer. Illustrez avec des schémas représentant le contenu de la table de cache ARP.
21. Que se passerait-il dans le cas de figure de doublons d'adresses IP au niveau des clients TCP/IP (voir plus haut dans ce chapitre) si une implémentation de TCP/IP choisissait d'ignorer les recommandations de la RFC 826 et de ne pas remplacer une entrée existante du cache ARP par une nouvelle adresse matérielle ?



Cela résoudrait-il le problème des doublons d'adresses IP ? Quels autres problèmes se poseraient alors ?

22. Dans quelles circonstances peut survenir une tempête RARP ?
23. Quel est le rôle des serveurs RARP primaires et auxiliaires ? Décrivez la façon dont un serveur auxiliaire peut répondre à une requête ARP.
24. Voici le dump hexadécimal d'un paquet. De quel type de paquet s'agit-il ?

```

0: 00 00 C0 24 28 2D 00 00 C0 DD 14 5C 08 00 45 00 |...$(.....\..E.
10: 00 29 E9 86 40 00 80 06 19 4F C7 F5 B4 0A C7 F5 |...)@....0.....
20: B4 03 00 17 1B 1E 04 E2 87 09 01 31 E0 1C 50 18 |.....1..P.
30: 3F E5 BD 7E 00 00 32 FF 03 FF F0 00 |?...~..2.....

```

25. Voici le dump hexadécimal d'un paquet. De quel type de paquet s'agit-il ? Décodez entièrement le paquet et expliquez-en la fonction.

```

0: FF FF FF FF FF FF 00 00 C0 24 28 2D 08 06 00 01 |.....$(.....
10: 08 00 06 04 00 01 00 00 C0 24 28 2D 90 13 4A 2C |.....$(...J,
20: FF FF FF FF FF FF 90 13 4A 2C 00 00 FF FF FF FF |.....J,.....
30: 00 00 C0 24 28 2D 08 00 45 00 40 02 |...$(...E.@.

```

26. Voici le dump hexadécimal d'un paquet. De quel type de paquet s'agit-il ? Décodez entièrement le paquet et expliquez-en la fonction.

```

0: FF FF FF FF FF FF 00 00 C0 24 28 2D 08 06 00 01 |.....$(.....
10: 08 00 06 04 00 01 00 00 C0 24 28 2D 90 13 4A 2C |.....$(...J,
20: FF FF FF FF FF FF 90 13 4A C9 00 00 FF FF FF FF |.....J,.....
30: 00 00 C0 24 28 2D 08 00 45 00 40 02 |...$(...E.@.

```

27. Voici le dump hexadécimal d'un paquet. De quel type de paquet s'agit-il ? Décodez entièrement le paquet et expliquez-en la fonction.

```

0: 00 00 C0 24 28 2D 00 00 C0 A2 0F 8E 08 06 00 01 |...$(.....
10: 08 00 06 04 00 02 00 00 C0 A2 0F 8E 90 13 4A C9 |.....J.
20: 00 00 C0 24 28 2D 90 13 4A 2C 00 3E 80 64 50 10 |...$(...J,.>.dP.
30: 77 FE 3F 19 00 00 74 0A 6C 74 72 65 |w.?...t.ltre

```

28. Voici le dump hexadécimal d'un paquet. De quel type de paquet s'agit-il ? Décodez entièrement le paquet et expliquez-en la fonction.

```

0: FF FF FF FF FF FF 00 00 C0 24 28 2D 80 35 00 01 |.....$(...5..
10: 08 00 06 04 00 03 00 00 C0 24 28 2D 52 41 52 50 |.....$(...RARP
20: 00 00 C0 24 28 2D 00 00 00 00 00 00 FF FF FF FF |...$(.....
30: 00 00 C0 24 28 2D 08 00 45 00 40 02 |...$(...E.@.

```

29. Voici le dump hexadécimal d'un paquet. De quel type de paquet s'agit-il ?
Décodez entièrement le paquet et expliquez-en la fonction.

```

0: 00 00 C0 24 28 2D 08 00 20 71 9B FA 80 35 00 01 |...$(... q...5..
10: 08 00 06 04 00 04 08 00 20 71 9B FA C7 F5 B4 21 |..... q.....!
20: 00 00 C0 24 28 2D C7 F5 B4 03 EF FF E2 C0 00 00 |...$(.....
30: 00 00 6D 65 2E 0A F5 D0 55 F0 00 02 |...me....U...

```

30. Il existe 50 nœuds sur un réseau, chacun envoyant deux diffusions par seconde du fait d'une défaillance du serveur RARP principal. A supposer qu'il s'agisse d'un réseau Ethernet, quel pourcentage de la bande passante du réseau est consommé par le trafic RARP ?
31. Donnez une formule générale exprimant le pourcentage de bande passant consommé par les paquets de diffusion ARP/RARP. On pose les paramètres suivants pour le réseau :

Taux de transfert maximal pour le réseau $= D$ bit/s.

Nombre de nœuds envoyant des diffusions ARP/RARP $= N$

Taux d'envoi des transmissions ARP/RARP $= R$ paquets/seconde.

32. Utilisez cette formule pour contrôler votre réponse à la question 30.



6

Internet Protocol

Le protocole IP (*Internet Protocol*) fournit le premier niveau d'abstraction permettant d'obtenir une vue virtuelle du réseau, où tous les nœuds sont traités comme des nœuds IP. La vue abstraite du réseau fournie par IP consiste en une représentation logique où le réseau est considéré comme un réseau idéalisé, dont les propriétés seront examinées dans ce chapitre.

Du fait de la couche d'abstraction fournie par IP, les protocoles se trouvant au-dessus d'IP – tels que les protocoles TCP et UDP – peuvent traiter le réseau comme un réseau purement IP. Il se peut en réalité que ce ne soit pas le cas et que le réseau prenne en charge d'autres protocoles en plus de TCP/IP. Les connexions de réseau du nœud IP sont identifiées par la valeur sur 32 bits appelée adresse IP. IP fournit des services sans connexion à des services de couches supérieures. Le service sans connexion est implémenté à l'aide de datagrammes contenant les adresses IP source et destination, ainsi que d'autres paramètres nécessaires au fonctionnement d'IP. Ce chapitre se propose d'analyser les concepts permettant de comprendre comment IP fonctionne dans des réseaux réels.

L'abstraction de IP

La couche IP s'appuie sur le matériel de réseau sous-jacent pour sa transmission. Cela signifie que le datagramme IP est encapsulé par les trames du réseau sous-jacent, qu'il s'agisse d'Ethernet, de Token Ring ou de X.25.

Les protocoles de couches supérieures, tels que TCP et UDP, n'ont pas besoin de connaître cette encapsulation ni le matériel sous-jacent. Les protocoles de couches supérieures peuvent espérer une certaine qualité de service, en termes de débit et de temps d'attente, par exemple. Il s'agit de ce que l'on appelle les paramètres de QoS (qualité de service). La couche supérieure passe à la couche IP les paramètres de QoS en même temps que les données. La couche IP peut alors tenter de faire correspondre les paramètres QoS et les services fournis par le matériel de réseau sous-jacent, capable ou non de fournir les services requis.

La Figure 6.1 montre que l'hôte IP possède trois connexions de réseau. Ces connexions concernent les réseaux Ethernet, Token Ring et X.25. Dans cet exemple, IP est exécuté sur chacune de ces trois connexions de réseau. Ces dernières sont

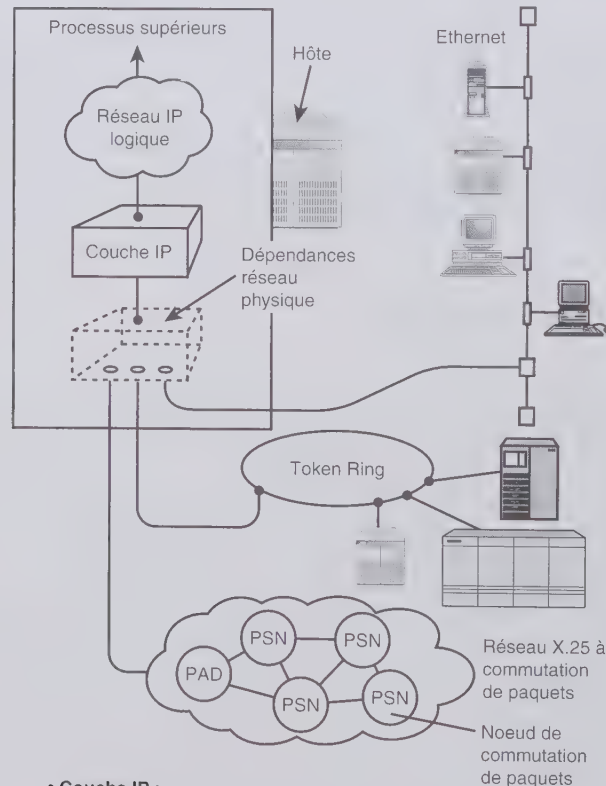


identifiées par un identificateur à 32 bits distinct, appelé l'adresse IP. La couche IP présente pour les couches supérieures une abstraction de chacun des trois réseaux. Cette abstraction est indépendante des caractéristiques physiques des réseaux telles que la taille d'adresse, la taille de l'unité de transmission maximale (MTU, *Maximum Transmission Unit*), la bande passante du réseau, le débit maximal, etc. La taille de MTU est la taille maximale du champ de données pour le réseau physique sous-jacent. Comme le champ de données contient des datagrammes IP pour des réseaux TCP/IP, la taille de MTU correspond aussi à la taille maximale du datagramme IP pouvant être transmis sur un réseau physique.

Dans le réseau de la Figure 6.1, la taille de MTU du réseau Ethernet est de 1 500 octets. Elle est de 4 400 octets pour le réseau Token Ring à 4 Mbit/s IEEE 802.5.

Figure 6.1

L'abstraction IP (source : Learning Tree).



• **Couche IP :**

- Fournit un niveau d'abstraction logique
- Cache les dépendances de la Couche Physique

• **Les processus supérieurs voient un réseau IP logique**

PAD : Assembleur/Désassembleur de Paquets

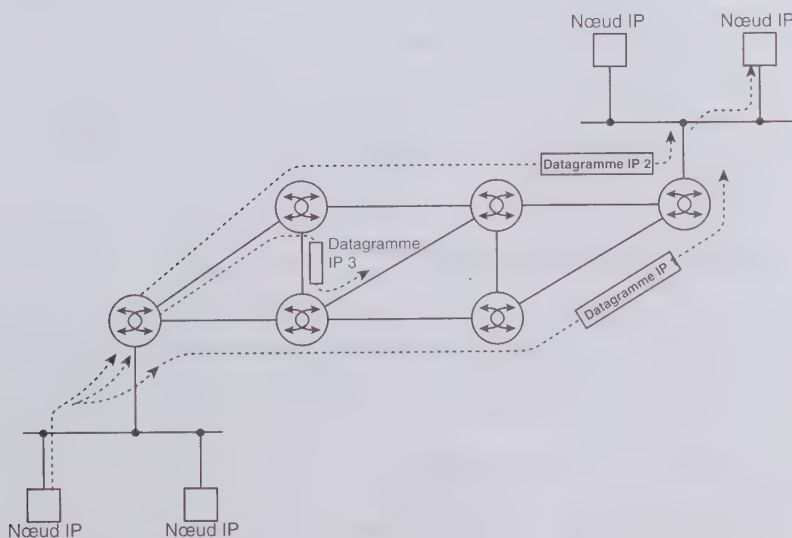
Elle est de 17 940 octets pour le réseau à 16 Mbit/s IEEE802.5. Dans le cas du réseau X25, la taille de MTU est variable et peut aller de 64 octets à 4 Ko. Les réseaux IP doivent pouvoir traiter des datagrammes d'au moins 576 octets de long. Comme le plus petit datagramme IP que tout routeur doit pouvoir traiter fait 576 octets de long, on n'utilise pas les plus basses longueurs de MTU dans un réseau X25 transmettant des données TCP/IP.

Quelles que puissent être les différences en termes de taille de MTU et de débit selon les réseaux, la couche IP fait une abstraction de ces réseaux pour former un réseau IP logique commun et indépendant des différences physiques.

Le réseau IP abstrait est sans connexion. Chaque datagramme est routé indépendamment. Il est donc possible que des datagrammes successifs soient transmis le long de chemins différents (voir Figure 6.2).

Figure 6.2

Routage indépendant des datagrammes IP.



Du fait de la différence du temps de propagation le long des divers chemins, il est possible que les datagrammes arrivent à destination dans un ordre différent de l'ordre de départ. La couche IP ne tente pas d'assurer l'arrivée des datagrammes dans leur ordre correct, ni de garantir une livraison fiable à la destination. La livraison de datagrammes dans l'ordre correct est appelé le *séquencement*. Le problème du séquencement et celui de la fiabilité sont traités par un protocole de couche supérieure, tel que TCP.



Comme IP n'est pas tenu d'assurer le séquençement ou la fiabilité des transferts de données, il est facile de déployer IP sur des matériels de réseau très variés. Les protocoles de couche supérieure peuvent ajouter des niveaux supplémentaires de fiabilité, selon les besoins des applications.

Il convient de bien comprendre que le réseau IP en lui-même, sans assistance de protocoles de plus haut niveau, est fondamentalement non fiable. Le réseau IP emploie pour la livraison la méthode de *remise au mieux*, c'est-à-dire la meilleure livraison possible du datagramme, sans que cette livraison soit garantie pour autant. C'est la raison pour laquelle on utilise presque toujours IP conjointement à des protocoles de plus haut niveau, qui assurent les fonctionnalités négligées par IP.

Le service IP est sans connexion, et chaque datagramme est routé indépendamment des autres. Le datagramme inclut les adresses IP de l'émetteur et de la destination. L'adresse de destination est utilisée par les routeurs sur le parcours afin de transmettre le datagramme vers la bonne destination.

L'abstraction IP permet de transmettre des datagrammes quelle que soit la taille de MTU du réseau. Les nœuds IP peuvent fragmenter un datagramme trop grand, si besoin est. Les sections qui suivent analysent plus en détail les mécanismes alors mis en œuvre.

• Taille du datagramme IP

IP a été conçu pour s'accommoder de types de matériels réseau très différents. Comme nous l'avons déjà vu, la taille maximale de données pouvant être transmises par la trame de couche de liaison diffère selon les réseaux. Le Tableau 6.1 dresse la liste des tailles de MTU de différents types de matériels.

Tableau 6.1 : Tailles de MTU

Type de réseau	MTU (en octets)
Ethernet	1 500
IEEE 802.3	1 492
Token Ring	4 440 à 17 940 (la taille dépend du temps de conservation du jeton)
FDDI	4 352
IEEE 802.4	8 166

Tableau 6.1 : Tailles de MTU (*suite*)

Type de réseau	MTU (en octets)
SMDS	9 180
X.25	1 007 (ARPANET via 576 réseaux publics de données, mais cette valeur peut être augmentée par négociation)

Les tailles de MTU des réseaux Token Ring (voir Tableau 6.1) dépendent du temps de conservation du jeton (THT, *Token Holding Time*), qui est d'environ 8,58 microsecondes. Pour un taux de transmission de 4 Mbit/s, la taille de trame maximale pour un réseau Token Ring est calculée comme suit :

$$\begin{aligned}
 &\text{Taille maximale de trame} \\
 &\text{pour un réseau Token Ring} \\
 &(\text{à 4 Mbit/s}) \\
 &= 8,58 \text{ ms} \times 4 \text{ Mbit/s} \\
 &= 8,58 \times 4 \times 1\,024 \times 1\,024 \\
 &= 36\,000 \text{ bits} \\
 &= 36\,000/8 \text{ octets} \\
 &= 4\,500 \text{ octets}
 \end{aligned}$$

(rappelons que 1 Mbit/s correspond à 2^{20} bit/s soit $1\,024 \times 1\,024$ bit/s)

Le plus grand en-tête de trame de réseau Token Ring se calcule comme suit :

$$\begin{aligned}
 &\text{En-tête MAC} &&= 15 \text{ octets} \\
 &\text{Champ d'informations de routage} &&= 30 \text{ octets} \\
 &\text{En-tête LLC} &&= 4 \text{ octets} \\
 &\text{En-tête SNAP} &&= 5 \text{ octets} \\
 &\text{En-queue MAC} &&= 6 \text{ octets} \\
 &\text{Total en-tête de trame} &&= 60 \text{ octets} \\
 &\text{Taille de MTU pour les réseaux} \\
 &\text{Token Ring (à 4 Mbit/s)} &&= \text{taille max. de trame} - \text{taille d'en-tête} \\
 &&&\text{de trame Token Ring} \\
 &&&= 4\,500 - 60 \text{ octets} \\
 &&&= 4\,440 \text{ octets}
 \end{aligned}$$



Pour un réseau Token Ring à 16 Mbit/s, la taille maximale de trame Token Ring se calcule comme suit :

Taille maximale de trame pour un réseau Token Ring (à 16 Mbit/s)	$= 8,58 \text{ ms} \times 16 \text{ Mbit/s}$ $= 8,58 \times 16 \times 1\,024 \times 1\,024$ $= 144\,000 \text{ bits}$ $= 144\,000/8 \text{ octets}$ $= 18\,000 \text{ octets}$
Taille de MTU pour les réseaux Token Ring (à 16 Mbit/s)	$= \text{taille max. de trame} - \text{taille d'en-tête de trame Token Ring}$ $= 18\,000 - 60 \text{ octets}$ $= 17\,940 \text{ octets}$

La taille de MTU pour FDDI du Tableau 6.1 se calcule comme suit :

Taille maximale de trame FDDI utilisant l'encodage 4B/5B	$= 4\,500 \text{ octets}$
--	---------------------------

Le plus grand en-tête de trame FDDI se calcule comme suit :

En-tête MAC	$= 16 \text{ octets}$
En-tête LLC	$= 4 \text{ octets}$
En-tête SNAP	$= 5 \text{ octets}$
En-queue MAC	$= 7 \text{ octets (approximativement)}$
Total en-tête de trame	$= 32 \text{ octets}$

Pour anticiper de futures extensions, il est recommandé d'allouer 148 octets à l'en-tête de trame FDDI.

Taille de MTU pour les réseaux FDDI (à 100 Mbit/s)	$= \text{taille max. de trame} - \text{taille ajustée d'en-tête de trame FDDI}$ $= 4\,500 - 148 \text{ octets}$ $= 4\,352 \text{ octets}$
--	---

Le datagramme IP peut faire jusqu'à 65 536 octets de long. Pour la plupart des réseaux, la taille de MTU est bien inférieure à ces 65 536 octets. La couche IP de l'émetteur limite généralement la taille du datagramme pour qu'elle n'excède pas

celle de MTU du réseau local. Si un datagramme IP doit franchir des réseaux intermédiaires, sa taille ne peut pas dépasser celles des MTU de ces réseaux, si l'on veut que le datagramme soit remis en un seul morceau. Les réseaux IP ne disposent pas de mécanismes efficaces pour déterminer à l'avance les tailles de MTU des réseaux intermédiaires. Si c'était le cas, l'émetteur pourrait limiter la taille de ses datagrammes en fonction des réseaux intermédiaires. En fait, les réseaux IP doivent s'appuyer sur un mécanisme de fragmentation pour remettre des datagrammes de taille arbitraire à n'importe quel réseau.

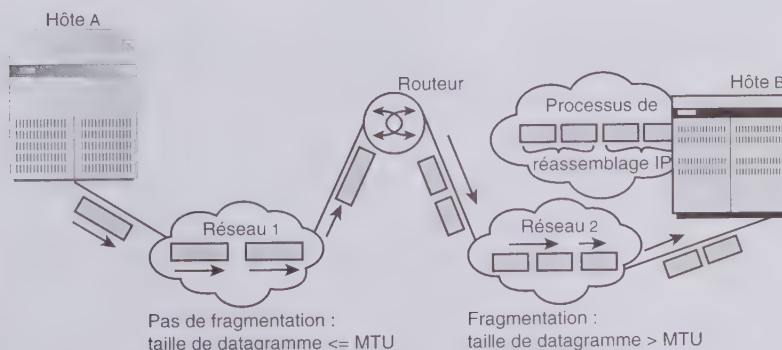
• Fragmentation IP

Si la taille d'un datagramme IP dépasse la taille de MTU du réseau qu'il doit franchir, le datagramme ne peut être envoyé en un seul morceau. Il doit être découpé en morceaux plus petits. On appelle cela la *fragmentation*. Chaque fragment du datagramme originel est envoyé sous forme de datagramme IP. L'en-tête IP contient les informations permettant d'identifier le fragment. Ces informations sont utilisées par l'hôte de destination pour assembler les divers fragments.

La Figure 6.3 illustre l'envoi d'un datagramme IP à travers plusieurs réseaux. Dans cet exemple, la taille de MTU du réseau 2 est plus petite que celle du datagramme. Le routeur s'en aperçoit et fragmente le datagramme IP en plusieurs datagrammes IP plus petits dont la taille ne dépasse pas celle de MTU du réseau 2. Les datagrammes fragmentés sont routés indépendamment les uns des autres et arrivent à l'hôte B. Le réassemblage est effectué par le module IP de destination, jamais par les routeurs intermédiaires.

Figure 6.3

Fragmentation d'un datagramme IP
(source : Learning Tree).



Que se passe-t-il dans la Figure 6.3 lorsque l'hôte B a besoin de répondre à l'hôte A ? L'hôte B ajustera la taille de datagramme pour que la taille de ce dernier ne dépasse pas celle de MTU du réseau 2. Le datagramme provenant de l'hôte B peut être remis à l'hôte A sans fragmentation. Quoi qu'il en soit, les routeurs intermédiaires ne tenteront pas de combiner plusieurs datagrammes en un seul datagramme plus grand pour optimiser la transmission.

Dans les réseaux IP, seuls les datagrammes fragmentés sont assemblés, et ce, seulement au niveau de la destination. Les datagrammes entiers ne sont jamais combinés pour former des datagrammes plus grands, ce qui optimiserait pourtant la transmission.

Format du datagramme IP

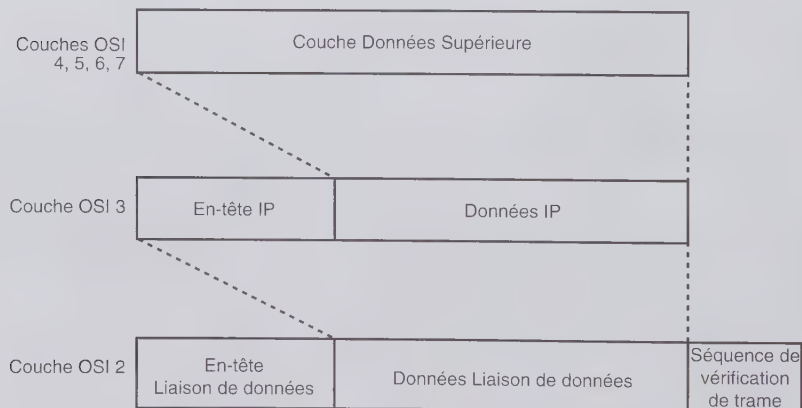
Le datagramme IP contient un en-tête IP suivi des données IP provenant des protocoles des couches supérieures (voir Figure 6.4). L'en-tête IP est conçu pour s'accommoder des particularités de la couche IP.

D'après ce que nous avons vu de la couche IP dans les sections précédentes, il apparaît que l'en-tête IP doit contenir au minimum les informations suivantes :

- **L'adresse IP de la source et de la destination.** Elles sont nécessaires, car IP est un protocole sans connexion.
- **Qualité de service.** Ce champ est nécessaire pour spécifier le type de service attendu de la part du réseau sous-jacent.

Figure 6.4

Format d'un datagramme IP.



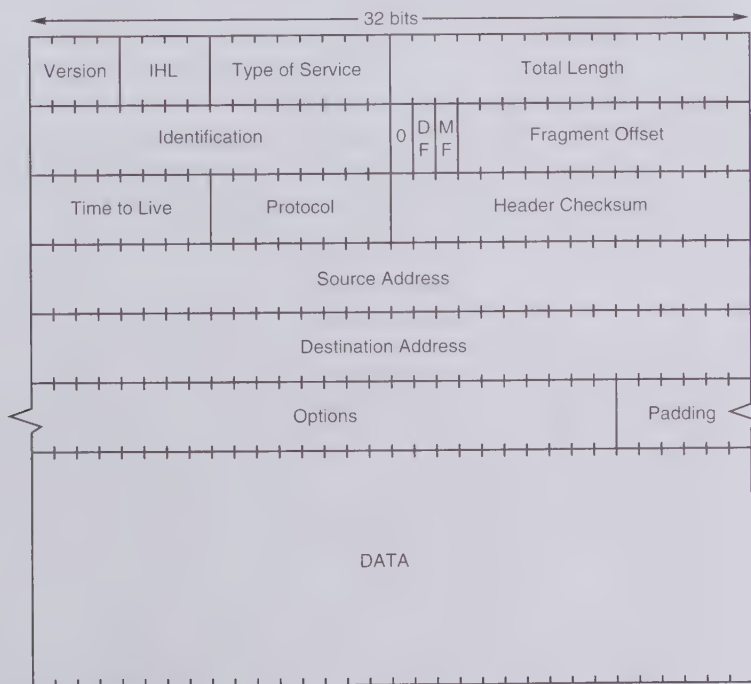
- **Informations de fragmentation.** Comme le datagramme IP peut être fragmenté lors du passage dans un autre réseau, il est nécessaire d'inclure des champs permettant d'identifier la position du fragment dans la datagramme originel.
- **Taille du datagramme.** Les datagrammes IP peuvent être de taille variable, ce qui rend nécessaire un champ indiquant la taille totale du datagramme IP.
- **Taille d'en-tête IP.** Les datagrammes IP peuvent contenir des champs optionnels spécifiant les options IP utilisées pour la sécurité, le routage à la source, etc. Ces options font que la taille de l'en-tête IP est variable, et qu'il est par conséquent nécessaire d'indiquer la taille de ce dernier.

La section qui suit détaille les différents champs composant un datagramme IP.

• Format de l'en-tête IP

L'en-tête IP a une longueur minimale de 20 octets. Si des options IP sont spécifiées dans le datagramme, cette longueur peut être plus grande. La Figure 6.5 montre les champs et la structure générale de l'en-tête IP.

Figure 6.5
En-tête IP.



Le format de l'en-tête IP est décrit en détail dans les RFC 791 et RFC 1122. La RFC 1349 détaille l'utilisation du champ TOS (*Type of Service*) de l'en-tête IP.

Le champ Version

Dans la Figure 6.5, le champ *Version* fait quatre bits de long et indique le format de l'en-tête IP. Cela permet une éventuelle définition de futures structures de paquets. Le numéro de version actuel est 4, et le protocole correspondant est appelé IPv4. La version suivante est la version 6 (IPv6) et permet des adresses IP à 128 bits. Le Tableau 6.2 montre les autres valeurs possibles du champ version

INFO



Les numéros de version 7, 8 et 9 ont été affectés pour des protocoles IP expérimentaux en projet, destinés à résoudre le problème de la limitation des adresses IP à 32 bits. Ces protocoles ont été supplantés par IPv6.

Tableau 6.2 : Valeurs du champ IP Version

Version IP	Description
0	Réservé
1 à 3	Non affectés
4	IPv4
5	Stream IP Datagram Mode (IP expérimental)
6	IPv6. Egalement appelé IPng.
7	TP/IX: The Next Internet
8	Protocole Internet "P"
9	TUBA
10 à 14	Non affectés
15	Réservé

Le champ Version est utilisé par l'émetteur, le récepteur et tout routeur intermédiaire pour déterminer le format de l'en-tête IP. Le logiciel IP doit vérifier la valeur de ce champ pour s'assurer que le format d'en-tête IP est bien celui qui est attendu. Ainsi, par exemple, si le logiciel IP ne peut traiter que des datagrammes de la version 4, il rejettera les datagrammes pour lesquelles la valeur du champ Version est différente de 4.

Voici quelques détails sur les entrées du Tableau 6.2 :

- Le protocole Stream IP est un protocole expérimental fournissant un service garanti de point à point. Stream IP est décrit dans la RFC 1819.
- Le protocole Internet "P", TP/IX et TUBA furent tous un temps candidats à la succession de IPv4. Mais IPv6 a été adopté depuis et l'intérêt de ces protocoles est maintenant secondaire.

- Le protocole Internet "P" est un nouveau protocole doté de fonctionnalités avancées et d'adresses à taille variable. Il a récemment fusionné avec SIP (*Simple Internet Protocol*). SIP utilise un adressage à 64 bits et a lui-même fusionné avec IP Address Encapsulation ("encapsulation d'adresses IP" ou IPAE), qui décrit la façon dont s'effectuera la transition entre l'adressage IPv4 et des adressages plus longs. SIP est décrit dans la RFC 1710.
- Le protocole TP/IX (voir Tableau 6.2) est un protocole relativement ancien qui est devenu la base de la Common Architecture for the Internet ("Architecture commune pour l'Internet" ou CATNIP). Le protocole CATNIP est maintenant obsolète. On le trouve décrit dans la RFC 1707.
- TUBA est l'acronyme de *TCP and UDP with Bigger Addresses* (TCP et UDP avec de plus grosses adresses). TUBA est fondé sur CLNP, et on le trouve décrit dans les RFC 1347, RFC 1526 et RFC 1561.

Le champ Internet Header Length ("Longueur d'en-tête Internet")

Le champ IHL (*Internet Header Length*) contient la longueur de l'en-tête, en mots de 32 bits. Ce champ fait 4 bits de long et il est obligatoire, car l'en-tête IP contient un champ d'options de longueur variable. Tous les autres champs ont une longueur fixe. Le champ Option est complété si nécessaire, pour obtenir un multiple de mots de 32 bits. La plupart des en-têtes IP ne spécifient pas d'options. Sans le champ d'options, la longueur de l'en-tête IP est de 20 octets. La valeur de IHL typique est donc de cinq mots de 32 bits. Le plus grand en-tête IP a une longueur de 60 octets et spécifie un certain nombre d'options. Dans cette hypothèse, la valeur de IHL serait de quinze mots de 32 bits.

Le champ Type of Service ("Type de service")

Le champ Type of Service (TOS) informe les réseaux de la qualité de service désirée, spécifiant ainsi la préséance, les délais, le débit et la fiabilité. La signification de ce champ de 8 bits est détaillée dans la Figure 6.6.

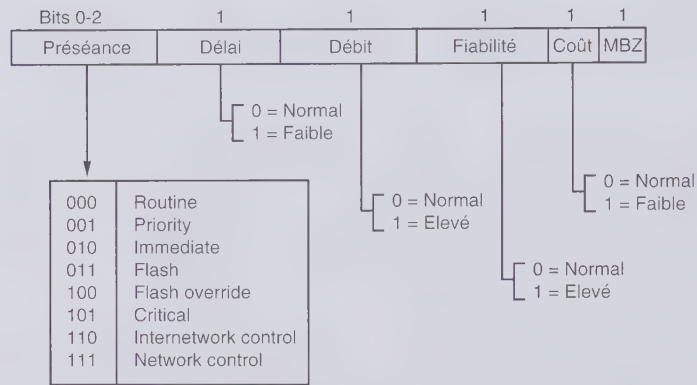
Les trois premiers bits ont trait à la valeur de préséance associée au datagramme IP. Voici quelques valeurs possibles pour le champ Precedence ("préséance") :

- **Flash.** Priorité maximale sur tous les circuits.
- **Immediate (immédiat).** Dans les quatre heures.
- **Priority (priorité).** Le même jour.
- **Routine.** D'ici au lendemain.



Figure 6.6

Le champ Type of Service d'un paquet IP.



Les valeurs de Precedence font 3 bits de long. Le Tableau 6.3 dresse la liste des différentes valeurs que peut prendre ce champ :

Tableau 6.3 : Valeurs du champ Precedence

Préséance	Signification
000	Routine
001	Priority
010	Immediate
011	Flash
100	Flash-override
101	Critical
110	Internetwork control
111	Network control

Le champ Precedence était à l'origine destiné aux applications du ministère de la Défense des Etats-Unis. L'utilisation de valeurs non nulles pour ce champ dépasse le cadre des spécifications IP standards. Les développeurs désireux de connaître les implications de ce champ doivent contacter le DISA – qui dépend du ministère de la Défense des Etats-Unis. L'utilisation de ce champ impliquera sans doute que sa valeur soit passée entre les différentes couches, un peu comme l'est le champ TOS. La couche IP doit fournir un mécanisme permettant à la couche de transport de définir le champ TOS de tout datagramme envoyé. Par défaut, la valeur du champ est 0. La couche IP doit transmettre les valeurs de TOS qu'elle a reçues à la couche de transport.

Les 4 bits qui suivent le champ Precedence correspondent à un code pour le type de service. Les significations de ces codes sont expliquées dans le Tableau 6.4.

Tableau 6.4 : Codes du champ Type of Service (TOS)

Code TOS	Signification
1000	Retard minimal
0100	Débit maximal
0010	Fiabilité maximale
0001	Coût monétaire minimal
0000	Service normal

Le dernier bit du champ TOS doit être mis à zéro. Ce champ d'un bit est appelé le champ *must be zero* (MBZ ou "doit valoir zéro") et il est réservé à un usage ultérieur. Les protocoles IP participant à des expériences sur Internet peuvent utiliser ce champ. Les routeurs et les destinataires de datagrammes ignorent la valeur de ce champ. Ce champ est copié lors de la fragmentation, de sorte que tous les fragments du datagramme possèdent le même type de service.

La plupart des implémentations de IP et des protocoles de routage – RIP, HELLO, etc. – ignorent le champ TOS. Quoi qu'il en soit, il semble que ce champ soit amené à jouer un rôle plus important dans le routage, en particulier avec des protocoles tels que OSPF. Ce champ pourrait ainsi entrer en considération dans les algorithmes de routage et de mise en file d'attente. Il peut aussi servir pour le partage de lignes en série au niveau de la couche de liaison, entre différentes classes de trafic TCP.

Le champ TOS peut servir d'indication aux algorithmes de routage pour sélectionner le chemin le plus approprié. Si un routeur connaît plusieurs routes pour une même destination, il peut sélectionner celle qui correspond le mieux à ce que spécifie la valeur du champ TOS. Prenons un exemple. Un routeur connaît deux chemins différents pour une même direction : l'un passant par une connexion à haut délai et à haut débit, une connexion par satellite par exemple, et l'autre passant par une connexion à faible délai et à faible débit, comme une ligne spécialisée. Si le trafic de l'application véhicule des saisies de touches par l'utilisateur, un court délai est nécessaire, mais le débit est moins important. Si le trafic de l'application implique un fort volume de transfert de données, c'est le débit qui est important et le délai qui l'est moins. Dans ce dernier cas de figure, le routeur choisira la liaison par satellite.



INFO

L'utilisation du champ TOS est décrite en détail dans la RFC 1349.



Le champ Total Length (Longueur totale)

Le champ Total Length contient la longueur de l'en-tête et des données IP, en octets. Ce champ fait 16 bits de long, ce qui limite à 65 535 la taille totale du datagramme. Ce nombre est la valeur maximale pouvant être représentée par 16 bits :

1111111111111111

Le nombre précédent est égal à $2^{16} - 1$, soit 65 535.

Les datagrammes de 65 535 octets de long ne sont pas pratiques pour la plupart des hôtes et réseaux. La plupart des réseaux ont une taille de MTU bien inférieure à ce nombre. De plus, les tampons de communication deviennent malaisés à gérer quand la taille des datagrammes IP devient trop importante. La taille des datagrammes IP sur la plupart des réseaux et des hôtes dépasse rarement 16 Ko.

Tous les nœuds IP doivent s'attendre à recevoir des datagrammes d'une longueur minimale de 576 octets, qu'ils soient entiers ou fragmentés. Cette taille minimale de 576 octets correspond à 512 octets de données plus 64 octets d'en-tête et de données propres au protocole.

Les datagrammes dont la taille dépasse celle de MTU du réseau sous-jacent doivent être fragmentés.

Le champ Identification

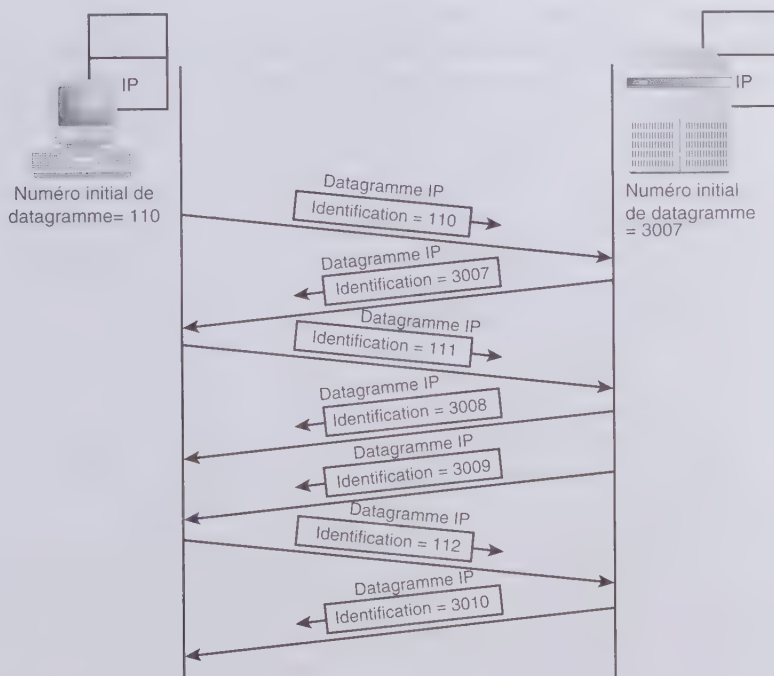
Le champ Identification est défini de manière distincte pour chaque datagramme et contient le numéro du datagramme. Chaque émetteur numérote le datagramme en partant d'une certaine valeur initiale (voir Figure 6.7). La plupart des implémentations de IP utilisent un compteur de mémoire globale qui est incrémenté à chaque datagramme IP envoyé. Si vous capturez du trafic de réseau provenant d'hôtes connectés depuis une longue période, tels que des serveurs, vous verrez des valeurs de champ Identification partant d'une valeur initiale apparemment arbitraire. En fait, les hôtes ont envoyé des datagrammes IP lors de sessions antérieures, et le champ Identification a été incrémenté pour chaque datagramme envoyé.

Le champ Identification fait 16 bits de long. Cela permet de numérotter les datagrammes de 0 à 65 535.

Ce champ sert principalement à identifier des fragments de datagrammes IP afin de les relier à un datagramme IP originel donné. On l'utilise conjointement aux flags Don't Fragment (DF, ou "Ne pas fragmenter"), More Fragments (MF, ou "plus de fragments") et les champs Fragment Offset ("décalage de fragment") pour réassembler le datagramme.

Figure 6.7

Valeurs du champ
Identification.



Flags de fragmentation et champs Fragment Offset ("décalage de fragment")

Si le flag DF est mis à 1, cela signifie que le datagramme ne doit pas être fragmenté. On peut spécifier un tel mode si l'émetteur estime que le destinataire ne dispose pas des capacités suffisantes pour réassembler les différents fragments d'un datagramme. C'est le cas d'un programme d'amorçage (*bootstrap*) s'exécutant dans la mémoire morte d'un ordinateur qui télécharge des données provenant d'un serveur. Si la totalité des données tient dans un seul datagramme, l'émetteur peut mettre à 1 le flag DF. On peut aussi le faire lorsque l'émetteur souhaite éliminer le temps d'attente nécessaire au réassemblage des datagrammes au niveau du destinataire. Si le flag DF est mis à 0, cela indique qu'un routeur, ou un hôte, est libre de fragmenter le datagramme IP.

Si le flag MF (*More Fragments*) est mis à 1, le destinataire est informé que d'autres fragments vont arriver. Un MF à 0 indique qu'il s'agit du dernier fragment. Dans le cas de datagrammes IP complets, le flag MF sera toujours à 0.



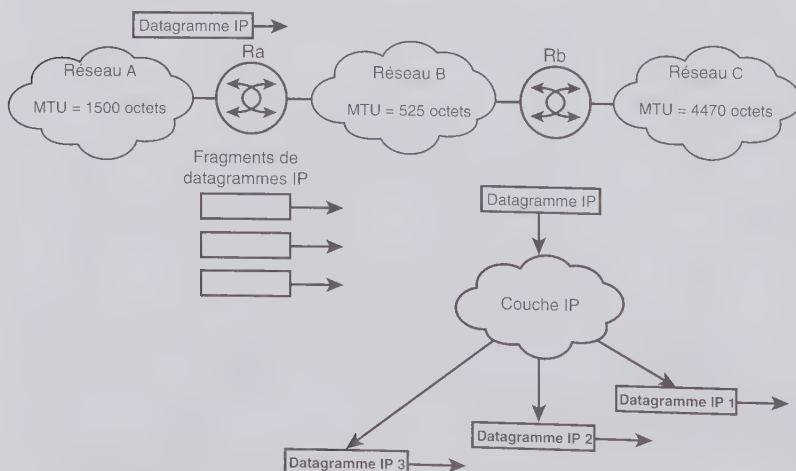
Dans l'idéal, tous les réseaux qu'un datagramme IP traversera doivent avoir une taille de MTU plus grande que le plus grand datagramme IP qui sera transmis par un émetteur. Dans un tel cas de figure, la fragmentation ne sera jamais nécessaire. Cependant, même si vous connaissez la plus petite taille de MTU dans un chemin donné, il est fort possible que des datagrammes successifs suivent différents chemins et parcourent des réseaux pour lesquels la taille de MTU est encore moindre.

Comment est fragmenté un datagramme IP ? Prenons l'exemple de la Figure 6.8, où un hôte sur le réseau A envoie des datagrammes IP à un hôte du réseau C. Les datagrammes doivent transiter par un réseau intermédiaire B. Les tailles de MTU des réseaux A, B et C sont respectivement de 1 500, 525 et 4 470 octets. Le routeur Ra relie les réseaux A et B, et le routeur Rb les réseaux B et C. En général, l'hôte A enverra des datagrammes IP dont la taille correspond à la taille de MTU du réseau A. Dans l'exemple, on suppose que l'hôte A envoie un datagramme IP dont le champ d'Identification vaut 5 et dont la taille est de 1 500 octets. Lorsque le routeur Ra rencontre ce datagramme, il le fragmentera avant de le transmettre au réseau B, car la taille de MTU du réseau B est inférieure à la taille du datagramme. La taille du fragment est choisie de manière qu'elle corresponde à celle de MTU du réseau B (525 octets). Le *décalage*, c'est-à-dire la position du début du champ de données par rapport aux données non fragmentées originelles, est placé dans le champ Fragment Offset de chaque fragment.

Le champ Fragment Offset indique la position des données du fragment par rapport au début du datagramme originel (nous reviendrons sur ce point un peu plus loin). Ce champ fait 13 bits de long. Si vous examinez le format de l'en-tête IP (voir Figure 6.5), vous remarquerez que seuls 13 bits sont utilisés, les autres bits servant pour les flags DF et MF.

Figure 6.8

Un exemple de fragmentation.



Comme le datagramme IP peut faire jusqu'à 65 535 octets, ce champ n'est pas assez long pour décrire des décalages supérieurs à 8 192 octets. C'est la raison pour laquelle le champ Fragment Offset est étendu en ajoutant trois bits à 000 :

valeur_fragment_offset000

La valeur finale de décalage est donc toujours un multiple de 8 octets, et elle décrit toujours des fragments par groupe de 8 octets. Cette valeur doit être multipliée par 8 afin que soit obtenu le décalage réel dans le fragment. Cela signifie que tous les fragments (à l'exception du tout dernier) doivent avoir une taille multiple de 8 octets. De plus, ils doivent tenir dans la taille de MTU du réseau. Dans l'exemple du datagramme IP de 1 500 octets de long, vous pouvez le diviser en plusieurs fragments de 525 octets de long :

Taille fragment 1	= 525 octets
Taille fragment 2	= 525 octets
Taille fragment 3	= 450 octets
Taille datagramme IP	= 1 500 octets

Cependant, 525 n'est pas un multiple de 8. Vous devez aussi prendre en compte la taille de l'en-tête IP (20 octets au minimum), le total devant être inférieur à la taille de MTU. Avec une taille de MTU de 525, il reste 525 - 20 (taille de l'en-tête), c'est-à-dire 505 octets pour les données. Ce nombre n'est pas non plus un multiple de 8. La taille inférieure divisible par 8 est 504. On peut donc choisir une taille de fragment de 504 octets. La Figure 6.9 montre le morcellement du datagramme IP en plusieurs fragments :

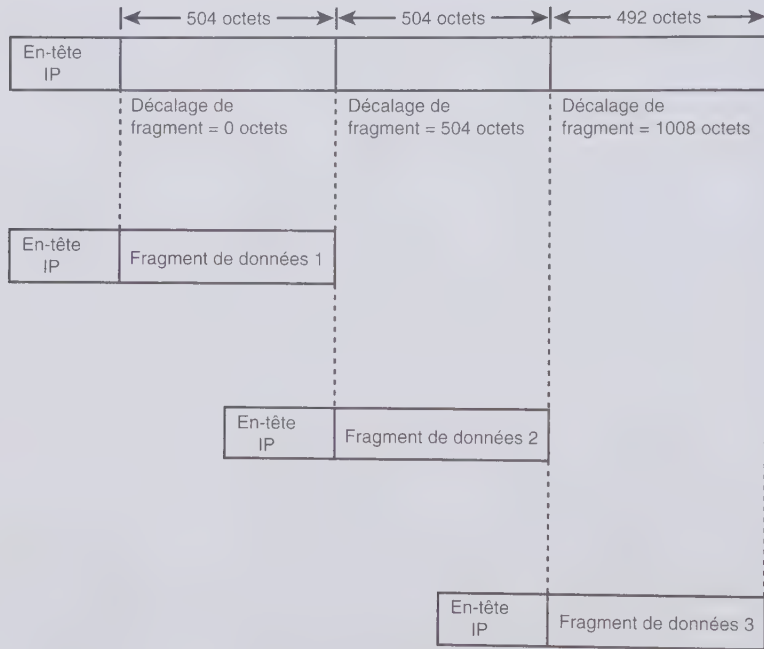
Fragment 1	= 504 octets
Fragment 2	= 504 octets
Fragment 3	= 492 octets
Datagramme IP	= 1 500 octets

Chaque fragment doit avoir son propre en-tête IP. La valeur du champ Identification des datagrammes fragmentés est celle du datagramme original. Cela indique à quel datagramme appartient le fragment.



Figure 6.9

Fragmentation du datagramme IP original.



Dans le fragment, les flags et les valeurs de décalage sont les suivants :

Fragment 1

Longueur totale = 524 octets

Identification = 5

Flag DF = 0

Flag MF = 1

Fragment Offset = 0

Fragment 2

Longueur totale = 524 octets

Identification = 5

Flag DF = 0

Flag MF = 1

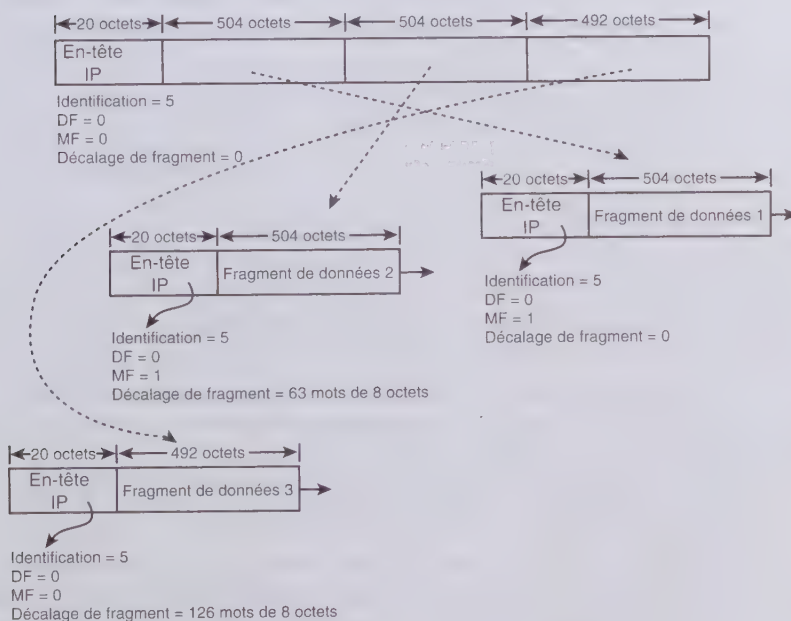
Fragment Offset = 63 mots de 8 octets (504 octets)

Fragment 3

Longueur totale = 512 octets
 Identification = 5
 Flag DF = 0
 Flag MF = 0
 Fragment Offset = 126 mots de 8 octets (1 008 octets)

Figure 6.10

Fragments d'un datagramme IP.



Pour que le destinataire puisse réassembler le datagramme IP, il doit obtenir tous les fragments, du fragment dont la valeur de décalage est 0 jusqu'au fragment de plus haute valeur de décalage. Le destinataire utilise la taille de chaque fragment (champ Total Length de l'en-tête), la valeur de décalage de fragment et le flag MF pour déterminer s'il a reçu tous les fragments. Comment sait-il s'il a reçu le dernier fragment ? Ce fragment a un flag MF valant 0. Le premier fragment est reconnaissable par un flag MF à 1 et un décalage de 0. Tous les fragments intermédiaires auront un flag MF à 1 et une valeur de décalage non nulle.



Une fois qu'un datagramme IP a été fragmenté, les différents morceaux cheminent jusqu'à destination en empruntant des chemins éventuellement distincts. Les fragments sont réassemblés par le destinataire, jamais par un routeur intermédiaire. Les conséquences sont les suivantes :

- Les fragments IP cheminent de leur lieu de fragmentation jusqu'à leur destination.
- Même si des réseaux intermédiaires ont une taille de MTU supérieure à celle du fragment, ce dernier restera de petite taille, ce qui rend le protocole peu efficace.
- Une fragmentation excessive entraîne un trafic de réseau excessif.

Si un fragment est perdu, le datagramme originel ne peut être réassemblé. Même si les autres fragments ont été transmis avec succès, le datagramme originel doit être écarté. Dans ce cas, des protocoles de couches supérieures tels que TCP, assurent que la retransmission de datagramme originel est bien effectuée, mais ce datagramme sera de nouveau fragmenté si sa taille dépasse la taille de MTU de réseaux intermédiaires.

Le nœud IP de destination lance un temporisateur de réassemblage à la réception du premier fragment IP. Si ce temporisateur expire avant réception de tous les autres fragments, le nœud IP écarte les fragments restants, même si ces derniers ont été reçus correctement. Plus le nombre de fragments est important, plus l'éventualité de perdre un datagramme IP est importante. Lorsque le temporisateur de réassemblage expire, un message de Internet Control Message Protocol (ICMP ou "Protocole de message de contrôle Internet") est envoyé à l'émetteur pour indiquer l'expiration du temporisateur. Un périphérique de surveillance, tel qu'un analyseur de protocole, permet d'attirer votre attention sur ce problème.

Le choix de réassembler le datagramme IP au niveau du destinataire a les inconvénients décrits précédemment, mais présente l'avantage de simplifier l'implémentation des routeurs. Les routeurs IP n'ont pas à se soucier du stockage et du réassemblage des fragments IP. Si le chemin du datagramme passait par plusieurs réseaux de petite taille de MTU, le réassemblage des fragments au niveau des routeurs intermédiaires serait une perte de temps, car le datagramme devrait être de nouveau fragmenté.

Le champ TTL (*Time to Live* ou "Durée de vie")

La *durée de vie* (TTL) se mesure en secondes et représente la durée maximale de vie d'un datagramme sur le réseau. Cette valeur est décrémentée à chaque routeur de la durée nécessaire au traitement du paquet. Lorsque le champ TTL tombe à 0, le temporisateur TTL expire, et le datagramme IP est écarté par le routeur (et pas par l'hôte de destination). Les hôtes servant de routeurs doivent appliquer les mêmes principes. Le champ TTL a une double fonction :

- limiter la durée des vie des segments TCP ;
- éliminer les boucles de routage Internet.

Lorsque le temporisateur TTL expire, un message ICMP en avertit la source.

Bien que la valeur de TTL se mesure en secondes, il est difficile pour les routeurs d'estimer le temps exact de transmission des datagrammes IP sur un segment de réseau. Les routeurs pourraient estimer le temps que passe un datagramme IP en son sein, en notant les heures d'arrivée et de départ du datagramme. Cela nécessiterait un traitement supplémentaire, et les routeurs préfèrent ne pas tenter d'évaluation du temps réellement passé : ils se contentent de diminuer de 1 la valeur de TTL. En ce sens, TTL s'apparente à un compteur de hops ("sauts"). Certaines anciennes implémentations commettaient l'erreur d'affecter 16 à TTL, car 16 représente l'infini pour le protocole RIP (*Routing Information Protocol*). TTL est cependant indépendant de la métrique de RIP.

Quelques autres remarques sur les champs TTL :

- Un hôte ne doit pas envoyer un datagramme dont la TTL vaut 0, et un hôte ne doit pas écarter un datagramme simplement parce qu'il a été reçu avec une valeur de TTL inférieure à 2.
- Un protocole de couche supérieure peut définir la valeur de TTL pour implémenter une recherche par propagation d'une ressource Internet donnée. Cette technique est utilisée par certains outils de diagnostic et permet par exemple de localiser le serveur le plus "proche" d'une classe donnée, grâce à la multidiffusion (multicast). Un protocole de transport particulier peut aussi spécifier sa propre limite de TTL.
- Une valeur fixée doit être assez grande pour que le paquet puisse transiter le long du *diamètre* de l'Internet, c'est-à-dire le plus long chemin possible. Une valeur raisonnable est le double du diamètre, ce qui permet de prendre en compte le développement de l'Internet.
- La couche IP doit fournir un moyen pour la couche transport d'agir sur le champ TTL de tout datagramme envoyé. Lorsqu'une valeur de TTL fixée est utilisée, cette valeur doit être configurable. Malheureusement, la plupart des implémentations ne permettent pas de définir la valeur de TTL initiale. Une valeur par défaut de 32 ou de 64 est courante.

Le champ Protocol

Le champ Protocol indique quel protocole de couche supérieure recevra les données IP. Ce champ est utilisé pour le multiplexage/démultiplexage des données vers des protocoles de couche supérieure (voir Chapitre 2). Par exemple, la valeur du champ Protocol est 6 pour TCP, 17 pour UDP et 1 pour ICMP. Lorsque IP voit la valeur 6 dans ce champ, il sait que le datagramme IP encapsule des



données TCP qui doivent être remises au module TCP. Si IP voit une valeur 17, c'est le module UDP qui est concerné. Et ainsi de suite.

Le Tableau 6.5 montre les valeurs du champ Protocol pour IP.

Tableau 6.5 : ID de protocoles

ID de protocole	Abréviation	Signification
0		(Réservé)
1	ICMP	Internet Control Message ("Message de contrôle Internet")
2	IGMP	Internet Group Management ("Gestion de groupe Internet")
3	GGP	Gateway-to-Gateway
4	IP	IP in IP (encapsulation)
5	ST	Stream (flux)
6	TCP	Transmission Control ("Contrôle de transmission")
7	UCL	UCL
8	EGP	Exterior Gateway Protocol ("Protocole de passerelle extérieure")
9	IGP	N'importe quelle passerelle intérieure privée
10	BBN-RCC-MON	Surveillance BBN RCC
11	NVP-II	Network Voice Protocol ("Protocole vocal de réseau")
12	PUP	PUP
13	ARGUS	ARGUS
14	EMCON	EMCON
15	XNET	Débogueur Cross Net
16	CHAOS	Chaos
17	UDP	Datagramme utilisateur
18	MUX	Multiplexage
19	DCN-MEAS	Sous-systèmes de mesure DCN
20	HMP	Host Monitoring
21	PRM	Packet Radio Measurement

Tableau 6.5 : ID de protocoles (suite)

ID de protocole	Abréviation	Signification
22	XNS-IDP	XEROX NS IDP
23	TRUNK-1	Trunk-1
24	TRUNK-2	Trunk-2
25	LEAF-1	Leaf-1
26	LEAF-2	Leaf-2
27	RDP	Reliable Data Protocol ("Protocole de données fiable")
28	IRTP	Internet Reliable Transaction ("Transaction Internet fiable")
29	ISO-TP4	ISO Transport Protocol Class 4 ("Protocole de transport ISO de classe 4")
30	NETBLT	Bulk Data Transfer Protocol
31	MFE-NSP	MFE Network Services Protocol ("Protocole de services de réseau MFE")
32	MERIT-INP	MERIT Internodal Protocol (Protocole internodal MERIT)
33	SEP	Sequential Exchange Protocol (Protocole d'échange séquentiel)
34	3PC	Third Party Connect Protocol
35	IDPR	Inter-Domain Policy Routing Protocol
36	XTP	XTP
37	DDP	Datagram Delivery Protocol (Protocole de remise de datagrammes)
38	IDPR-CMTP	IDPR Control Message Transport Protocol (Protocole de transport de message de contrôle IDPR)
39	TP++	TP++ Transport Protocol (Protocole de transport TP++)
40	IL	IL Transport Protocol (Protocole de transport IL)



Tableau 6.5 : ID de protocoles (suite)

ID de protocole	Abréviation	Signification
41	SIP	Simple Internet Protocol (Protocole Internet simple)
42	SDRP	Source Demand Routing Protocol
43	SIP-SR	SIP Source Route
44	SIP-FRAG	SIP Fragment
45	IDRP	Inter-Domain Routing Protocol
46	RSVP	Reservation Protocol
47	GRE	General Routing Encapsulation
48	MHRP	Mobile Host Routing Protocol
49	BNA	BNA
50	SIPP-ESP	SIPP Encap Security Payload
51	SIPP-AH	SIPP Authentication Header
52	I-NLSP	Integrated Net Layer Security TUBA
53	SWIPE	IP with Encryption
54	NHRP	NBMA Next Hop Resolution Protocol
61		(n'importe quel protocole interne d'hôte)
62	CFTP	CFTP
63		(N'importe quel réseau local)
64	SAT-EXPAK	SATNET and Backroom EXPAK
65	KRYPTOLAN	Kryptolan
66	RVD	MIT Remote Virtual Disk Protocol
67	IPPC	Internet Pluribus Packet Core
68		(N'importe quel système de fichiers distribué)
69	SAT-MON	SATNET Monitoring
70	VISA	Protocole VISA
71	IPCV	Internet Packet Core Utility
72	CPNX	Computer Protocol Network Executive
73	CPHB	Computer Protocol Heart Beat
74	WSN	Wang Span Network

Tableau 6.5 : ID de protocoles (*suite*)

ID de protocole	Abréviation	Signification
75	PVP	Packet Video Protocol
76	BR-SAT-MON	Backroom SATNET Monitoring
77	SUN-ND	SUN ND PROTOCOL-Temporary
78	WB-MON	WIDEBAND Monitoring
79	WB-EXPAK	WIDEBAND EXPAK
80	ISO-IP	ISO Internet Protocol
81	VMTP	VMTP
82	SECURE-VMTP	SECURE-VMTP
83	VINES	VINES
84	TTP	TTP
85	NSFNET-IGP	NSFNET-IGP
86	DGP	Dissimilar Gateway Protocol
87	TCF	TCF
88	IGRP	IGRP
89	OSPFIGP	OSPFIGP
90	Sprite-RPC	Sprite RPC Protocol
91	LARP	Locus Address Resolution Protocol
92	MTP	Multicast Transport Protocol
93	AX.25	Trames AX.25
94	IPIP	IP-within-IP Encapsulation Protocol
95	MICP	Mobile Internetworking Control Pro.
96	SCC-SP	Semaphore Communications Sec. Pro.
97	ETHERIP	Ethernet-within-IP Encapsulation
98	ENCAP	Encapsulation Header
99		(N'importe quel mécanisme de chiffrement privé)



Tableau 6.5 : ID de protocoles (*suite*)

ID de protocole	Abréviation	Signification
100	GMTP	GMTP
101–254		(Non affecté)
255		(Réservé)

Le champ Header Checksum ("total de contrôle de l'en-tête")

Le champ Header Checksum ne sert que pour l'en-tête IP. Le complément à un de chaque valeur 16 bits constituant l'en-tête est ajouté (excepté le champ Header Checksum lui-même). On prend alors le complément à 1 de la somme. Ce champ est recalculé à chaque routeur, puisque le champ TTL est décrémenté au niveau du routeur, ce qui modifie l'en-tête IP.

Le calcul de totaux de contrôle est des plus simples, et l'expérience montre qu'il permet de bien détecter les erreurs de transmission du datagramme.

Adresses IP source et destination

Les adresses source et destination sont les adresses sur IP 32 bits des nœuds source et destination. Elles figurent dans chaque datagramme IP, car le réseau IP est un réseau sans connexion, et IP ne sait donc pas à priori d'où viennent et où vont les datagrammes.

Les routeurs utilisent la valeur d'adresse de destination pour effectuer le routage de chaque datagramme IP.

• Options IP

IP implémente un mécanisme permettant d'indiquer le niveau de sécurité d'un datagramme, des informations de routage par la source et des informations de marques de temps. Comme ces précisions sont rarement utilisées, on a préféré les implémenter sous forme de champs facultatifs appelés des *options IP*.

Les options IP sont les suivantes :

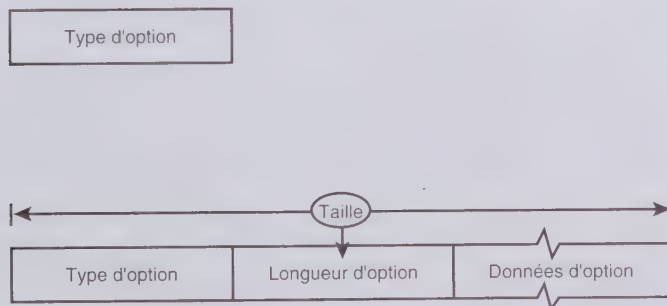
- Sécurité ;
- Record route ;
- Strict source routing ;
- Loose source routing ;
- Internet Timestamp.

Les options IP doivent être prises en compte par tous les nœuds IP et apparaissent au sein des datagrammes IP près de la fin de l'en-tête. Leur présence dans un datagramme est facultative contrairement à leur implémentation. Ainsi, dans un environnement nécessitant un haut niveau de sécurité, l'option Sécurité peut être obligatoire dans tous les datagrammes IP.

Le champ Option est de taille variable, et les valeurs doivent être des multiples de mots de 32 bits. On remplit éventuellement avec des zéros pour que la taille du champ soit bien un multiple de mots de 32 bits. Il existe deux formats pour les valeurs d'options IP (voir Figure 6.11) :

- **Format 1 d'option.** Un seul octet de type d'option.
- **Format 2 d'option.** Un octet de type d'option, un octet de longueur d'option et les octets de données d'option. L'octet de longueur d'option contient la taille totale de l'option, en comptant l'octet de type, l'octet de longueur et les données d'options.

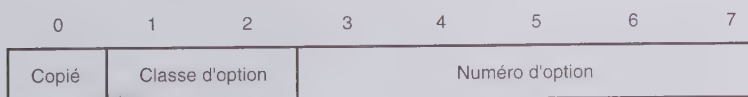
Figure 6.11
Formats d'options.



L'octet de type d'option est le premier octet apparaissant dans toute spécification d'option. Dans le format 1, cet octet est le seul spécifié, et dans le format 2, il est suivi de deux autres champs. Le type d'option est constitué des trois champs suivants (voir Figure 6.12) :

- Flag "copié" (1 bit) ;
- Classe d'option (2 bits) ;
- Numéro d'option (5 bits).

Figure 6.12
Octet d'option.



Le flag "copié" spécifie la façon dont les routeurs traitent les champs d'option lors de la fragmentation. Si ce champ vaut 1, les options seront copiées dans tous les fragments du datagramme IP. S'il vaut 0, les options seront copiées dans le premier fragment de datagramme IP, mais pas dans les suivants.

Le champ Classe d'option fait 2 bits de long et peut contenir une valeur comprise entre 0 et 3. Ces valeurs sont analysées dans le Tableau 6.6.

Tableau 6.6 : Valeurs des champs de classe d'options dans l'octet Type d'option

Code	Signification
0	Contrôle réseau
1	Réservé pour un usage ultérieur
2	Débogage et mesures
3	Réservé pour un usage ultérieur

Le champ de classe détermine la catégorie à laquelle appartient l'option. Ces catégories sont : Contrôle réseau et Débogage. Le champ Numéro d'option fait 5 bits de long et permet de spécifier jusqu'à 32 options pour une catégorie d'option donnée. Le Tableau 6.7 dresse la liste des différentes options déjà définies.

Tableau 6.7 : Options IP

Classe d'option	Numéro d'option	Longueur	Signification
0	0	-	Fin de la liste d'options. Cette option n'occupe qu'un octet et ne possède pas d'octet de longueur. On l'utilise si les options ne se terminent pas à la fin de l'en-tête.
0	1	-	Pas d'opération. Cette option n'occupe qu'un octet et ne possède pas d'octet de longueur. Permet d'aligner les octets dans une liste d'options.
0	2	11	Sécurité de base. Contient les options de Sécurité, de Compartimentation, de Groupe utilisateur (TCC) et les codes de restriction de traitement compatibles avec les exigences du ministère américain de la Défense.
0	3	variable	Loose Source Routing. Permet de router le datagramme IP selon les informations fournies par la source.

Tableau 6.7 : Options IP (*suite*)

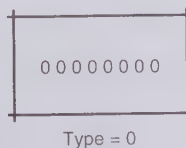
Classe d'option	Numéro d'option	Longueur	Signification
0	5	variable	Sécurité étendue. Spécifie des options de sécurité supplémentaires.
0	7	variable	Record route. Permet d'enregistrer le chemin suivi par un datagramme IP.
0	8	4	ID de flux. Permet de transporter l'identificateur de flux SATNET 16 bits dans des réseaux ne prenant pas en charge le concept de flux. Ce champ est maintenant obsolète.
0	9	variable	Strict Source Routing. Permet de router le datagramme IP selon les informations fournies par la source.
2	4	variable	Internet Timestamp. Permet d'enregistrer des marques de temps au long du chemin. Pour l'instant, il s'agit de la seule option définie pour la catégorie de débogage et de mesure.

Les options Fin de la liste d'options et Pas d'opération

Les options Fin de la liste d'options et Pas d'opération sont des options à un seul octet. L'option Fin de la liste d'options (voir Tableau 6.7) est composée de bits à zéro (voir Figure 6.13).

Figure 6.13

L'option Fin de la liste d'options.

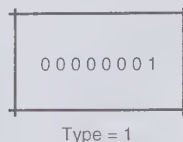


Cette option indique la fin de la liste d'options, qui ne coïncide pas forcément avec la fin de l'en-tête spécifiée par le champ IHL. On utilise cette option après toutes les options, et pas après chaque option. Elle n'est utile que si la fin des options ne s'aligne pas sur 32 bits.

L'option Pas d'opération (voir Tableau 6.7) est une option à un octet, constituée du motif de bits suivant (voir Figure 6.14) :

Figure 6.14

L'option Pas d'opération.



L'option Pas d'opération peut être utilisée entre deux options pour aligner le début de l'option suivante sur 32 bits.

L'option Sécurité

L'option Sécurité permet aux hôtes de spécifier des paramètres de sécurité, de compartimentation, de restriction de traitement et de groupes utilisateurs fermés. On compte deux types d'options de sécurité :

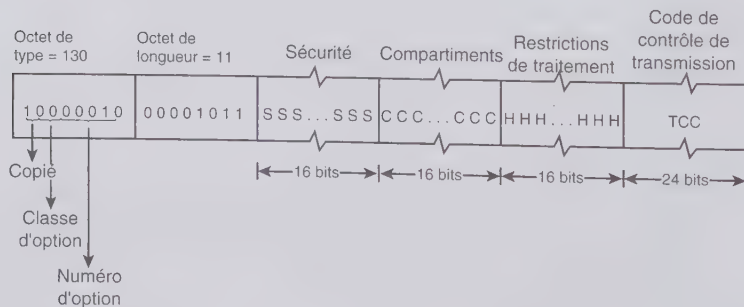
- **Sécurité de base.** Indique le niveau de sécurité : Confidentiel, Non classé, Secret, etc.
- **Sécurité étendue.** Spécifie des mesures supplémentaires de sécurité, en fonction des besoins propres à une organisation militaire.

Option de Sécurité de base

La Figure 6.15 montre le format d'option de Sécurité de base. L'octet de type correspondant vaut 130. Les codes définissent les autorités dont les règles influent sur le traitement du datagramme IP. Parmi ces autorités (américaines) on peut citer le NSA, la CIA et le DOE (*Department of Energy*).

Figure 6.15

Format de l'option Sécurité de base.



Le champ de Sécurité fait 16 bits et spécifie un niveau de sécurité parmi 16 autres, mais 8 sont encore réservés pour un usage ultérieur. Le Tableau 6.8 dresse la liste des différentes valeurs. La signification de certaines valeurs dépend de l'autorité employant l'option.

Tableau 6.8 : Valeurs des codes de sécurité

Code	Signification
00000000 00000000	Non classé
11110001 00110101	Confidentiel
01111000 10011010	EFTO
10111100 01001101	MMMM
01011110 00100110	PROG
10101111 00010011	Restricted
11010111 10001000	Secret
01101011 11000101	Top Secret
00110101 11100010	Réservé pour un usage ultérieur
10011010 11110001	Réservé pour un usage ultérieur
01001101 01111000	Réservé pour un usage ultérieur
00100100 10111101	Réservé pour un usage ultérieur
00010011 01011110	Réservé pour un usage ultérieur
10001001 10101111	Réservé pour un usage ultérieur
11000100 11010110	Réservé pour un usage ultérieur
11100010 01101011	Réservé pour un usage ultérieur

Le champ Compartiments fait 16 bits de long et vaut zéro lorsque l'information transmise n'est pas compartimentée. D'autres valeurs peuvent être obtenues en s'adressant à la Defense Intelligence Agency américaine.

Le champ Handling Restrictions de traitement fait 16 bits de long. Les valeurs représentent des marques de contrôle et de version et sont des digraphes alphanumériques définis dans le *Defense Intelligence Agency Manual* (DIAM) 65-19, "Standard Security Markings".

Le champ Code de contrôle de transmission (TCC) fait 24 bits de long et permet de discriminer le trafic et de définir des communautés d'intérêt contrôlées, formées d'abonnés. Les valeurs de TCC sont des trigraphes, disponibles auprès de la *Defense Communication Agency* (DCA) américaine.



Option Sécurité étendue

L'option sécurité étendue permet de spécifier des mesures de sécurité supplémentaires (voir Figure 6.16). Elle est caractérisée par un octet de type de valeur 133.

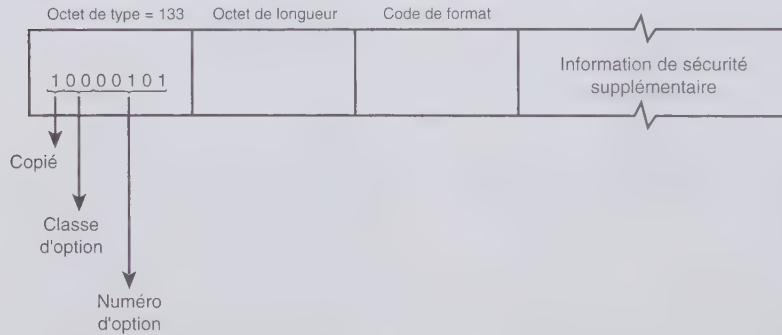
Les routeurs commerciaux prennent rarement en charge les options de sécurité IP.

Option Record Route

L'option Record Route est utilisée par l'émetteur pour enregistrer les adresses IP des routeurs transmettant le datagramme IP jusqu'à destination. La liste des routeurs effectivement rencontrés est enregistrée dans le datagramme.

Figure 6.16

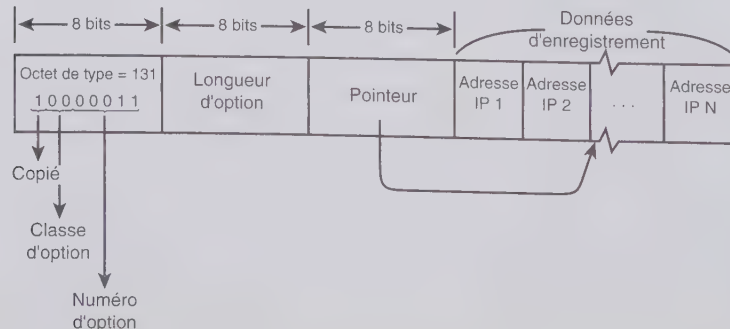
Option Sécurité étendue.



L'option Record Route commence par le code de type d'option de valeur 131 (voir Figure 6.17). Le deuxième octet correspond à la longueur d'option. Le champ Record Data (Données d'enregistrement) contient des emplacements pour une liste d'adresses IP. Le troisième octet est un pointeur vers le champ Route Data et indique le début de l'emplacement suivant à traiter. Le pointeur est relatif à l'option même, et sa valeur minimale est 4. Si la valeur du pointeur est supérieure

Figure 6.17

Option Record Route.



à celle de l'octet de longueur, le champ Record Data ne contient pas d'emplacements vides, autrement dit, le champ Record Data est alors plein. Dans ce cas, le routeur transmet le datagramme sans insérer sa propre adresse dans la liste.

Si le pointeur est inférieur à la valeur de l'octet de longueur, la liste Record Data n'est pas pleine. Le routeur insère sa propre adresse IP à la position spécifiée par le champ Pointeur et incrémente la valeur de ce dernier par 4, c'est-à-dire par la taille de l'adresse IP. L'adresse enregistrée est l'adresse IP de l'interface de réseau du routeur par lequel transite le datagramme.

L'émetteur et la destination doivent tout deux se mettre d'accord pour que les informations de routages soient indiquées et traitées. L'émetteur ajoute l'option Record Route en prévoyant assez d'emplacements pour les adresses IP enregistrées. La destination accepte de traiter la liste des adresses IP inscrites dans le champ Record Data. Si la destination n'accepte pas de traiter les données, ces informations sont tout simplement ignorées.

Options Strict Source Routing et Loose Source Routing

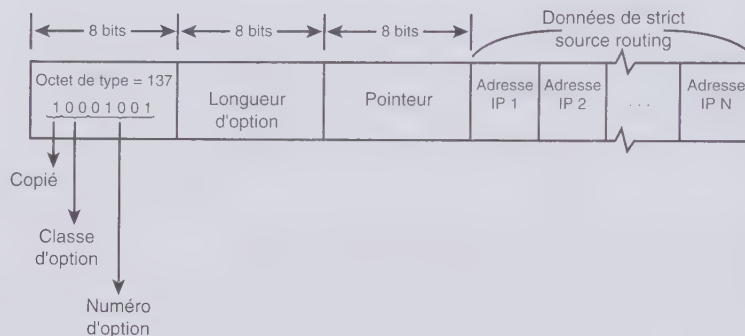
Dans le cas du routage par la source, l'émetteur fournit le chemin jusqu'à la destination. Le chemin est constitué d'une liste d'adresses IP de routeurs devant être traversés. En temps normal, le chemin jusqu'à destination est décidé par les routeurs qui se basent sur le chemin optimal tel qu'il est estimé par les protocoles de routage. Il en va différemment dans le cas du routage par la source. Cette technique est utile pour tester un chemin particulier qui ne serait pas autrement sélectionné par les routeurs. Elle permet aussi de router arbitrairement des datagrammes dans des réseaux que l'on sait fiables. L'inconvénient du source routing est qu'il rend nécessaire la connaissance de la topologie du réseau et de la liste des routeurs utilisés pour la transmission d'un datagramme. Dans un réseau complexe, la topologie et le chemin suivi par un datagramme IP sont rarement faciles à déterminer.

Il existe deux types de routage par la source :

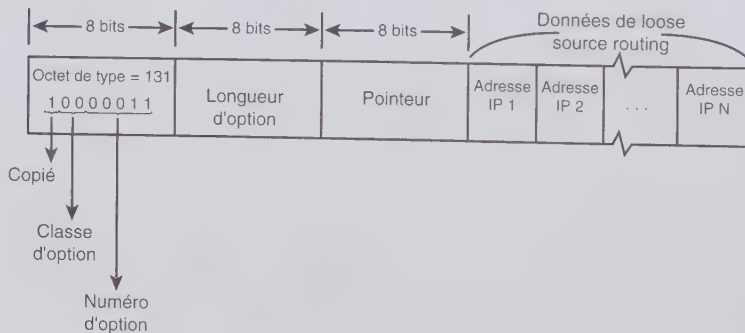
- Strict Source Routing ;
- Loose Source Routing.

Dans le cas de l'option Strict Source Routing (voir Figure 6.18), l'émetteur inclut une séquence d'adresses IP qui doivent être parcourues exactement dans cet ordre, lors de la transmission du datagramme. Le chemin entre deux adresses IP successives dans la liste ne peut être qu'un segment physique de réseau. Un segment physique de réseau ne peut contenir que des périphériques de couche 1 (répéteurs) et 2 (ponts), mais pas des périphériques de couche 3 (routeurs). Si les routeurs ne peuvent suivre exactement la séquence d'adresses IP, le datagramme est rejeté.



Figure 6.18*L'option Strict Source Routing.*

Dans le cas de l'option Loose Source Routing (voir Figure 6.19), l'émetteur inclut aussi une séquence d'adresses IP à parcourir lors de la transmission du datagramme. Toutefois, dans ce cas, le chemin entre deux adresses IP successives dans la liste peut passer par un nombre quelconque de routeurs. La Figure 6.20 compare Strict Source Routing et Loose Source Routing.

Figure 6.19*L'option Loose Source Routing.*

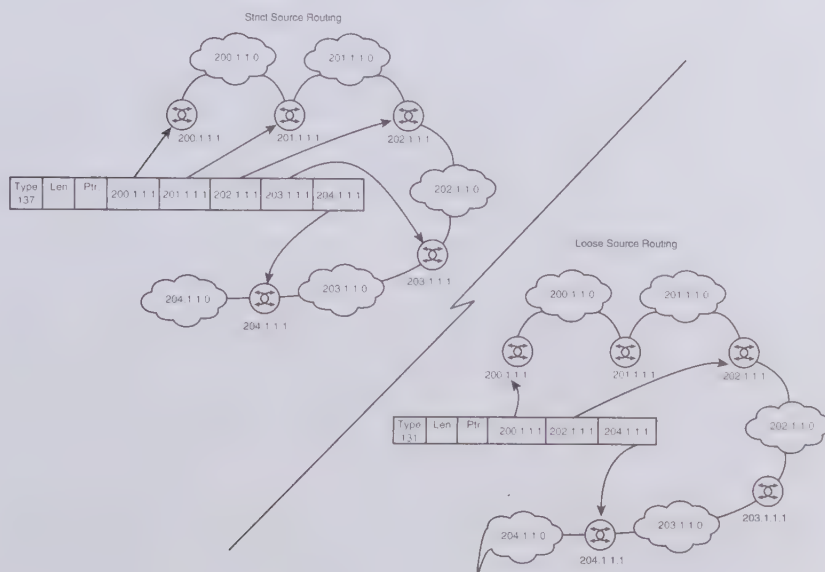
Dans les deux options, les routeurs écrasent l'adresse source avec leurs adresses IP locales. Lorsqu'un datagramme arrive à destination, la liste d'adresses IP contient la route qui a été suivie. Cette liste est identique à celle qui est produite par l'option Record Route. Pour cette raison, l'option Loose Source Routing est aussi appelée l'option Loose Source and Record Route (LSRR) et l'option Strict Source Routing Option, l'option Strict Source and Record Route (SSRR).

En examinant les formats des options Record Route, Strict Source Routing et Loose Source Routing qui apparaissent dans les Figures 6.17, 6.18 et 6.19, on remarque une grande similarité entre ces divers formats. Le champ Pointeur tient

lieu d'indice pour la liste d'adresses IP. Le champ Longueur indique le nombre d'emplacements pour les adresses IP. Lorsqu'un routeur a fini de traiter une adresse IP, il avance l'indice Pointeur jusqu'au prochain emplacement d'adresse IP disponible. Comme les adresses IP font 4 octets de long, la valeur du champ Pointeur est incrémentée de 4 pour passer à l'emplacement suivant.

Figure 6.20

*Comparaison entre
Strict Source Routing
et Loose Source Routing.*



Lorsque la valeur du champ Pointeur est plus grande que celle du champ Longueur, cela signifie que la liste d'adresses IP a été traitée ou qu'elle est épuisée. Le datagramme IP est alors transmis de manière standard en fonction de la table de routage du routeur, sans que ce dernier se réfère aux informations figurant dans les champs d'options IP. Si la valeur de Pointeur est inférieure à celle du champ Longueur, la liste d'adresses IP n'a pas été entièrement traitée (voir Figure 6.21). Dans ce cas, le routeur utilise l'indice Pointeur pour obtenir l'adresse du prochain routeur vers lequel transmettre le datagramme. Le routeur place ensuite l'adresse IP de son interface de réseau dans l'emplacement d'adresses IP dont Pointeur est l'indice, cela afin d'enregistrer la route qui a été prise. On modifie ensuite l'indice Pointeur – en l'incrémentant de 4 – pour indiquer le prochain emplacement d'adresses IP.

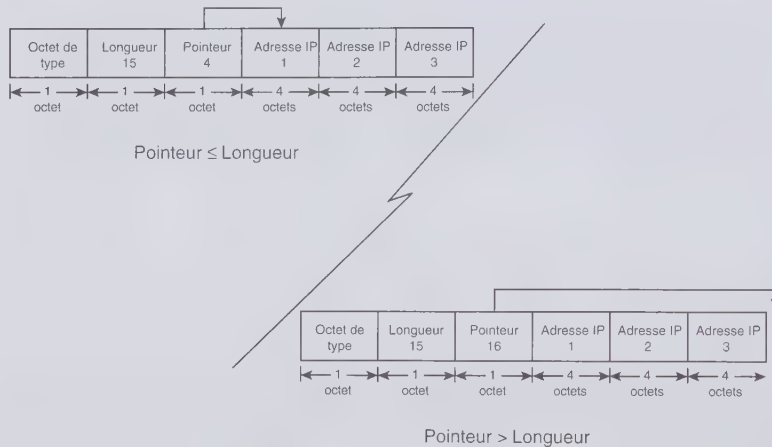
© 2000



LE MACMILLAN

Figure 6.21

Utilisation des champs Pointeur et Longueur.



Le champ Route Data est constitué d'une liste d'adresses IP. Chaque adresse IP fait 4 octets de long. Si la valeur de Pointeur est supérieure à celle du champ Longueur, la route source est vide et la route enregistrée est pleine. Le routage se fonde alors sur le champ d'adresse de destination du datagramme IP.

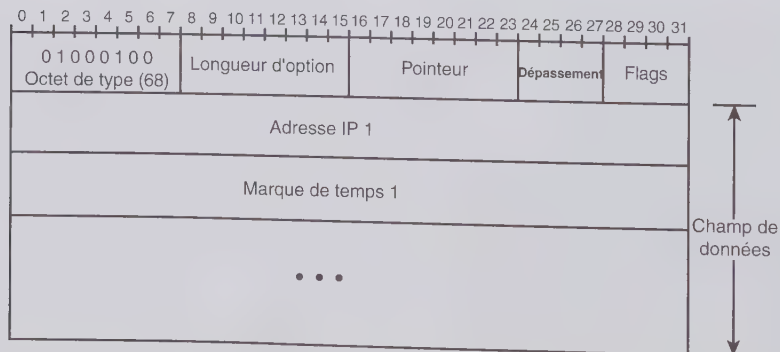
L'option Internet Timestamp

L'option Internet Timestamp permet d'enregistrer les marques de temps correspondant à l'heure de réception du datagramme IP pour chaque routeur. L'adresse IP du routeur et l'heure de réception du datagramme sont enregistrées. Les marques de temps sont mesurées comme le nombre de millisecondes s'étant écoulées depuis minuit UTC (ex GMT), c'est-à-dire depuis minuit à 0° de longitude.

En sus des marques de temps, cette option permet d'enregistrer les adresses IP des routeurs. La Figure 6.22 montre le format de l'option Internet Timestamp.

Figure 6.22

L'option Internet Timestamp.



Le type d'option vaut 68, et Longueur d'option est le nombre d'octets dans l'option, en comptant tous les champs. Chaque entrée du champ Données de l'option Timestamp contient deux valeurs de 32 bits : l'adresse IP du routeur et la marque de temps qu'il a enregistrée.

Le champ Longueur d'option spécifie l'espace réservé dans le champ Données pour les adresses IP et les marques de temps. Le champ Pointeur représente un indice dans le champ de données, pointant vers le prochain emplacement non utilisé. Cet indice est mesuré à partir du début de l'option. La plus petite valeur valide du champ Pointeur est 5. Lorsque la valeur du Pointeur est supérieure à la valeur du champ Longueur d'option, cela signifie que la zone de marque de temps est pleine. L'émetteur doit créer l'option Timestamp avec assez d'espace pour les données, afin que toutes les marques de temps souhaitées puissent y prendre place. La taille de cette option ne varie pas lors du transit du datagramme IP. L'émetteur doit initialiser la zone de données en la remplissant de zéros.

Le champ Dépassement fait 4 bits de long et contient le nombre de routeurs qui n'ont pu indiquer la marque de temps, car le champ de données s'est révélé trop petit. La valeur maximale du champ Dépassement est 15 (routeurs). S'il n'y a pas assez d'espace pour insérer une marque du temps complète dans la zone de données, ou si le compteur de dépassements dépasse 15, on considère que le datagramme IP est en erreur, et on l'écarte. Dans les deux cas, un message ICMP de problème de paramètres est envoyé à l'émetteur initial.

Le champ Flags détermine le format des informations du champ Données. Il peut prendre trois valeurs, dont les codes sont décrits dans le Tableau 6.9.

Tableau 6.9 : Valeur de flag pour l'option Timestamp

Valeur de flag	Signification
0	Seuls les marques de temps sont enregistrées. Elles sont stockées dans des mots de 32 bits consécutifs. Les adresses IP ne sont pas indiquées.
1	Chaque marque de temps est précédée par l'adresse IP de l'entité qui enregistre (le routeur). Ce format est reproduit dans la Figure 6.22.
3	Les champs d'adresses IP sont spécifiés par l'émetteur. Un module IP (routeur) n'enregistre sa marque du temps que si son adresse IP correspond à l'adresse IP spécifiée qui suit.

Les valeurs marques de temps sont justifiées à droite, avec des valeurs de 32 bits, exprimées en millisecondes, depuis minuit UTC. Si l'heure n'est pas disponible en millisecondes ou ne peut être exprimée relativement à minuit UTC, n'importe



quelle valeur d'heure, l'heure locale par exemple, peut être insérée dans le champ Marque de temps, pour peu que le bit de poids fort du champ soit mis à 1. Ce bit de poids fort à 1 indique qu'on utilise une valeur non standard.

Les routeurs peuvent être configurés à l'heure locale et leurs horloges peuvent ne pas être configurées. Du fait des différences dans les valeurs données par les horloges des routeurs, les valeurs de marques de temps doivent être considérées comme des approximations.

L'option Timestamp n'est pas copiée en cas de fragmentation, et elle n'est transmise que dans le premier fragment du datagramme IP.

• Ordre réseau des octets

La couche physique transmet les bits dans l'ordre de leur réception en provenance des couches supérieures. Les bits qui constituent les données transmises peuvent être stockés de différente manière selon les ordinateurs. Ainsi, tous les ordinateurs ne stockent pas les valeurs entières à 32 bits (telles que les adresses IP) de la même façon. Certains stockent les octets de poids faible de l'entier à 32 bits dans des adresses mémoire plus basses. Cette méthode est dite *petit-boutiste* (*little-endian*). L'autre méthode est la méthode *gros-boutiste* (*big-endian*), selon laquelle les octets de poids forts sont stockés dans des adresses mémoire plus basses.

Certaines machines, les ordinateurs 16 bits les plus anciens par exemple, stockent les données par unités de 16 bits, avec les octets de poids faible stockés dans les adresses mémoire plus basses, mais avec les octets inversés dans le mot de 16 bits.

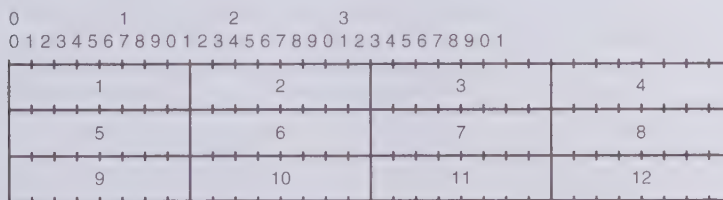
Du fait des différences de stockage des octets de données, vous ne pouvez directement copier les données d'un nœud IP à un autre. Les protocoles TCP/IP définissent un standard d'ordre réseau des octets (NSBO en anglais), utilisé pour le stockage des champs binaires dans les paquets IP. Tous les nœuds IP doivent représenter les données des paquets à l'aide de ce standard d'ordre réseau des octets. Avant d'envoyer un paquet, le nœud IP convertit sa représentation locale des données en une représentation conforme au NSBO. Lors de la réception d'un paquet IP, le nœud IP convertit les données au format NSBO afin d'obtenir la représentation locale des données, propre à ce nœud. Dans la plupart des implémentations de TCP/IP, la conversion entre le NSBO et la représentation propre au nœud est effectuée par les interfaces de programmation et les routines d'assistance de l'application TCP/IP.

Le network standard byte order (NSBO) spécifie que les entiers de l'octet le plus significatif sont envoyés en premier (méthode gros-boutiste). Autrement dit, si vous examinez l'en-tête d'un paquet en transit, les valeurs entières d'octet le plus significatif se trouveront près du début du paquet, et ceux d'octet le moins significatif, vers la fin.

Les en-têtes de protocole TCP/IP sont souvent formatés sous forme de mots de 32 bits (voir Figure 6.5). L'ordre de transmission des octets dans l'en-tête est celui de leur lecture normale (de gauche à droite). Ainsi, dans la Figure 6.23, les octets sont transmis dans l'ordre de leur numérotation.

Figure 6.23

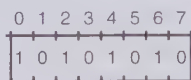
Ordre de transmission des octets.



Lorsqu'un octet représente une valeur numérique, le bit le plus à gauche est le bit de poids fort – ou bit le plus significatif. Autrement dit, le bit de numéro 0 est le plus significatif. Ainsi, la Figure 6.24 représente la valeur 170 (décimal).

Figure 6.24

Signification des bits.



Trace IP

Les sections qui suivent vous proposent d'étudier des décodages de paquets IP capturés sur un vrai réseau. Il convient de suivre cette partie après avoir bien lu les sections précédentes qui décrivaient de façon théorique le format et la signification des champs IP. La Figure 6.25 montre la trace de protocole des cinquante premiers paquets capturés lors d'une session Telnet. Comme ce chapitre ne traite que du protocole IP lui-même, nous ne nous attarderons que sur certains paquets de cette trace.

Vous pouvez remarquer que les trois premiers paquets de cette trace sont des paquets ARP. Le rôle d'ARP dans la résolution des adresses matériel est détaillé dans le Chapitre 5.

L'en-tête Ethernet de tous les datagrammes IP contient une valeur EtherType de 800 (hexadécimal). Cette valeur indique que la trame Ethernet encapsule un datagramme IP.



Figure 6.25

Trace Telnet.

No. Source	Destination	Layer	Size	Summary
1 0000C024282D	FFFFFFFFFFFF	arp	0064	Req by 144.19.74.44 for 144.19.74
2 0000C024282D	FFFFFFFFFFFF	arp	0064	Req by 144.19.74.44 for 144.19.74
3 0000C0A20F8E	0000C024282D	arp	0064	Reply 144.19.74.201=0000C0A20F8E
4 0000C024282D	0000C0A20F8E	tcp	0064	Port:6295 ---> TELNET SYN
5 0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK SYN
6 0000C024282D	0000C0A20F8E	telnt	0069	Cmd=Do; Code=Echo; Cmd=Do; Code=S
7 0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
8 0000C0A20F8E	0000C024282D	telnt	0064	Cmd=Do; Code=;
9 0000C024282D	0000C0A20F8E	telnt	0064	Cmd=Won't; Code=;
10 0000C0A20F8E	0000C024282D	telnt	0067	Cmd=Will; Code=Echo; Cmd=Will; Co
11 0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
12 0000C024282D	0000C0A20F8E	telnt	0072	Cmd=Do; Code=Suppress Go Ahead; C
13 0000C0A20F8E	0000C024282D	telnt	0070	Cmd=Do; Code=Terminal Type; Cmd=D
14 0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
15 0000C024282D	0000C0A20F8E	telnt	0072	Cmd=Subnegotiation Begin; Code=
16 0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
17 0000C0A20F8E	0000C024282D	telnt	0064	Cmd=Subnegotiation Begin; Code=Te
18 0000C024282D	0000C0A20F8E	telnt	0071	Cmd=Subnegotiation Begin; Code=Te
19 0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
20 0000C0A20F8E	0000C024282D	telnt	0067	Cmd=Do; Code=Echo; Cmd=Will; Code
21 0000C024282D	0000C0A20F8E	telnt	0069	Cmd=Won't; Code=Echo; Cmd=Don't;
22 0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
23 0000C0A20F8E	0000C024282D	telnt	0104	Data=..Linux 1.3.50 (ltreec1.ltre
24 0000C024282D	0000C0A20F8E	tcp	0064	Port:6295 ---> TELNET ACK
25 0000C0A20F8E	0000C024282D	telnt	0064	Data=..
26 0000C024282D	0000C0A20F8E	tcp	0064	Port:6295 ---> TELNET ACK
27 0000C0A20F8E	0000C024282D	telnt	0073	Data=ltreec1 login:
28 0000C024282D	0000C0A20F8E	tcp	0064	Port:6295 ---> TELNET ACK
29 0000C024282D	0000C0A20F8E	telnt	0064	Data=u
30 0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
31 0000C0A20F8E	0000C024282D	telnt	0064	Data=u
32 0000C024282D	0000C0A20F8E	tcp	0064	Port:6295 ---> TELNET ACK
33 0000C024282D	0000C0A20F8E	telnt	0064	Data=s
34 0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
35 0000C0A20F8E	0000C024282D	telnt	0064	Data=s
36 0000C024282D	0000C0A20F8E	tcp	0064	Port:6295 ---> TELNET ACK
37 0000C024282D	0000C0A20F8E	telnt	0064	Data=e
38 0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
39 0000C0A20F8E	0000C024282D	telnt	0064	Data=e
40 0000C024282D	0000C0A20F8E	tcp	0064	Port:6295 ---> TELNET ACK
41 0000C024282D	0000C0A20F8E	telnt	0064	Data=r
42 0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
43 0000C0A20F8E	0000C024282D	telnt	0064	Data=r
44 0000C024282D	0000C0A20F8E	tcp	0064	Port:6295 ---> TELNET ACK
45 0000C024282D	0000C0A20F8E	telnt	0064	Data=1
46 0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
47 0000C0A20F8E	0000C024282D	telnt	0064	Data=1
48 0000C024282D	0000C0A20F8E	tcp	0064	Port:6295 ---> TELNET ACK
49 0000C024282D	0000C0A20F8E	telnt	0064	Data=..
50 0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK

Le champ Protocol des en-têtes de tous les datagrammes IP a une valeur 6. Cela indique que le datagramme IP encapsule un paquet TCP.

• **Datagramme IP n° 1 : Paquet IP avec Flags = 0, MF = 0, DF = 0**

La Figure 6.26 montre le premier datagramme IP (paquet n° 4 dans la Figure 6.25) envoyé par la source (adresse 144.19.74.44) à l'hôte Telnet (adresse 144.19.74.201). Les valeurs de paramètres d'en-tête pour ce paquet sont les suivantes :

Version : 4
 Longueur d'en-tête (mots de 32 bits) : 5
 TOS : 0 (préséance : routine, délai normal, débit normal, fiabilité normale)
 Longueur totale : 44
 Identification : 1
 Flags : 0, DF = 0, MF = 0 (fragmentation autorisée, dernier fragment)
 Décalage de fragment : 0
 Time to Live : 100 secondes
 Protocole : 6 (TCP)
 Total de contrôle : 0xA1AF(valide)
 Adresse IP source : 144.19.74.44
 Adresse IP de destination : 144.19.74.201

Le champ version vaut 4, le format du datagramme IP est donc IPv4.

Le champ Header Length fait 5 mots (20 octets). C'est la taille minimale de l'en-tête IP, aucune option n'est donc spécifiée.

La longueur totale de ce datagramme est de 44 octets. L'en-tête Ethernet et le total de contrôle CRC font 18 octets. En les additionnant, on obtient une taille de la trame Ethernet de 62 octets. Deux octets de remplissage sont ajoutés pour amener la taille de la trame Ethernet à 64 octets.

Ce datagramme IP a un champ Identification de valeur 1 et une valeur de champ TTL de cent secondes. Cette valeur de TTL est la valeur par défaut utilisée par l'hôte à 144.19.74.44 pour tous les datagrammes IP qu'il émet.

La valeur de TOS est 0, ce qui signifie que ce datagramme IP doit être traité avec une préséance de routine, un délai normal, un débit normal et une fiabilité normale.

Le flag de fragmentation est à 0, de même que les flags DF et MF. Ce datagramme pourrait être fragmenté, dans ce cas il s'agirait du dernier fragment. La valeur nulle de Fragment Offset indique qu'il s'agit aussi du premier fragment. Puisque



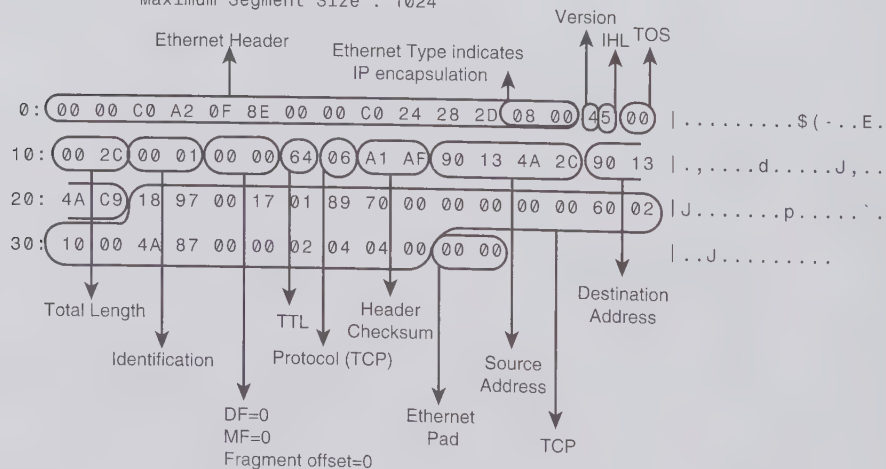
ce datagramme est le premier et le dernier fragment, il s'agit d'un datagramme complet, non fragmenté.

Figure 6.26

Datagramme IP n°1.

```

Packet Number : 4           6:38:38 PM
Length : 64 bytes
ether: ===== Ethernet Datalink Layer =====
      Station: 00-00-C0-24-28-2D ----> 00-00-C0-A2-0F-8E
      Type: 0x0800 (IP)
ip: ===== Internet Protocol =====
      Station: 144.19.74.44 ----> 144.19.74.201
      Protocol: TCP
      Version: 4
      Header Length (32 bit words): 5
      Precedence: Routine
          Normal Delay, Normal Throughput, Normal Reliability
      Total length: 44
      Identification: 1
      Fragmentation allowed, Last fragment
      Fragment Offset: 0
      Time to Live: 100 seconds
      Checksum: 0xA1AF(Valid)
tcp: ===== Transmission Control Protocol =====
      Source Port: 6295
      Destination Port: TELNET
      Sequence Number: 25784320
      Acknowledgement Number: 0
      Data Offset (32-bit words): 6
      Window: 4096
      Control Bits: Synchronize Sequence Numbers (SYN)
      Checksum: 0x4A87(Valid)
      Urgent Pointer: 0
      Option: MAXIMUM SEGMENT SIZE
          Option Length: 4
          Maximum Segment Size : 1024
  
```



Le champ Protocol vaut 6, indiquant que le datagramme IP transporte un message TCP.

La valeur de Checksum est A1AF (hexadécimal). Le total de contrôle est valide.

• **Datagramme IP n° 2 : paquet IP de réponse avec Flags = 0, MF = 0, DF = 0**

La Figure 6.27 montre le deuxième datagramme IP (paquet n° 5 dans la Figure 6.25), envoyé en réponse au datagramme IP présenté précédemment. Ce datagramme est envoyé par l'hôte Telnet d'adresse 144.19.74.201 au client Telnet d'adresse 144.19.74.44. Les paramètres de l'en-tête IP sont les suivants :

Version : 4

Longueur de l'en-tête (mots de 32 bits) : 5

TOS : 0 (préséance : routine, délai normal, débit normal, fiabilité normale)

Longueur totale : 44

Identification : 21410

Flags : 0, DF = 0, MF = 0 (fragmentation autorisée, dernier fragment)

Décalage de fragment : 0

Time to Live : 64 secondes

Protocol : 6 (TCP)

Total de contrôle : 0x720E (valide)

Adresse IP source : 144.19.74.201

Adresse IP destination : 144.19.74.44

Le champ Version vaut 4, le format du datagramme IP est donc IPv4.

Le champ Header Length fait 5 mots (20 octets). C'est la taille minimale de l'en-tête IP, aucune option n'est donc spécifiée.

La longueur totale de ce datagramme est de 44 octets. L'en-tête Ethernet et le total de contrôle CRC font 18 octets. En les additionnant, on obtient une taille de la trame Ethernet de 62 octets. Deux octets de remplissage sont ajoutés pour amener la taille de la trame Ethernet à 64 octets.

Ce datagramme IP a un champ Identification de valeur 21 410. Cette valeur est différente de celle qui est envoyée par le client Telnet. Cet identificateur a une valeur élevée, car l'hôte Telnet a envoyé de nombreux datagrammes IP avant cette session.



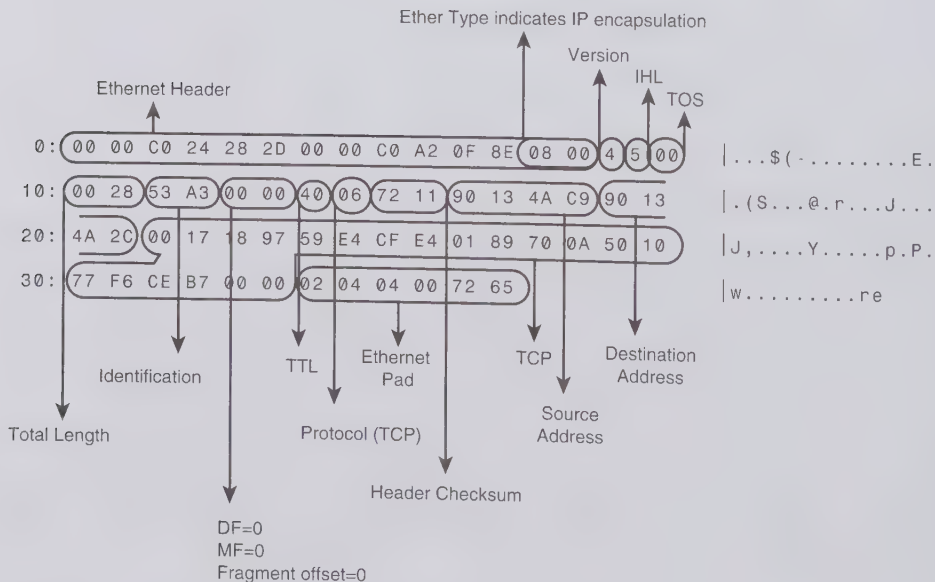
Figure 6.27

Datagramme IP
n°2.

```

Packet Number : 7           6:38:38 PM
Length : 64 bytes
ether: ===== Ethernet Datalink Layer =====
      Station: 00-00-C0-A2-0F-8E ----> 00-00-C0-24-28-2D
      Type: 0x0800 (IP)
ip: ===== Internet Protocol =====
      Station: 144.19.74.201 ----> 144.19.74.44
      Protocol: TCP
      Version: 4
      Header Length (32 bit words): 5
      Precedence: Routine
          Normal Delay, Normal Throughput, Normal Reliability
      Total length: 40
      Identification: 21411
      Fragmentation allowed, Last fragment
      Fragment Offset: 0
      Time to Live: 64 seconds
      Checksum: 0x7211(Valid)
tcp: ===== Transmission Control Protocol =====
      Source Port: TELNET
      Destination Port: 6295
      Sequence Number: 1508167652
      Acknowledgement Number: 25784330
      Data Offset (32-bit words): 5
      Window: 30710
      Control Bits: Acknowledgement Field is Valid (ACK)
      Checksum: 0xCEB7(Valid)
      Urgent Pointer: 0

```



Il convient aussi de noter que la valeur de TTL est de 64 secondes. Cette valeur de TTL est la valeur par défaut utilisée par l'hôte à 144.19.74.201, pour tous les datagrammes IP qu'il émet. Elle est différente de celle qui est envoyée par 144.19.74.44. Les valeurs initiales de TTL dépendent des implémentations de TCP/IP.

La valeur de TOS est 0. Cela signifie que ce datagramme IP doit être traité avec une préséance de routine, un délai normal, un débit normal et une fiabilité normale.

Le flag de fragmentation est à 0, de même que les flags DF et MF. Ce datagramme pourrait être fragmenté ; dans ce cas, il s'agirait du dernier fragment. La valeur nulle de Fragment Offset indique qu'il s'agit aussi du premier fragment. Puisque ce datagramme est le premier et le dernier fragment, il s'agit d'un datagramme complet, non fragmenté.

Le champ Protocol vaut 6, indiquant que le datagramme IP transporte un message TCP.

La valeur de Checksum est 720E (hexadécimal). Le total de contrôle est valide.

• Datagrammes IP n° 3 et 4 : étude du champ Identification

Les Figures 6.28 et 6.29 montrent les deux datagrammes IP suivants, échangés par le client Telnet d'adresse 144.19.74.44 et l'hôte Telnet d'adresse 144.19.74.201.

Les champs Identification de ces datagrammes sont des plus intéressants. Le datagramme de la Figure 6.28 est le deuxième datagramme envoyé par le client à l'hôte Telnet. Le champ d'identification de ce datagramme vaut 2, soit un de plus que celui du datagramme de la Figure 6.26. Le datagramme de la Figure 6.28 est le deuxième datagramme envoyé par l'hôte au client Telnet. Le champ d'identification de ce datagramme est 21 411, soit un de plus que celui du datagramme précédent (Figure 6.27). Cet exemple montre que les champs d'identification sont gérés séparément par chacune des parties.



Figure 6.28

Datagramme IP n°3.

```

Packet Number : 6                6:38:38 PM
Length : 69 bytes
ether: ===== Ethernet Datalink Layer =====
      Station: 00-00-C0-24-28-2D ---> 00-00-C0-A2-0F-8E
      Type: 0x0800 (IP)
ip: ===== Internet Protocol =====
      Station: 144.19.74.44 ---> 144.19.74.201
      Protocol: TCP
      Version: 4
      Header Length (32 bit words): 5
      Precedence: Routine
              Normal Delay, Normal Throughput, Normal Reliability
      Total length: 49
      Identification: 2
      Fragmentation allowed, Last fragment
      Fragment Offset: 0
      Time to Live: 100 seconds
      Checksum: 0xA1A9(Valid)
tcp: ===== Transmission Control Protocol =====
      Source Port: 6295
      Destination Port: TELNET
      Sequence Number: 25764321
      Acknowledgement Number: 1508167652
      Data Offset (32-bit words): 5
      Window: 4096
      Control Bits: Acknowledgement Field is Valid (ACK)
                    Push Function Requested (PSH)
      Checksum: 0x18A9(Valid)
      Urgent Pointer: 0
telnt: ===== Telnet Protocol =====
      Command: Do
      Option Code: Echo
      Command: Do
      Option Code: Suppress Go Ahead
      Command: Will
      Option Code:

```

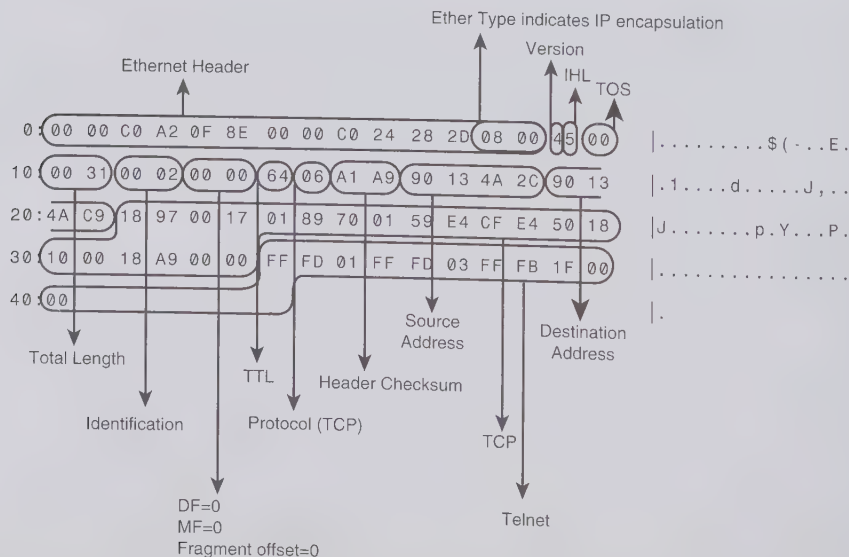


Figure 6.29

Datagramme IP
n°4.

```

Packet Number : 7           6:38:38 PM
Length : 64 bytes
ether: ===== Ethernet Datalink Layer =====
      Station: 00-00-C0-A2-0F-8E ----> 00-00-C0-24-28-2D
      Type: 0x0800 (IP)
ip: ===== Internet Protocol =====
      Station: 144.19.74.201 ----> 144.19.74.44
      Protocol: TCP
      Version: 4
      Header Length (32 bit words): 5
      Precedence: Routine
          Normal Delay, Normal Throughput, Normal Reliability
      Total length: 40
      Identification: 21411
      Fragmentation allowed, Last fragment
      Fragment Offset: 0
      Time to Live: 64 seconds
      Checksum: 0x7211(Valid)
tcp: ===== Transmission Control Protocol =====
      Source Port: TELNET
      Destination Port: 6295
      Sequence Number: 1508167652
      Acknowledgement Number: 25784330
      Data Offset (32-bit words): 5
      Window: 30710
      Control Bits: Acknowledgement Field is Valid (ACK)
      Checksum: 0xCEB7(Valid)
      Urgent Pointer: 0

```

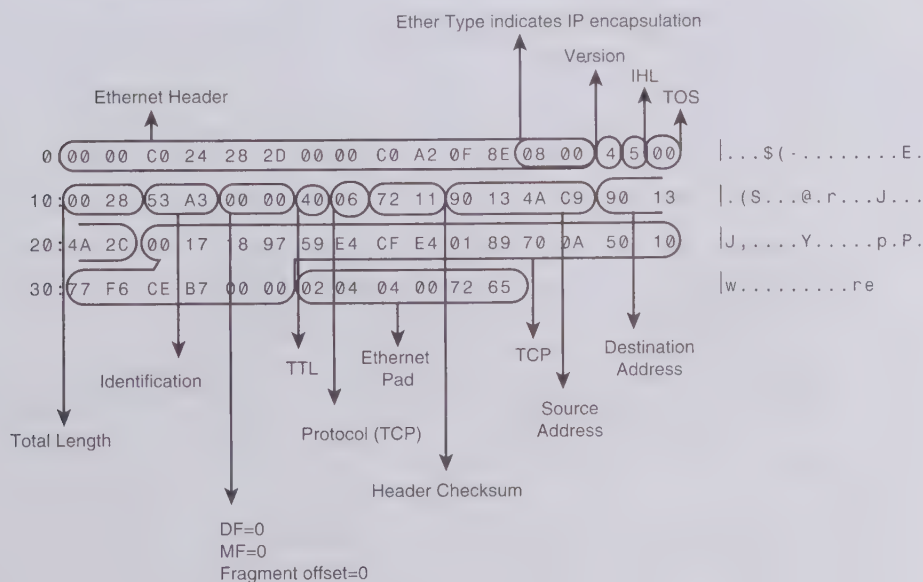


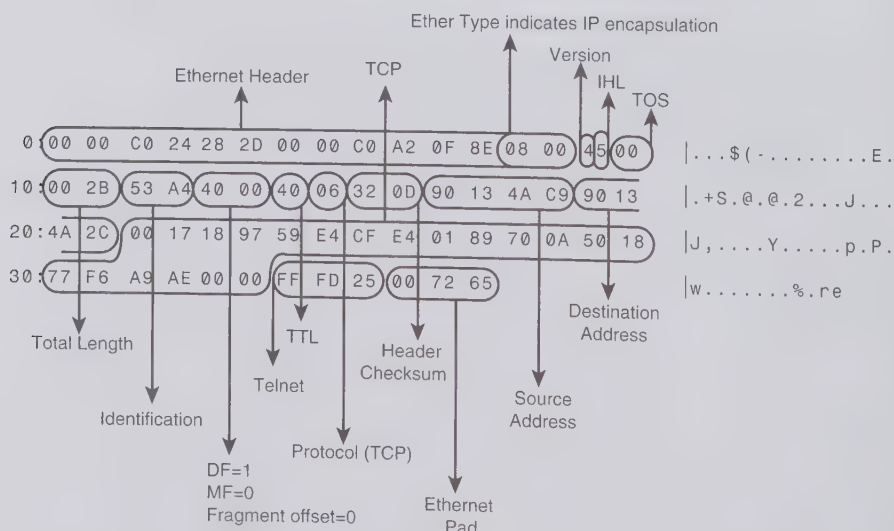
Figure 6.30

Datagramme IP n°5.

```

Packet Number : 8          6:38:39 PM
Length : 64 bytes
ether: ===== Ethernet Datalink Layer =====
      Station: 00-00-C0-A2-0F-8E ----> 00-00-C0-24-28-2D
      Type: 0x0800 (IP)
ip: ===== Internet Protocol =====
      Station:144.19.74.201 ---->144.19.74.44
      Protocol: TCP
      Version: 4
      Header Length (32 bit words): 5
      Precedence: Routine
          Normal Delay, Normal Throughput, Normal Reliability
      Total length: 43
      Identification: 21412
      Fragmentation not allowed, Last fragment
      Fragment Offset: 0
      Time to Live: 64 seconds
      Checksum: 0x320D(Valid)
tcp: ===== Transmission Control Protocol =====
      Source Port: TELNET
      Destination Port: 6295
      Sequence Number: 1508167652
      Acknowledgement Number: 25784330
      Data Offset (32-bit words): 5
      Window: 30710
      Control Bits: Acknowledgement Field is Valid (ACK)
                   Push Function Requested (PSH)
      Checksum: 0xA9AE(Valid)
      Urgent Pointer: 0
telnt: ===== Telnet Protocol =====
      Command: Do
      Option Code:

```



• Datagramme IP n° 5 : DF = 1

La Figure 6.30 montre un datagramme IP envoyé par l'hôte Telnet d'adresse 144.19.74.201 au client Telnet d'adresse 144.19.74.44. Les paramètres de l'en-tête IP sont les suivants :

Version : 4

Longueur d'en-tête (mots de 32 bits) : 5

TOS : 0 (Préséance : routine, délai normal, débit normal, fiabilité normale).

Longueur totale : 43

Identification : 21412

Flags : 0, DF = 1, MF = 0 (Fragmentation non autorisée, dernier fragment)

Décalage de fragment : 0

TTL : 64 secondes

Protocol : 6 (TCP)

Total de contrôle : 0x320D (Valide)

Adresse IP source : 144.19.74.201

Adresse IP destination : 144.19.74.44

Les valeurs de champs sont similaires à celles des paquets précédents. La seule différence tient dans les flags de fragmentation, puisque, ici, le flag DF vaut 1. Cela indique que l'hôte Telnet demande explicitement que ce datagramme ne soit pas fragmenté.



Récapitulatif

IP, le Protocole Internet, fournit une vue logique du réseau, quelles que puissent être les différentes technologies de réseau utilisées pour construire le réseau physique. Le réseau logique est indépendant des adresses de couche physique, de la taille de MTU et de toute autre caractéristique physique. Les protocoles de couche supérieure, tels que TCP et UDP, traitent le réseau comme s'il s'agissait d'un réseau purement IP. Le réseau IP fournit des services sans connexion aux services de couche supérieure. Ce service sans connexion est implémenté à l'aide de datagrammes contenant les adresses IP source et destination et d'autres paramètres nécessaires au fonctionnement de IP. Une caractéristique notable du fonctionnement de IP est que les datagrammes qui transitent sont fragmentés, si besoin est, par les routeurs, afin d'assurer que les datagrammes soient véhiculés dans la taille de paquet des réseaux sous-jacents. Lorsque la fragmentation IP survient, c'est le destinataire qui doit assembler les différents fragments du datagramme IP pour obtenir le datagramme originel.

Testez vos connaissances

1. Décrivez les caractéristiques d'un réseau IP logique.
2. Le réseau IP essaye-t-il d'ordonner les datagrammes qui arrivent dans le désordre ? A qui incombe cette tâche ?
3. Un réseau purement IP est-il fiable ? Que doit faire une application TCP/IP pour garantir la fiabilité de la transmission ?
4. Faites un schéma d'un en-tête IPv4. Nommez et présentez les différents champs.
5. Quelle est la taille minimale d'un en-tête IP ? Qu'est-ce qui rend la taille d'en-tête variable ?
6. Etant donné une taille de datagramme IP de 100 octets, quelle est la part due aux couches de liaison et réseau si le datagramme IP est envoyé sur un réseau Ethernet ?
7. Le datagramme IP fait 44 octets de long. Si ce datagramme est envoyé dans une trame Ethernet, un remplissage est-il nécessaire ? Si oui, combien d'octets doivent être ajoutés ?
8. Si un logiciel IP ne peut traiter que des datagrammes IPv4, que se passe-t-il s'il reçoit des datagrammes dont le champ Version ne vaut pas 4 ?

9. Quelle est la taille maximale de l'en-tête IPv4 ? Justifiez votre réponse. Pour cette taille maximale, combien d'octets sont utilisés pour spécifier des options IP ?
10. Quelles sont les différentes catégories de valeurs de préséance ? Quelle est la valeur de préséance habituellement utilisée pour les datagrammes IP générés par des applications TCP/IP commerciales ?
11. Comment le champ TOS est-il utilisé par les algorithmes de routage ?
12. Dans le cas d'applications nécessitant l'envoi de données interactives, quel est le code TOS le plus approprié ?
13. Dans le cas d'applications nécessitant le transfert de gros fichiers, quel est le code TOS le plus approprié ?
14. Dans le cas d'applications nécessitant l'envoi de données à moindre coût, quel est le code TOS le plus approprié ?
15. Quelles sont les tailles minimale et maximale d'un datagramme IP ? Rencontre-t-on souvent des datagrammes IP de taille minimale ? Quels sont les problèmes posés par l'envoi d'un datagramme IP de taille maximale ?
16. Quelle est la taille minimale de datagramme IP que tout nœud IP est tenu de savoir traiter ?
17. A quoi sert le champ Identification ?
18. Dans quelles circonstances un datagramme IP est-il envoyé avec un flag DF valant 1 ?
19. Si un datagramme IP est envoyé avec un flag DF à 1, et qu'un routeur s'aperçoit qu'il n'est pas en mesure de transmettre ce datagramme parce que la taille de MTU du réseau est inférieure à la taille du datagramme IP, que se passe-t-il ?
20. A quoi sert le champ TTL dans un datagramme IP ?
21. Etudiez le format de l'en-tête IP et trouvez la raison pour laquelle les tailles de fragments commencent à des décalages multiples de 8.
22. Comment un destinataire sait-il qu'il a reçu le dernier fragment ?
23. Comment un destinataire sait-il qu'un fragment IP est le premier fragment ?
24. Quelles sont les valeurs des flags DF et MF pour un datagramme IP non fragmenté ?



25. Un datagramme IP est reçu, avec un décalage de fragment de 0 et un flag MF à 1. Quelle conclusion tirer de ces paramètres ?
26. Quels sont les désavantages d'une fragmentation IP excessive ?
27. Lorsqu'un fragment IP est perdu, que se passe-t-il au niveau du destinataire ?
28. Un datagramme IP est reçu, avec un décalage de fragment de 0 et un flag MF à 0. Quelle conclusion tirer de ces paramètres ?
29. Un datagramme IP est reçu, avec un décalage de fragment de 200 et un flag MF à 1. Quelle conclusion tirer de ces paramètres ?
30. Un datagramme IP est reçu, avec un décalage de fragment de 600 et un flag MF à 0. Quelle conclusion tirer de ces paramètres ?
31. Un datagramme IP dont la taille est de 1 500 octets est envoyé avec un flag DF à 1 sur un réseau de taille de MTU de 1 500 octets. Le chemin jusqu'au nœud de destination passe par des réseaux de taille de MTU de 2 000 et 4 470 octets. Y aura-t-il fragmentation ? IP pourra-t-il remettre le datagramme à bon port ?
32. Un datagramme IP dont la taille est de 1 500 octets et dont le champ Identification vaut 100 est envoyé avec un flag DF à 0 sur un réseau de taille de MTU de 1 500 octets. Le chemin jusqu'au nœud de destination passe par un réseau de taille de MTU de 2 000 octets. Le réseau de destination a une taille de MTU de 1 500 octets. Y aura-t-il fragmentation ? Si oui, dressez la liste des fragments, en précisant les valeurs des champs Identification, MF, DF et Fragment Offset.
33. Un datagramme IP dont la taille est de 1 500 octets et dont le champ Identification vaut 100 est envoyé avec un flag DF à 0 sur un réseau de taille de MTU de 1 500 octets. Le chemin jusqu'au nœud de destination passe par un réseau de taille de MTU de 625 octets. Le réseau de destination a une taille de MTU de 1 500 octets. Y aura-t-il fragmentation ? Si oui, dressez la liste des fragments, en précisant les valeurs des champs Identification, MF, DF et Fragment Offset. Essayez de proposer l'algorithme utilisé par le destinataire pour réassembler les fragments IP.
34. Un datagramme IP dont la taille est de 1 500 octets et dont le champ Identification vaut 100 est envoyé avec un flag DF à 0 sur un réseau de taille de MTU de 1 500 octets. Le chemin jusqu'au nœud de destination passe par un réseau de taille de MTU de 625 octets. Le réseau de destination a une taille de MTU de 525 octets. Y aura-t-il fragmentation ? Si oui, dressez la liste des fragments, en précisant les valeurs des champs Identification, MF, DF et Fragment Offset.
35. Si la place manque pour insérer une marque de temps entière dans la zone de données de l'option Timestamp, que se passe-t-il ?
36. Que se passe-t-il en cas de dépassement de l'option Timestamp ?

37. Etant donné une adresse IP de 199.245.201.33, indiquez le Network Standard Byte Order (NSBO) pour la transmission de cette valeur sur le réseau. Numérotez les octets de cette adresse de 1 à 4 pour indiquer l'ordre de leur transmission. Numérotez les bits de chaque octet (en commençant par 0 pour le bit le plus significatif), et spécifiez l'ordre de transmission des bits.
38. Soit une valeur entière de 171, occupant un octet et constituant la valeur du champ Protocol. Spécifiez l'ordre de transmission des bits.
39. Indiquez les différences entre Strict Source Routing et Loose Source Routing.
40. Dans l'option Source Routing, comment sont utilisés les champs Pointeur et Longueur d'option pour déterminer si les données de routage ont bien été traitées ?
41. A quoi sert l'option Internet Timestamp ?

42. A partir du dump hexadécimal ci-après, décrivez un datagramme IP encapsulé dans une trame Ethernet, décodez les valeurs de l'en-tête IP. Faites un schéma de l'en-tête IP et placez les valeurs décodées dans les champs de l'en-tête IP.

```

0: 00 00 C0 24 28 2D 00 00 C0 A2 0F 8E 08 00 45 00 |...$(.....E.
10: 00 29 53 B2 40 00 40 06 32 01 90 13 4A C9 90 13 |...)S.@.2...J...
20: 4A 2C 00 17 18 97 59 E4 D0 4A 01 89 70 3A 50 18 |J,...Y..J..p:P.
30: 77 FF 59 0F 00 00 75 74 72 65 65 63             |w.Y...utreec

```

43. A partir du dump hexadécimal ci-après, décrivez un datagramme IP encapsulé dans une trame Ethernet, décodez les valeurs de l'en-tête IP. Faites un schéma de l'en-tête IP et placez les valeurs décodées dans les champs de l'en-tête IP.

```

0: 00 00 C0 A2 0F 8E 00 00 C0 24 28 2D 08 00 45 00 |.....$(...E.
10: 00 28 00 0C 00 00 64 06 A1 A8 90 13 4A 2C 90 13 |).(....d....J,..
20: 4A C9 18 97 00 17 01 89 70 3A 59 E4 D0 4B 50 10 |J.....p:Y..KP.
30: 10 00 36 17 00 00 00 00 00 00 FF FE 05         |..6.....

```

44. A partir du dump hexadécimal ci-après, décrivez un datagramme IP encapsulé dans une trame Ethernet, décodez les valeurs de l'en-tête IP. Faites un schéma de l'en-tête IP et placez les valeurs décodées dans les champs de l'en-tête IP. Quel est le protocole de couche supérieure encapsulé par l'en-tête IP ?

```

0: 00 00 C0 DD 14 5C 00 A0 24 AB D1 E6 08 00 45 00 |.....\...$....E.
10: 05 DC 3A 01 20 00 20 01 43 16 C7 F5 B4 14 C7 F5 |...:..C.....
20: B4 0A 08 00 EA FB 01 00 07 00 61 62 63 64 65 66 |.....abcdef
30: 67 68 69 6A 6B 6C 6D 6E 6F 70 71 72 73 74 75 76 |ghijklmnopqrstuv
40: 77 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F |wabcdefghijklmno
50: 70 71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 |pqrstuvwxyzabdefgh

```



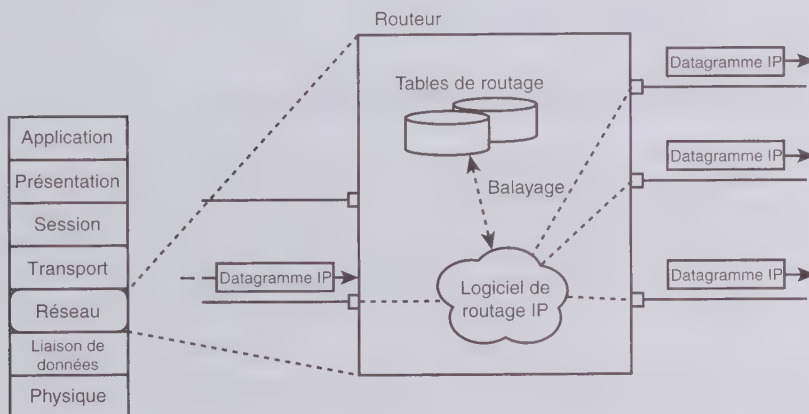
7 Le routage IP

IP permet de construire des réseaux aussi complexes que l'on veut. Ces réseaux complexes peuvent étendre la portée de réseaux sur de vastes distances. Les réseaux IP complexes sont reliés entre eux à l'aide de périphériques de relais appelés des *routeurs*. Les routeurs IP savent interpréter le format des en-têtes IP et transmettre les datagrammes IP en accord avec les informations figurant dans ces en-têtes. La transmission des datagrammes IP vers leurs destinations est appelée le *routage*. Les routeurs se comportent comme des périphériques de commutation de paquet, sélectionnant un chemin plutôt qu'un autre en se fondant sur des critères spécifiques. La plupart des routeurs IP commerciaux ne sont pas des routeurs purement IP, car ils peuvent être configurés pour router d'autres protocoles.

Concepts de base du routage

Comme nous l'avons vu dans le Chapitre 3, les routeurs sont des périphériques de couche OSI 3. La couche 3 du modèle OSI est chargée des fonctions de routage. La Figure 7.1 montre un modèle de routeur simplifié.

Figure 7.1
Un modèle de routeur simplifié.



Un routeur dispose de plusieurs ports, deux au minimum. Les datagrammes IP arrivent sur le port du routeur. Le logiciel de routage IP examine l'en-tête du datagramme pour déterminer la manière dont on doit la transmettre. L'information la plus déterminante est l'adresse de destination du datagramme. Le logiciel de routage consulte la table de routage qui se trouve sur le routeur, puis transmet le datagramme par le biais d'un des ports du routeur.

• Routage statique et routage dynamique

La table de routage contient une liste de réseaux et d'hôtes de destination. Elle contient aussi des informations sur la meilleure manière d'atteindre ces destinations. La façon dont les informations sont consignées dans la table de routage dépend de la méthode utilisée pour l'initialisation de la table :

- méthode statique ;
- méthode dynamique.

La méthode statique consiste à entrer manuellement les informations dans la table de routage. Dans le cas de la méthode dynamique, le routeur peut assimiler dynamiquement de nouvelles informations sur le routage vers d'autres réseaux et hôtes. Le routeur assimile ces informations grâce à des *protocoles de routage*.

Les réseaux modernes utilisent le plus souvent la méthode dynamique. La méthode statique convient pour des réseaux simples à la topologie immuable, ou à des fins de dépannage, lorsqu'il est nécessaire d'effectuer des corrections manuelles à la table de routage.

• Définition des protocoles de routage

Les protocoles de routage définissent la manière dont les routeurs s'échangent des informations afin de déterminer la meilleure route jusqu'à une destination sur l'Internet. Comme les protocoles de routage doivent échanger leurs messages dans un réseau IP, la transmission se fait à l'aide de datagrammes IP. Les protocoles de routage peuvent s'exécuter directement au-dessus de IP, comme le fait OSPF (*Open Shortest Path First*), au-dessus de UDP, un protocole de couche de transport, comme le fait RIP (*Routing Information Protocol*), ou au-dessus de TCP, un protocole de couche de transport, comme le fait BGP (*Border Gateway Protocol*).

Comme les protocoles de routage sont des clients de IP, TCP ou UDP, on peut les considérer comme une catégorie particulière de protocoles de couche d'application.

• Configuration des routeurs

Quel que soit le protocole de routage utilisé par un routeur, vous aurez toujours à configurer manuellement le routeur pour qu'il puisse fonctionner dans un réseau. En effet, un routeur est un périphérique de la couche 3 OSI, dont les interfaces de réseau sont identifiées par des adresses IP. Vous devez donc configurer manuellement les adresses IP et les paramètres associés de chaque interface de réseau du routeur. Certains routeurs commerciaux vous donnent la possibilité de définir ces paramètres dans un fichier pouvant être téléchargé en provenance d'un périphérique local ou du réseau. Ces fichiers de configuration permettent d'automatiser l'installation des routeurs, mais vous devez tout de même modifier manuellement le fichier de configuration.

Par tradition, la plupart des routeurs sont des hôtes d'application, des machines Unix par exemple, qui effectuent de plus des fonctions de routage. En fait, la plupart des systèmes Unix se comportent automatiquement comme des routeurs, par défaut, s'ils possèdent plus d'une interface de réseau (*hôte multiconnecté*). C'est aussi le cas de la plupart des systèmes d'exploitation sur PC.

Si un hôte serveur ou d'application est configuré pour tenir lieu de routeur, il doit effectuer ses tâches à la fois de routage et d'application. Comme chaque datagramme IP doit être routé séparément, l'algorithme de routage sur un réseau actif peut consommer une proportion non négligeable de la puissance de calcul de la machine. Cela peut affecter notablement les performances de la machine en tant que serveur d'applications. C'est la raison pour laquelle sont commercialisés des périphériques dont la seule finalité est le routage. Ces périphériques permettent de bien faire la distinction entre les fonctions de communication et celles d'applications. Cette séparation permet une meilleure gestion de ces deux aspects et évite tout conflit d'intérêts.

Bien qu'un hôte d'application polyvalent puisse tenir lieu de routeur, on supposera dans la suite de ce chapitre que les hôtes sont des machines qui exécutent des applications TCP/IP, telles que Telnet, FTP, NFS, SNMP, HTTP, etc. Les routeurs désignent des périphériques spécialisés, dotés d'au moins deux connexions, et qui n'effectuent que des fonctions de routage.

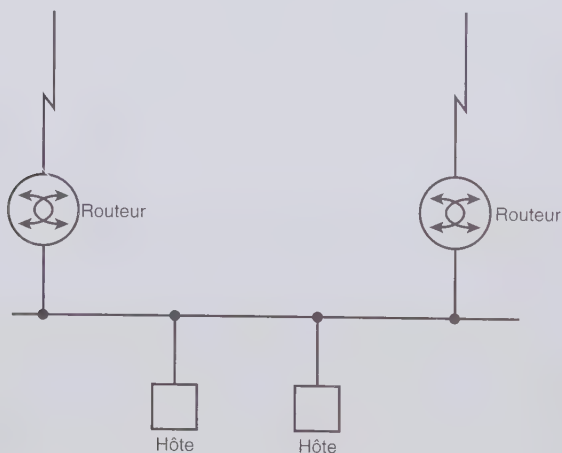


Remise des datagrammes

Les hôtes et les routeurs participent l'un comme l'autre à la remise d'un datagramme IP. Les datagrammes IP sont générés par un hôte émetteur qui veut contacter les services d'un autre hôte. La Figure 7.2 montre un réseau possédant des hôtes et des routeurs. Les routeurs fournissent des connexions vers des réseaux extérieurs.

Figure 7.2

Hôtes et routeurs.



Lorsque l'hôte émetteur envoie un datagramme IP, il doit décider s'il va envoyer son datagramme directement à un hôte du réseau, ou le transmettre à un des routeurs de ce même réseau. Cette décision est le routage.

Bien que les ports d'un routeur soient connectés au réseau, le routeur ne transmettra que les datagrammes qui ont été envoyés sur le port du routeur. Il incombe à l'hôte émetteur de transmettre le datagramme IP à un port de routeur essentiellement. Dans l'exemple de la Figure 7.2, l'hôte émetteur transmet le datagramme en prenant connaissance de l'adresse matérielle du port du routeur connecté au réseau, et en utilisant cette adresse matérielle dans le champ d'adresse de destination de la trame de couche de liaison de données. Dans un réseau à diffusion, un protocole tel que ARP (voir Chapitre 5) permet de connaître l'adresse matérielle du port du routeur.

• Types de remises de datagrammes

Un datagramme IP peut être remis à destination à l'aide d'une de ces deux méthodes :

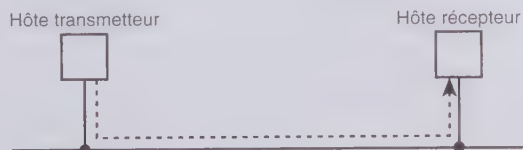
- remise directe ;
- remise indirecte.

Le routage d'un datagramme IP dans un réseau complexe se fait par combinaison de ces deux méthodes.

La remise directe a lieu lorsque l'émetteur et le destinataire se trouvent sur le même réseau physique. L'émetteur peut ainsi envoyer le datagramme IP directement au destinataire sans intervention d'un routeur (voir Figure 7.3). La méthode de transmission physique du réseau est utilisée pour la remise.

Figure 7.3

Remise directe de datagramme.



Dans un réseau à diffusion (*broadcast*) tel qu'un réseau local, la remise de datagramme IP peut être précédée par des paquets ARP, par exemple une requête ARP provenant de l'émetteur, pour découvrir l'adresse matérielle du destinataire, et une réponse ARP du destinataire, contenant son adresse matérielle. Une fois connue l'adresse matérielle du destinataire, la remise directe du datagramme est effectuée en plaçant l'adresse matérielle du destinataire dans le champ d'adresse de destination de la trame de couche de liaison de données, puis en envoyant cette trame.

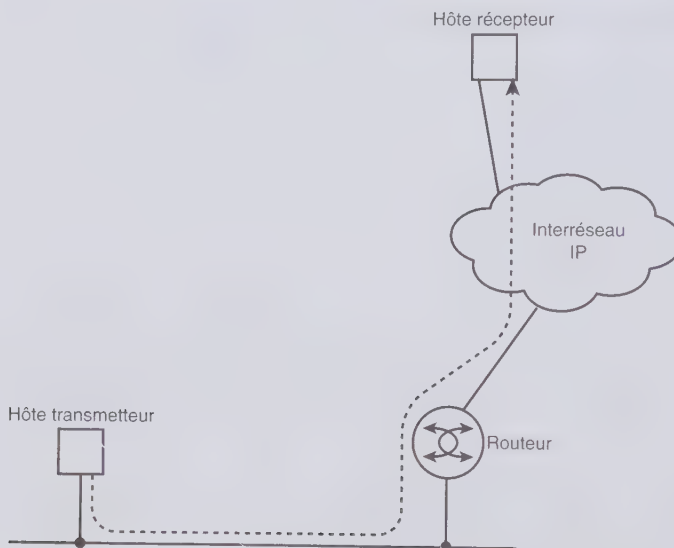
Dans le cas de la remise indirecte, le destinataire ne se trouve pas sur le même réseau physique que l'émetteur (voir Figure 7.4). L'émetteur doit alors transmettre le datagramme IP par le biais d'un des routeurs connectés à son réseau. Comme plusieurs routeurs peuvent être mis à contribution, on dit qu'il s'agit là d'une remise indirecte.

L'exemple de la Figure 7.5 montre qu'une remise de datagramme IP dans un réseau complexe se traduit par une combinaison de remises directe et indirecte.

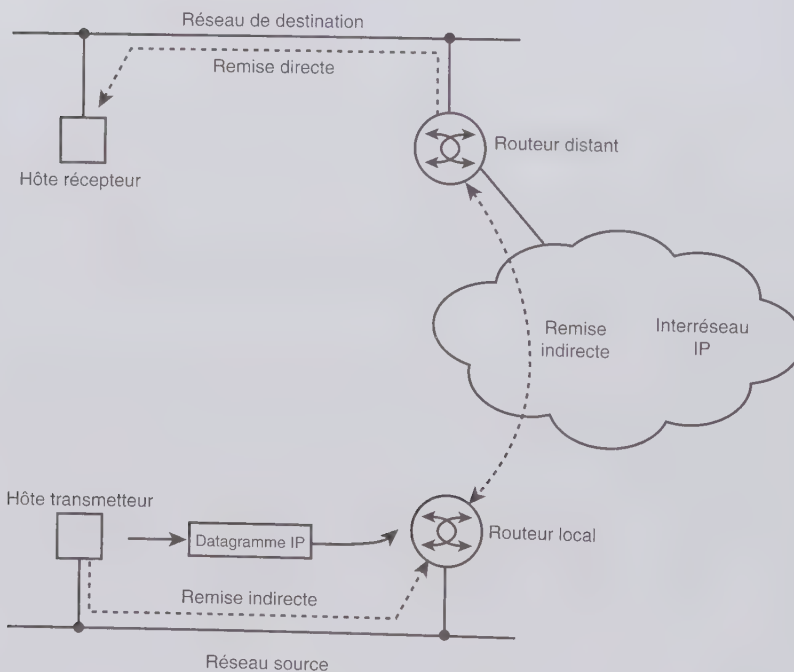


Figure 7.4

Remise indirecte de datagramme.

**Figure 7.5**

Remise d'un datagramme dans un réseau complexe.



La remise directe se fait de l'hôte émetteur au routeur local, et du routeur local au routeur distant. La remise indirecte se fait du routeur distant au destinataire. Si le réseau de destination est un réseau à diffusion, tel qu'un réseau local, le routeur distant utilisera le protocole ARP pour découvrir l'adresse matérielle de l'hôte de destination, avant de transmettre le datagramme.

• Un exemple de remises directe et indirecte

La meilleure manière de comprendre le processus mis en œuvre ici consiste encore à s'intéresser de près à la succession d'événements survenant lors de l'envoi du datagramme IP vers l'hôte de destination de la Figure 7.5. On supposera que les réseaux source et destination sont des réseaux locaux Ethernet. Le tableau suivant contient les paramètres des périphériques de la Figure 7.5.

Adresse	Paramètre
Adresse IP émetteur	199.245.180.1
Adresse matérielle émetteur	08000034AD01
Adresse IP routeur local	199.245.180.99
Adresse matérielle routeur local	0004DAC0031A
Adresse IP destination	200.1.1.1
Adresse matérielle destination	0000C000CDA4
Adresse IP routeur distant	200.1.1.50
Adresse matérielle routeur distant	0004DAC006AA

L'émetteur détermine grâce à ses fonctions de routage que l'adresse IP de destination ne se trouve pas sur le réseau local. Il est donc nécessaire de transmettre le datagramme au routeur local. Avant de transmettre le datagramme, l'émetteur doit toutefois connaître l'adresse matérielle du port du routeur local, car on ne peut envoyer le datagramme qu'au port de routeur local utilisant le réseau local Ethernet. L'adresse matérielle du port du routeur local doit être placée dans le champ d'adresse de destination de la trame Ethernet. L'émetteur utilise le protocole ARP pour prendre connaissance de l'adresse matérielle. Autrement dit, l'émetteur utilise ARP pour faire correspondre l'adresse IP du port du routeur local à son adresse locale. Ce processus passe généralement par une diffusion (*broadcast*) de requête



ARP, suivie d'une réponse ARP. Les deux premiers paquets sont donc les suivants (voir Tableaux 7.1 et 7.2) :

Tableau 7.1 : Paquet 1, diffusion de requête ARP

Champ	Valeur
<i>Trame Ethernet</i>	
AD Ethernet	FFFFFFFFFFFF
AS Ethernet	08000034AD01
Type Ethernet	0806 (hexadécimal)
<i>En-tête header</i>	
Type de matériel	1
Type de protocole	800 (hexadécimal)
Hlen	= 6
Plen	4
Opération	1 (requête ARP)
Adresse matérielle émetteur	08000034AD01
Adresse IP émetteur	199.245.180.1
Adresse matérielle destination	xxxx (la valeur recherchée)
Adresse IP destination	199.245.180.99

Tableau 7.2 : Paquet 2, réponse ARP

Champ	Valeur
<i>Trame Ethernet</i>	
DA Ethernet	08000034AD01
SA Ethernet	0004DAC0031A
Type Ethernet	0806 (hexadécimal)
<i>En-tête ARP</i>	
Type de matériel	1
Type de protocole	800 (hexadécimal)
Hlen	6
Plen	4
Opération	2 (réponse ARP)

Tableau 7.2 : Paquet 2, réponse ARP (suite)

Champ	Valeur
Adresse matérielle émetteur	0004DAC0031A (la réponse recherchée)
Adresse IP émetteur	199.245.180.99
Adresse matérielle destination	08000034AD01
Adresse IP destination	199.245.180.1

Une fois établie la correspondance entre adresse IP et adresse matérielle du routeur local, l'émetteur transmet le datagramme au routeur local à l'aide d'une trame Ethernet. Cet enchaînement constitue un exemple de remise directe du datagramme IP au port du routeur local. La trame suivante est le paquet n° 3 (voir Tableau 7.3).

Tableau 7.3 : Paquet 3, remise directe du datagramme IP au port du routeur local

Champ	Valeur
<i>Trame Ethernet</i>	
Adresse de destination Ethernet	08000034AD01
Adresse source Ethernet	0004DAC0031A
Type Ethernet	0800 (hexadécimal)
<i>En-tête IP</i>	
Version	4
Longueur d'en-tête (mots de 32 bits)	5
TOS	0 (Préséance : routine, retard normal, débit normal, fiabilité normale)
Longueur totale	xxxx (dépend de la taille des données de protocole de couche supérieure)
Identification	1 (la valeur réelle dépend des datagrammes IP envoyés précédemment)
Flags	0, DF = 0, MF = 0 (fragmentation autorisée, dernier fragment)
Fragment offset	0



Tableau 7.3 : Paquet 3, remise directe du datagramme IP au port du routeur local (suite)

Champ	Valeur
TTL	64 secondes (la valeur réelle dépend de l'implémentation IP de l'hôte émetteur)
Protocole	6 (protocole de couche supérieure TCP)
Total de contrôle	xxxx (la valeur réelle dépend des données de protocole de couche supérieure)
Adresse IP source	199.245.180.1
Adresse IP destination	200.1.1.1

Le routeur local consulte sa table et transmet le datagramme IP au routeur suivant. Ce processus est répété jusqu'à ce que le datagramme IP arrive au routeur distant. Le champ TTL du datagramme est décrémenté à chaque routeur. Supposons que le datagramme IP soit traité par cinq routeurs, chacun d'entre eux décrémentant de une seconde le champ TTL. La valeur de ce champ TTL, avant que le datagramme ne quitte le routeur distant, est donc de $64 - 5 = 59$ secondes.

Une fois le datagramme IP arrivé, les fonctions de routage déterminent que l'adresse IP de destination (200.1.1.1) peut être atteinte par le biais du port du routeur d'adresse IP 200.1.1.50. Avant de transmettre le datagramme IP à l'adresse IP 200.1.1.1, le routeur distant doit déterminer l'adresse matérielle de l'hôte correspondant à cette adresse. En effet, le datagramme doit être envoyé au moyen de la technologie de réseau sous-jacente du réseau de destination, en l'occurrence Ethernet. L'adresse matérielle de l'hôte de destination, 200.1.1.1, doit être placée dans le champ d'adresse de destination de la trame Ethernet envoyée du port du routeur distant vers la destination finale. Là encore, on utilise le protocole ARP pour découvrir l'adresse matérielle correspondant à l'adresse IP. Une requête ARP est envoyée par le port du routeur distant (voir Tableau 7.4), et une réponse ARP contenant l'adresse matérielle de 200.1.1.1 est reçue (voir Tableau 7.5).

Tableau 7.4 : Paquet 4, requête ARP (en diffusion)

Champ	Valeur
<i>Trame Ethernet</i>	
Adresse de destination Ethernet	FFFFFFFFFFFF
Adresse source Ethernet	0004DAC006AA
Type Ethernet	0806 (hexadécimal)

Tableau 7.4 : Paquet 4, requête ARP (en diffusion) (suite)

Champ	Valeur
<i>En-tête ARP</i>	
Type de matériel	1
Type de protocole	800 (hexadécimal)
Hlen	6
Plen	4
Opération	1 (requête ARP)
Adresse matérielle émetteur	0004DAC006AA
Adresse IP émetteur	200.1.1.50
Adresse matérielle destination	xxxx (la valeur recherchée)
Adresse IP destination	200.1.1.1

Tableau 7.5 : Paquet 5, réponse ARP

Champ	Valeur
<i>Trame Ethernet</i>	
Adresse de destination Ethernet	0004DAÇ006AA
Adresse source Ethernet	0000C000CDA4
Type Ethernet	0806 (hexadécimal)
<i>En-tête ARP</i>	
Type de matériel	1
Type de protocole	800 hex
Hlen	6
Plen	4
Opération	2 (réponse ARP)
Adresse matérielle émetteur	0000C000CDA4 (la valeur recherchée)
Adresse IP émetteur	200.1.1.1
Adresse matérielle destination	0004DAC006AA
Adresse IP destination	200.1.1.50

P 20 T



LE MACMILLAN

Après avoir fait correspondre adresse matérielle et adresse IP, le routeur distant transmet le datagramme par le biais de son interface de réseau d'adresse 200.1.1.50. Le datagramme IP est encapsulé à l'aide d'une trame Ethernet. Le paquet 6 constitue un exemple de remise directe du datagramme IP à l'hôte de destination par le port du routeur distant (voir Tableau 7.6).

Tableau 7.6 : Paquet 6, remise directe du datagramme IP à l'hôte de destination par le routeur distant

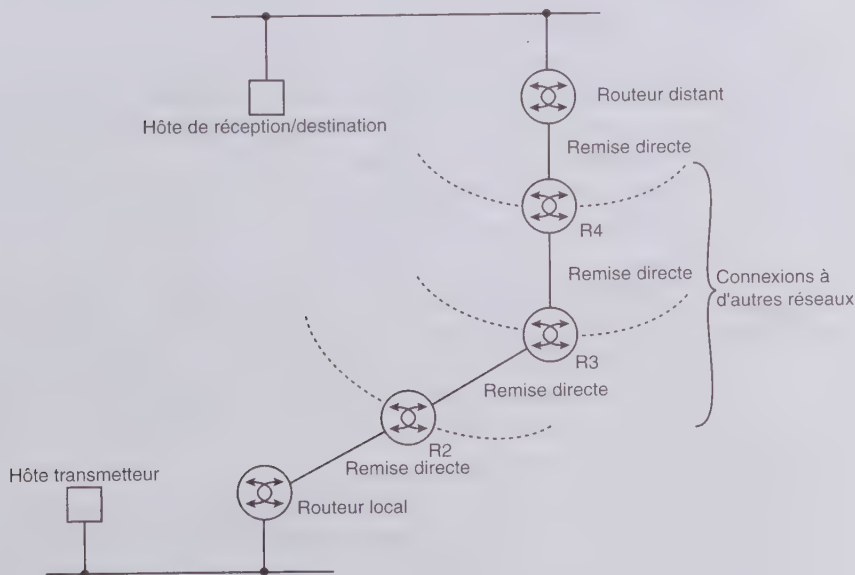
Champ	Valeur
<i>Trame Ethernet</i>	
Adresse de destination Ethernet	0000C000CDA4
Adresse source Ethernet	0004DAC006AA
Type Ethernet	0800 (hexadécimal)
<i>En-tête IP</i>	
Version	4
Longueur d'en-tête (mots de 32 bits)	= 5
TOS	0 (préséance : routine, retard normal, débit normal, fiabilité normale)
Longueur totale	XXX (dépend de la taille des données de protocole de couche supérieure)
Identification	1 (dépend des datagrammes IP envoyés précédemment)
Flags	0, DF = 0, MF = 0 (fragmentation autorisée, dernier fragment)
Fragment offset	0
TTL	59 secondes (la valeur réelle dépend de l'implémentation de IP de l'hôte émetteur)
Protocole	6 (protocole de couche supérieure TCP)
Total de contrôle	xxxx (la valeur réelle dépend des données de protocole de couche supérieure)
Adresse IP source	199.245.180.1
Adresse IP destination	200.1.1.1

• Analyse des remises directe et indirecte

Si l'on regarde le datagramme IP des paquets 3 et 6, on voit que les adresse IP source et destination sont restées les mêmes, bien que le paquet ait été traité par différents routeurs. La valeur du champ TTL a été décrémentée pour tenir compte du temps passé sur le réseau, mais c'est là la seule modification notable. En revanche, en examinant les en-têtes Ethernet des paquets 3 et 6, on constate que les adresses source et destination dans les trames Ethernet sont bien différentes. Cette différence est prévisible puisque les adresses de couche de liaison de données sont utilisées dans les paquets au sein d'un même segment de réseau physique. Ces adresses ne sont généralement pas utilisées dans des protocoles au-delà du segment. Dans le cas du paquet 3, les adresses Ethernet source et destination étaient celles de l'hôte émetteur et du port du routeur local dans le réseau émetteur. Dans le cas du paquet 6, les adresses Ethernet source et destination étaient celles du port du routeur distant et de l'hôte de destination dans le réseau de destination. Les informations concernant les adresses Ethernet ne sont jamais envoyées en dehors du réseau Ethernet lui-même. Les datagrammes IP et la plupart des protocoles de couche supérieure ne disposent pas de place où faire figurer ces adresses.

Dans la Figure 7.5, le transport du datagramme IP du routeur local au routeur distant est présenté comme un exemple de remise indirecte. Toutefois, une remise

Figure 7.6
La remise indirecte se décompose en une succession de remises directes.



indirecte peut être décomposée en une série de remises directes. Prenons l'exemple de la Figure 7.6, dans laquelle trois routeurs séparent les routeurs local et distant. Le routeur local consulte sa table de routage et détermine que le meilleur chemin pour le datagramme IP implique la transmission du datagramme au routeur R2. Le routeur local et le routeur R2 sont connectés au même segment de réseau physique. Le transport du datagramme entre ces deux routeurs constitue donc une remise directe. Le protocole de couche de liaison de données du réseau sous-jacent est utilisé pour transporter le datagramme IP. On peut ainsi utiliser le protocole PPP pour envoyer le datagramme du routeur local au routeur R2.

De même, les transports du datagramme de R2 à R3, de R3 à R4 et de R4 au routeur distant constituent des remises directes.

Pour résumer, toute remise indirecte d'un datagramme IP se décompose en fait en une succession de remises directes.

Tables de routage de l'hôte et du routeur

Hôtes aussi bien que routeurs disposent de tables de routage IP. La table de routage contient des informations sur des destinations et la manière d'atteindre ces destinations. Les hôtes et les routeurs consultent leur table de routage pour déterminer comment router un datagramme.

Dans le cas d'un hôte, le module IP consulte la table de routage de l'hôte pour les datagrammes sortants nécessitant des remises indirectes. Dans la section précédente, nous avons vu que les remises indirectes ont lieu lorsque la destination et la source ne se trouvent pas sur un même réseau. Les routeurs consultent leur table de routage pour les datagrammes entrants et créent ensuite un chemin en fonction des informations qu'ils y trouvent.

Quelles sont les informations stockées dans la table de routage ? Si cette table contenait la liste de tous les hôtes de destination possibles, elle serait énorme et impossible à gérer, car elle devrait contenir une entrée pour chaque adresse IP de destination possible, soit un total théorique de quatre milliards d'adresses IPv4 ! Les ordinateurs n'auraient pas assez de mémoire pour stocker ces tables, et les gestionnaires de réseau pas assez de patience pour les gérer. Une solution plus adaptée est donc nécessaire.

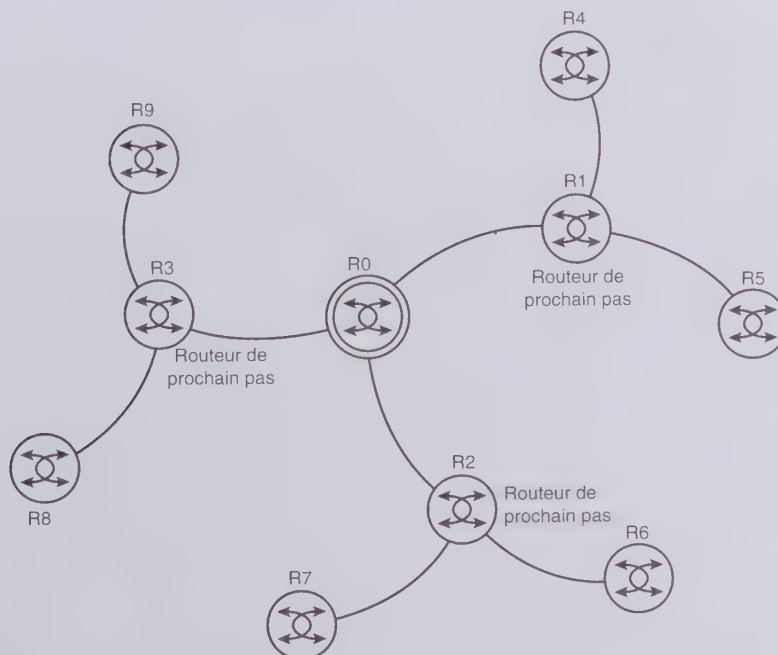
La table de routage IP est conçue de telle manière que le routage peut s'effectuer avec un minimum d'informations dans la table. Plutôt que de stocker l'adresse IP de tous les hôtes imaginables, on stocke le numéro de réseau du réseau de destination. Vous devez vous souvenir qu'une adresse IP est constituée d'un ID de réseau et d'un ID d'hôte. L'ID de réseau est le numéro du réseau, qui est le même pour tous les hôtes de ce réseau. Autrement dit, les hôtes présents sur un même réseau

physique ont en commun un préfixe d'adresse IP appelé l'ID de réseau. En utilisant l'ID de réseau dans la table de routage, on réduit considérablement le nombre des entrées. Ainsi, si l'adresse IP d'un réseau contient H bits pour l'hôte, en utilisant le numéro de réseau plutôt que les adresses des hôtes, on réduit le nombre des entrées dans la table de routage par un facteur de 2^H .

Une entrée de table de routage contient la destination et le routeur de prochain pas (*next hop*) pour transmettre le datagramme (le passage d'un datagramme à travers un routeur constitue un pas ou *hop*). Le routeur de prochain pas est un routeur connecté au même segment de réseau physique. La Figure 7.7 montre que les routeurs R1, R2 et R3 sont des routeurs de prochain pas pour le routeur R0, car ils sont connectés au même segment de réseau physique. Les routeurs R0 et R1 sont connectés par une liaison point à point. De même, les routeurs R0, R2 et R0, R3 sont connectés par une liaison point à point.

Figure 7.7

Routeur de prochain pas.



Il est important de bien comprendre que la table de routage ne pointe que sur le routeur suivant dans le même réseau physique et ne contient pas le chemin complet jusqu'à la destination. Autrement dit, l'entrée de routage ne contient pas la liste de tous les routeurs par lesquels doit transiter le datagramme. Dans l'exemple de la Figure 7.7, la table de routage du routeur R0 ne contient que les entrées pour R1, R2 et R3. La table de routage de R0 ne mentionne pas les routeurs R4, R5, R6, R7 et R8, car ceux-ci ne sont pas des routeurs de prochain pas par rapport à R0.

Le routeur de prochain pas est sélectionné comme étant la destination intermédiaire la moins coûteuse pour atteindre la destination finale. Le coût minimal est mesuré en termes de retards, de charges financières ou de nombres de pas. Le protocole de routage utilisé par les routeurs pour communiquer des informations de routage spécifie le critère d'estimation du coût minimal. Le coût associé à l'envoi d'un datagramme vers un routeur de prochain pas donné est évalué à partir d'une connaissance globale de la topologie du réseau. Mais cette connaissance se limite à la topologie du réseau local.

Comme la table de routage ne contient que des informations concernant les routeurs voisins, elle reste simple. Elle doit tenir compte des modifications affectant les routeurs voisins. Il incombe au protocole de routage de calculer le coût afférent au transfert vers un routeur voisin particulier.

L'utilisation du numéro de réseau dans la table de routage plutôt que l'adresse IP des hôtes réduit la taille de la table, mais a quelques inconvénients. Tous les hôtes d'un même réseau sont représentés par une seule entrée et donc un seul chemin. Dans ce modèle simple, on ne tient pas compte des retards, priorités ou débits concernant le trafic du réseau.

Certains protocoles de routage s'efforcent de prendre en compte ces facteurs en associant un coût à chaque route figurant dans la table de routage. L'hôte de destination est atteint après une succession de remises directes de l'émetteur au routeur final. Tant que le routeur final n'a pas tenté de remettre le datagramme à l'hôte de destination, il n'y a pas moyen de savoir si l'hôte de destination existe bien. Si le routeur final n'arrive pas à remettre le datagramme, il envoie à l'émetteur un rapport d'erreurs sous forme d'un message ICMP (*Internet Control Message Protocol*). Si l'hôte de destination se trouve sur un réseau à diffusion, tel qu'un réseau local Ethernet, le routeur final tente de remettre le datagramme en utilisant ARP pour faire correspondre l'adresse IP de destination et l'adresse matérielle pour le réseau. Si cette résolution échoue, le routeur final considère que l'hôte de destination ne peut pas être atteint.

La table de routage d'un hôte est similaire à celle d'un routeur. La différence tient dans le fait qu'un hôte ne route que des datagrammes IP émis par lui-même. Il ne transmettra pas de datagrammes provenant d'autres nœuds IP, à la différence des routeurs.

Les sections qui suivent examinent des exemples concrets de tables de routage d'hôtes et de routeurs.

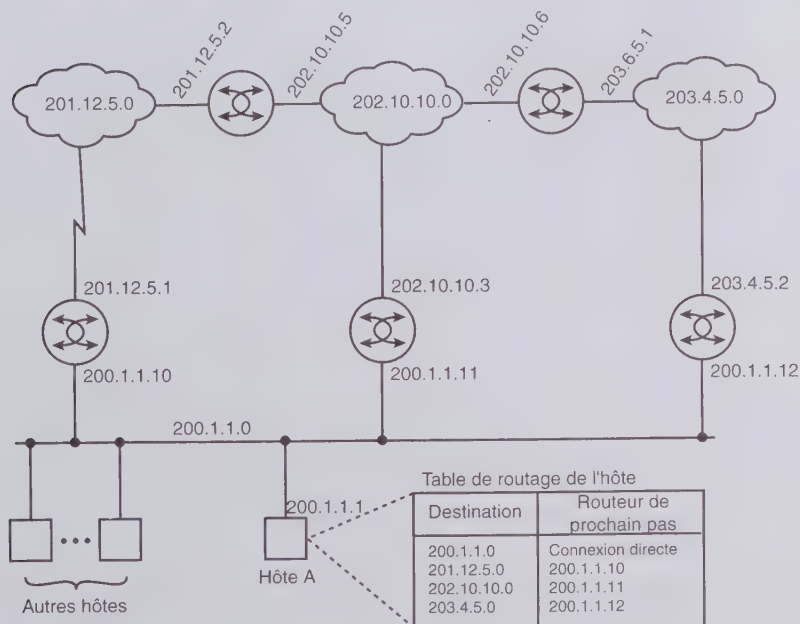
• Un exemple de table de routage d'hôte

La Figure 7.8 montre la table de routage d'un hôte. Le réseau contient trois routeurs. L'hôte A peut choisir n'importe lequel des routeurs, mais il ne le fait pas de manière arbitraire. Il consulte sa table de routage à cet effet. La table de routage de l'hôte A contient les entrées suivantes :

Destination	Routeur de prochain pas
200.1.1.0	Connexé directement
201.12.5.0	200.1.1.10
202.10.10.0	200.1.1.11
203.4.5.0	200.1.1.12

Figure 7.8

Un exemple de table de routage d'hôte.



La première entrée dans la table de routage concerne le réseau local 200.1.1.0. La colonne routeur de prochain pas indique qu'il s'agit d'un réseau directement connecté, ce qui signifie que l'hôte peut effectuer une remise directe à la destination, et qu'aucun datagramme destiné au réseau local 200.1.1.0 n'aura besoin d'être transmis à un routeur. L'hôte utilise le mécanisme de transport physique pour transmettre le datagramme à tout hôte du réseau 200.1.1.0. Dans le cas d'un réseau à diffusion tel qu'Ethernet, la transmission du datagramme nécessite l'emploi de ARP pour déterminer l'adresse matérielle de l'hôte de destination local.

La deuxième entrée de la table de routage indique que tous les datagrammes pour le réseau 201.12.5.0 doivent être transmis au port du routeur 200.1.1.10. Ce réseau peut être atteint par le biais d'autres routeurs locaux, mais les chemins correspondants traverseraient d'autres réseaux, tels que 202.10.10.0 et 203.4.5.0. Le protocole de routage est assez optimisé pour déterminer que le meilleur chemin jusqu'au réseau 201.12.5.0 passe par le routeur 201.1.1.10. Il se fonde pour cela sur le calcul du coût de chaque chemin. Si le routeur à 201.1.1.10 est hors service, l'entrée de routage pour 201.12.5.0 indiquera d'autres routeurs permettant d'atteindre la destination.

De manière similaire, la troisième entrée indique que tous les datagrammes pour le réseau 202.10.10.0 doivent être transmis au port du routeur 200.1.1.11.

La dernière entrée de la table de routage indique que tous les datagrammes à destination du réseau 203.4.5.0 doivent être transmis au port du routeur 200.1.1.12. Là encore, on pourrait atteindre ce réseau par le biais d'autres routeurs, mais il faudrait pour cela traverser d'autres réseaux, tels que 201.12.5.0 et 202.10.10.0. Le protocole de routage est assez optimisé pour déterminer que le meilleur chemin jusqu'au réseau 203.4.5.0 passe par le routeur 201.1.1.12. Il se fonde pour cela sur le calcul du coût de chaque chemin. Si le routeur à 201.1.1.12 est hors service, l'entrée de routage pour 203.4.5.0 indiquera d'autres routeurs permettant d'atteindre la destination.

Si vous examinez les adresses IP de routeurs de prochain pas, vous pouvez remarquer que le port connecté au réseau local a le même préfixe d'adresse 200.1.1. De plus, ce préfixe est le même que celui du réseau local et de tous les hôtes connectés au réseau local 200.1.1.0. En effet, les routeurs de prochain pas ont en commun le préfixe du réseau sur lequel sont connectés leurs ports. Comme un routeur doit posséder plus d'un port, les autres ports doivent aussi avoir le même préfixe de réseau que le port sur lequel il sont connectés (voir Figure 7.8).

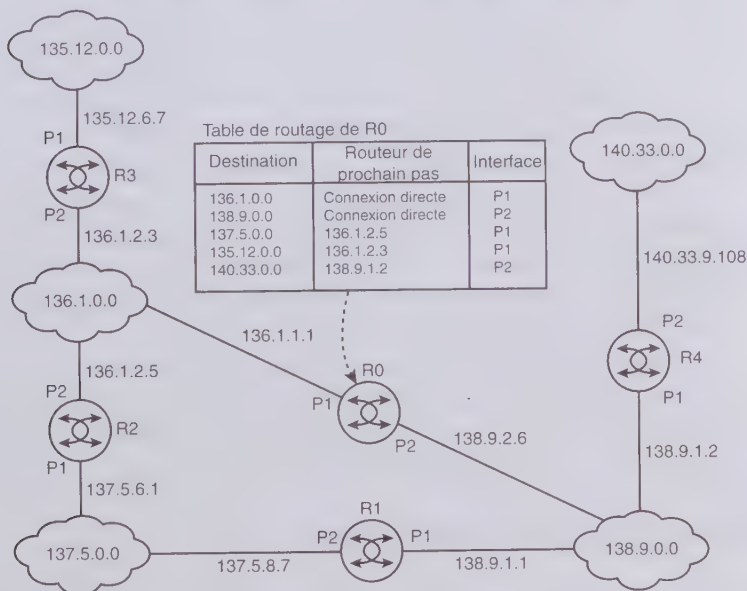
• Un exemple de table de routage de routeur

À la différence des hôtes, les routeurs peuvent transmettre des datagrammes IP provenant d'autres nœuds IP. La Figure 7.9 montre la table de routage pour le routeur R0 dans un interréseau constitué de cinq réseaux et de cinq routeurs. La table de routage de R0 contient les entrées suivantes :

Destination	Routeur de prochain pas	Interface
136.1.0.0	Connecté directement	P1
138.9.0.0	Connecté directement	P2
137.5.0.0	136.1.2.5	P1
135.12.0.0	136.1.2.3	P1
140.33.0.0	138.9.1.2	P2

Figure 7.9

Un exemple de table de routage de routeur.



Les première et deuxième entrées de la table concernent les réseaux connectés localement 136.1.0.0 et 138.9.0.0. Comme les routeurs disposent de plus d'un port, ils donnent en interne une étiquette de port à chacun d'eux. Dans l'exemple de la Figure 7.9, les interfaces du routeur ont pour étiquettes P1 et P2. Une implémentation de routeur peut utiliser le système d'étiquetage de son choix. Le routeur de prochain pas pour le réseau connecté localement indique que ces réseaux sont connectés directement. Il est donc possible d'utiliser une remise directe pour atteindre un hôte appartenant à un de ces réseaux. La troisième colonne indique le numéro d'interface du port de routeur à travers lequel on peut atteindre la destination.



Le réseau connecté directement 136.1.0.0 peut être atteint via le port P1, et le réseau connecté directement 138.9.0.0, par le port P2.

La troisième entrée dans la table indique que l'on peut atteindre le réseau 137.5.0.0 en transmettant le datagramme au routeur d'adresse 136.1.2.5 via le port P1. Remarquons au passage que l'adresse IP du port P1 est 136.1.1.1, et celle du port du routeur est 136.1.2.5. Le routeur à 136.1.2.5 est un routeur de prochain pas pour R0, en effet les routeurs ont le même préfixe de réseau 136.1, correspondant au réseau de classe B 136.1.0.0.

La quatrième entrée dans la table indique que l'on peut atteindre le réseau 135.12.0.0 en transmettant le datagramme au routeur d'adresse 136.1.2.3 via le port P1. Remarquons au passage que l'adresse IP du port P1 est 136.1.1.1, et celle du port du routeur est 136.1.2.3. Le routeur à 136.1.2.5 est un routeur de prochain pas pour R0, en effet les routeurs ont le même préfixe de réseau 136.1, correspondant au réseau de classe B 136.1.0.0.

La cinquième entrée dans la table indique que l'on peut atteindre le réseau 140.33.0.0 en transmettant le datagramme au routeur d'adresse 138.9.1.2 via le port P2. Remarquons au passage que l'adresse IP du port P2 est 138.9.2.6, et celle du port du routeur est 136.9.1.2. Le routeur à 138.9.1.2 est un routeur de prochain pas pour R0, en effet les routeurs ont le même préfixe de réseau 138.9, correspondant au réseau de classe B 138.9.0.0.

Comme nous l'avons vu précédemment, les adresses IP de routeur de prochain pas doivent avoir le même préfixe d'adresse que le port utilisé pour transmettre le datagramme IP. Cela indique que l'on peut utiliser la remise directe pour la transmission du datagramme au routeur de prochain pas.

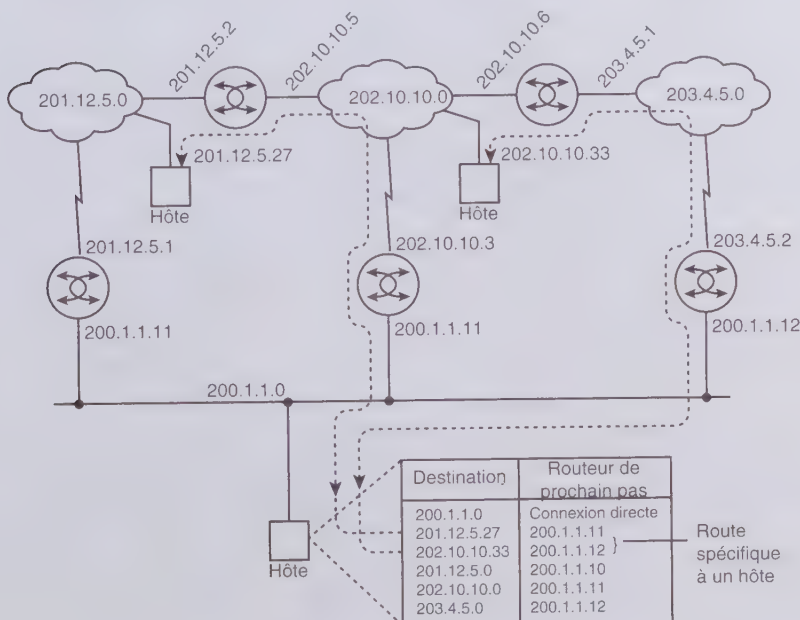
• Routes spécifiques des hôtes

La plupart des entrées de routage indiquent le numéro de réseau comme destination. Cependant, de nombreuses implémentations permettent de spécifier une adresse d'hôte IP comme destination. Lorsque l'adresse IP d'un hôte est utilisée comme destination dans une table de routage, la maîtrise de la transmission est meilleure, car on peut alors spécifier le routeur de prochain pas en fonction de l'adresse IP plutôt qu'en fonction du numéro de réseau, plus général. Cela est particulièrement utile pour les tests et le débogage. Prenons l'exemple du réseau de la Figure 7.10, indiquant la table de routage suivante :

Destination	Routeur de prochain pas
200.1.1.0	Connecté directement
201.12.5.27	200.1.1.11 <i>Route spécifique pour un hôte</i>
202.10.10.33	200.1.1.12 <i>Route spécifique pour un hôte</i>

Destination	Routeur de prochain pas
201.12.5.0	200.1.1.10
202.10.10.0	200.1.1.11
203.4.5.0	200.1.1.12

Figure 7.10
Routes spécifiques des hôtes.



L'exemple de la Figure 7.10 est similaire à celui de la Figure 7.8, à ceci près qu'ici les hôtes 201.12.5.27 et 202.10.10.33 ont des routes directement spécifiées pour eux dans la table de routage de l'hôte A.

Si on n'utilisait pas la route spécifique de l'hôte, on atteindrait la destination 201.12.5.27 via le routeur 200.1.1.11, spécifié pour le réseau de destination 201.12.5.0. De même, on atteindrait la destination 202.10.10.33 via le routeur 200.1.1.11, spécifié pour le réseau de destination 201.12.5.0. Pour le test ou le débogage, il arrive qu'il soit nécessaire d'envoyer des données à un hôte via un routeur de prochain pas bien déterminé. Ces routes spécifiques d'un hôte permettent de le faire, comme le montre la Figure 7.10.

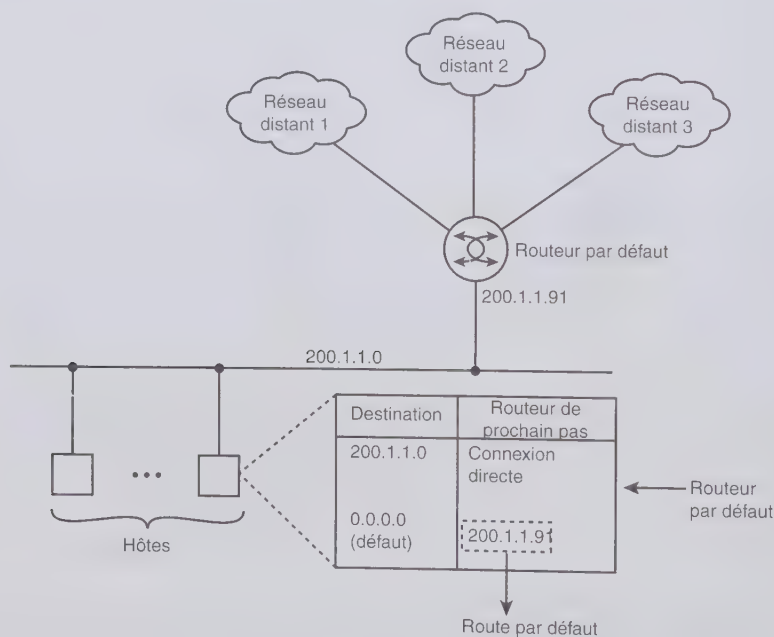


• Routes par défaut

Si un réseau ne dispose que d'un seul routeur de prochain pas, toutes les destinations distantes doivent être atteintes via ce routeur (voir Figure 7.11). Dans un tel cas de figure, il n'est pas nécessaire de spécifier explicitement chaque destination. Il suffit de spécifier une route par défaut, qui est la même pour toutes les destinations. Le routeur spécifié par cette route par défaut est appelé le *routeur par défaut*.

Figure 7.11

Un routeur de prochain pas unique.

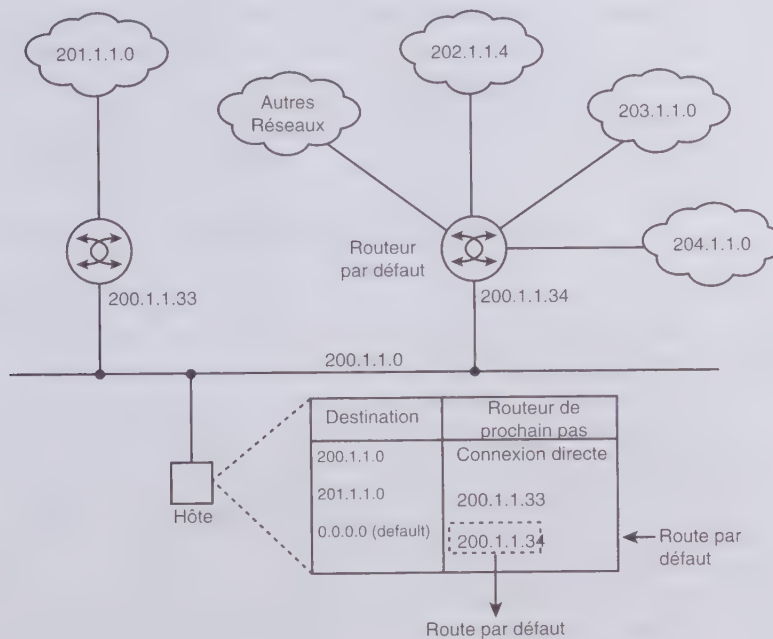


Dans la Figure 7.11, la table de routage des hôtes ne contient que deux entrées : une pour le réseau local et une autre pour la route par défaut. Le logiciel de routage n'a besoin d'examiner que deux choix pour déterminer la manière dont il convient de router le datagramme.

En général, la route par défaut permet de limiter la taille de la table de routage, dans les cas où il est nécessaire d'atteindre de nombreuses destinations par le biais du même routeur de prochain pas (voir Figure 7.12). Le logiciel de routage recherche dans la table une correspondance exacte pour la destination. S'il ne trouve rien, c'est la route par défaut qui est utilisée. La route par défaut est représentée dans la table de routage par la valeur de destination spéciale 0.0.0.0.

Figure 7.12

La route par défaut permet de réduire la taille de la table de routage.



Dans la Figure 7.12, la table de routage pour un hôte peut être écrite de la manière suivante, sans utiliser de routes par défaut :

Destination	Routeur de prochain pas
200.1.1.0	Connecté directement
201.1.1.0	200.1.1.33
202.1.1.0	200.1.1.34
203.1.1.0	200.1.1.34
204.1.1.0	200.1.1.34



Cette table de routage comporte cinq entrées. En employant une route par défaut, vous pouvez remplacer les trois dernières routes par une seule, comme le montre la table de routage équivalente ci-après :

Destination	Routeur de prochain pas
200.1.1.0	Connecté directement
201.1.1.0	200.1.1.33
0.0.0.0	200.1.1.34 <i>Route par défaut</i>

• L'algorithme de routage IP

Les exemples de routage que nous avons pu voir jusqu'ici vous ont sans doute éclairé sur certains aspects du processus de routage. Voici un algorithme qui décrit le processus dans son ensemble. L'algorithme de routage IP suit les étapes suivantes (il n'est pas fait référence ici aux masques de sous-réseau, sujet qui sera abordé dans les Chapitres 10 et 11).

1. On regarde AdIPDest, le champ d'adresse de destination du datagramme IP. A partir de ce champ AdIPDest, on détermine le SuffRes, le suffixe du réseau. Pour un réseau de classe A, SuffRes est le premier octet de AdIPDest. Pour un réseau de classe B, il s'agit des deux premiers octets de AdIPDest. Pour un réseau de classe C, il s'agit des trois premiers octets de AdIPDest.
2. Si SuffRes correspond à l'ID de réseau d'un des réseaux connectés directement, l'hôte de destination se trouve sur le réseau directement connecté. Dans ce cas, on effectue une remise directe du datagramme IP. La remise directe peut nécessiter la résolution de AdIPDest en son adresse physique correspondante, en employant un protocole tel que ARP. Le datagramme IP est encapsulé par la trame de couche de liaison de données et est envoyé vers le réseau directement connecté.
3. Si on ne trouve pas de correspondance à l'étape 2, on examine la table de routage à la recherche d'une entrée spécifique d'un hôte, correspondant à AdIPDest. Si une telle entrée existe, on transmet le datagramme IP, comme indiqué par l'entrée de routeur de prochain pas correspondante.
4. Si on ne trouve pas de correspondance à l'étape 3, on examine la table de routage à la recherche d'une entrée de réseau correspondant à SuffRes. Si une telle entrée existe, on transmet le datagramme IP de la manière indiquée par l'entrée de routeur de prochain pas correspondante.
5. Si on ne trouve pas de correspondance à l'étape 4, on regarde dans la table la valeur correspondant à 0.0.0.0, l'entrée par défaut. Si cette entrée existe, on

transmet le datagramme IP de la manière indiquée par l'entrée de routeur par défaut correspondante.

6. Si on ne trouve rien à l'étape 5, le datagramme IP ne peut être routé. On renvoie une erreur de routage aux protocoles de couche supérieure.

Traitement des datagrammes entrants

Jusqu'ici, nous n'avons parlé que des datagrammes sortant d'un hôte ou d'un routeur. Nous allons maintenant voir comment sont traités les datagrammes entrants. Lorsqu'un hôte ou un routeur reçoit un datagramme, une succession d'événements précis doit avoir lieu afin que le datagramme soit traité correctement.

L'hôte ou le routeur regarde l'adresse IP de destination dans le datagramme et la compare avec sa propre adresse IP. Si les adresses correspondent, le datagramme est accepté, et il est transmis aux protocoles de couche supérieure pour traitement. Dans le cas d'un hôte, les protocoles de couche supérieure sont les protocoles de transport et d'application qui traitent le datagramme. Dans le cas d'un routeur, le module de routage traite le datagramme entrant et le transmet. Il se peut que le datagramme ne soit pas destiné au module de routage, mais à un protocole de traitement de couche supérieure dans le routeur. C'est le cas pour des datagrammes envoyés au module de l'agent SNMP qui s'exécute sur le routeur. Le multiplexage de protocole qui a lieu au niveau des couches IP et de transport indique quel module d'application traitera le datagramme IP.

Si l'adresse IP de destination ne correspond pas, l'hôte doit écarter le datagramme IP. Par définition, un hôte ne peut accepter ni transmettre un datagramme qui ne lui est pas destiné. En revanche, un routeur est conçu pour transmettre des datagrammes qui ne leur sont pas directement adressés. Le routage de ces datagrammes se fait en fonction des informations figurant dans la table de routage du routeur.

Le processus consistant à déterminer une correspondance d'adresse IP est simple lorsque le nœud IP, un hôte par exemple, n'a qu'une interface de réseau. Il suffit de comparer l'adresse IP de destination à l'adresse IP d'une unique connexion de réseau.

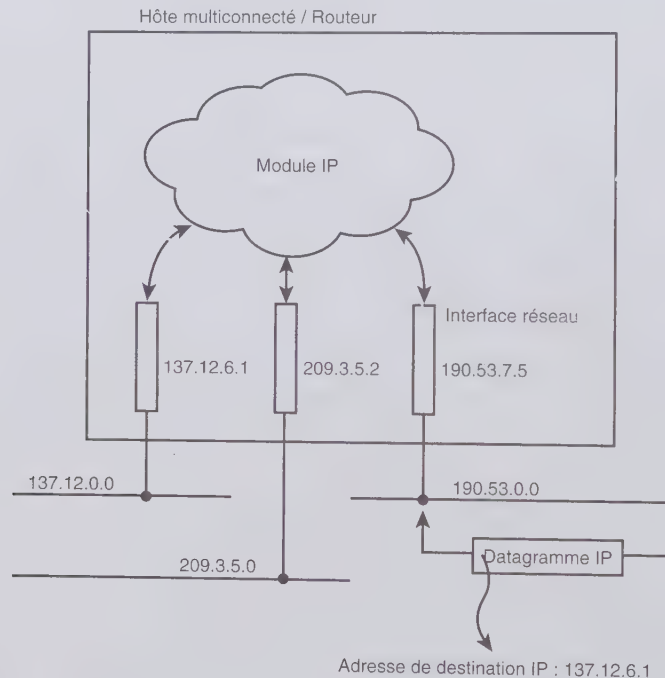
Les routeurs doivent posséder au moins deux connexions de réseau. Les hôtes multiconnectés peuvent aussi posséder plusieurs connexions de réseau. Si c'est le cas d'un nœud IP, la comparaison des adresse IP n'est pas aussi simple, car il est



nécessaire d'effectuer la comparaison avec les adresses IP de chacune des interfaces de réseau. En effet, un datagramme IP adressé à une des interfaces de réseau peut très bien être reçu par une autre interface de réseau du même nœud IP (voir Figure 7.13). Si l'adresse IP de destination indiquée dans le datagramme correspond à celle d'une des interfaces de réseau, le nœud IP conserve le datagramme et l'envoie au module approprié pour traitement.

Figure 7.13

Un datagramme IP adressé à une interface de réseau appartenant à un routeur ou à un hôte multiconnecté.



Le nœud IP doit aussi accepter les datagrammes reçus par diffusion (broadcast) sur n'importe laquelle de ces interfaces de réseau. Il peut s'agir d'une diffusion limitée (255.255.255.255) ou d'une diffusion dirigée vers un réseau ou un sous-réseau. Le nœud IP doit aussi accepter les multidiffusions (multicast), qui sont des adresses de classe D. Les adresses de diffusion et de multidiffusion sont décrites dans le Chapitre 4.

Si un routeur décide de transmettre un datagramme, il doit décrémenter le champ TTL figurant dans l'en-tête IP. Si la valeur de ce champ est nulle, le datagramme est écarté, et le routeur envoie un message ICMP informant l'émetteur de l'expiration du temporisateur TTL.

Après avoir déterminé l'interface de réseau vers laquelle il convient de transmettre le datagramme, le routeur détermine alors si le datagramme doit être fragmenté. On a vu qu'il y a fragmentation du datagramme si la taille du datagramme dépasse la taille de MTU de l'interface de réseau vers laquelle sera transmis le datagramme. Si le datagramme doit être fragmenté, les en-têtes IP de fragmentation que sont DF et MF, et le champ de décalage de fragmentation doivent être définis correctement. De plus, comme le champ TTL change après chaque traitement du datagramme par un routeur, le champ Total de contrôle doit être recalculé à chaque routeur.

Récapitulatif

IP offre une très grande souplesse pour la construction de réseaux de complexité arbitraire. On peut utiliser des périphériques de routage pour étendre la portée des réseaux. Les datagrammes sont remis à des destinations distantes en utilisant la méthode de la remise indirecte. La remise indirecte se décompose en une succession de remises directes, le datagramme étant transmis de routeur en routeur jusqu'à destination. Le routage a lieu aussi bien au niveau des hôtes que des routeurs.

les hôtes ne routent que les datagrammes sortants. Un hôte ne transmet pas de datagramme entrant à moins qu'il n'ait été spécifiquement configuré pour tenir lieu de routeur. Les routeurs, en revanche, sont conçus pour accepter et retransmettre les datagrammes entrants.

Hôtes comme routeurs utilisent des tables de routage. Une table de routage est constituée d'un ensemble d'entrées de routage, spécifiant la destination et le routeur de prochain pas vers lequel sera transmis le datagramme afin d'atteindre sa destination.



Testez vos connaissances

1. Dessinez un schéma illustrant le principe d'un routeur. Nommez-en les différentes parties et expliquez brièvement le fonctionnement du routeur.
2. Quelle est la différence entre le routage statique et le routage dynamique ?
3. Comment les routeurs tiennent-ils leurs tables à jour dans le cadre du routage dynamique ?
4. Qu'est-ce qu'un protocole de routage ? Citez quelques exemples de protocoles de routage.
5. Pourquoi est-il nécessaire de configurer un routeur avant de le faire fonctionner sur un réseau ?
6. Quelles sont les deux méthodes de remises de datagramme ?
7. Quand utilise-t-on la remise directe ?
8. Quand utilise-t-on la remise indirecte ?
9. Donnez un exemple où la remise indirecte d'un datagramme est constituée par une succession de remises directes.
10. Avant une remise directe dans un réseau à diffusion, quel protocole utilise-t-on généralement ?
11. Soit un hôte d'adresse IP 20.10.22.11 et une destination de 20.200.122.122.11. Y aura-t-il remise directe (en supposant que le réseau n'est pas mis en sous-réseau) ?
12. Soit un hôte d'adresse IP 134.100.122.111 et une destination de 134.100.222.1. Y aura-t-il remise directe (en supposant que le réseau n'est pas mis en sous-réseau) ?
13. Soit un hôte d'adresse IP 212.144.108.18 et une destination de 212.144.108.99. Y aura-t-il remise directe (en supposant que le réseau n'est pas mis en sous-réseau) ?
14. Soit un hôte d'adresse IP 144.144.108.18 et une destination de 144.145.108.18. Y aura-t-il remise directe (en supposant que le réseau n'est pas mis en sous-réseau) ?
15. Soit un hôte d'adresse IP 144.144.108.18 et une destination de 201.144.108.18. Y aura-t-il remise directe (en supposant que le réseau n'est pas mis en sous-réseau) ?
16. Les réseaux 202.34.56.0 et 202.34.57.0 sont des réseaux locaux Ethernet connectés par un routeur dont les deux interfaces de réseaux sont configurées avec les adresses IP 202.34.56.100 et 202.34.57.100. L'hôte 202.34.56.1 est connecté à 202.34.56.0 et envoie un datagramme IP à 202.34.57.1 à 202.34.57.0. En supposant que c'est la première fois que 202.34.56.1 tente de

communiquer en dehors du réseau, indiquer la succession de paquets que l'on devrait voir pour que le datagramme atteigne sa destination. En supposant que la valeur de TTL est de 64 secondes, quelle valeur de TTL aura le datagramme lorsqu'il atteindra sa destination ?

17. Pour cette question, on supposera que les adresses matérielles des interfaces de réseau sont les suivantes :

AM de 202.34.56.1 est 0000AA000033

AM de 202.34.56.100 est 00000C00ACB7

AM de 202.34.57.100 est 00000C00BCD9

AM de 202.34.57.1 est 0000AA0023C9

On suppose aussi que la valeur du champ Identification du datagramme est 33, celle du champ Protocole est 17 et que la valeur du total de contrôle est de 33DA (hexadécimal). Indiquez les valeurs d'en-têtes pour tous les paquets qui devraient apparaître sur le réseau.

18. La table de routage d'un hôte à 221.3.4.3 contient les entrées suivantes :

Destination	Routeur de prochain pas
221.3.4.0	Connecté directement
221.12.5.0	221.3.4.100
222.10.10.0	221.3.4.110
223.4.5.0	221.3.4.109
221.22.1.0	221.3.4.100

Pour chacune des destinations suivantes, spécifiez s'il est possible de router la destination à partir de l'hôte. Spécifiez aussi le routeur de prochain pas nécessaire pour atteindre la destination.

221.3.4.1

221.10.10.44

221.10.11.44

222.10.10.7

221.22.1.9

223.4.5.7

220.1.1.1



19. La table de routage d'un hôte à 100.3.4.3 contient les entrées suivantes :

Destination	Routeur de prochain pas
100.0.0.0	Connecté directement
22.0.0.0	100.3.5.9
222.0.44.0	100.45.22.224
134.6.0.0	100.56.45.66
199.22.1.0	100.99.99.99

Pour chacune des destinations suivantes, spécifiez s'il est possible de router la destination à partir de l'hôte. Spécifiez aussi le routeur de prochain pas nécessaire pour atteindre la destination.

221.3.4.1
221.10.10.44
100.66.85.66
199.22.1.9
222.10.10.7
221.22.1.9
222.0.44.44
22.55.44.56

20. Dans la table de routage suivante, suggérez une route par défaut qui permettrait de combiner certaines entrées et de réduire la taille de la table.

Destination	Routeur de prochain pas
210.1.1.0	Connecté directement
133.1.1.0	210.1.1.3
134.1.1.0	210.1.1.4
135.1.1.0	210.1.1.4
136.1.1.0	210.1.1.4

21. Quels sont les avantages ou inconvénients d'un hôte multiconnecté tenant lieu de routeur ?
22. Si un hôte multiconnecté qui n'est pas conçu comme un routeur reçoit un datagramme IP ne lui étant pas destiné, doit-il transmettre le datagramme s'il est capable de le router ? Votre réponse doit préciser les éventuels avantages ou inconvénients d'une telle méthode.
23. Dans quel cas utilise-t-on des routes spécifiques d'un hôte ? Quelle est l'incidence de leur utilisation sur la taille de la table de routage ?
24. Quels sont les avantages – s'il y en a – à utiliser une route par défaut ?



Le protocole ICMP

Le protocole IP offre un service de remise de datagramme sans connexion. IP ne tente pas d'assurer une remise fiable des datagrammes. Cette tâche incombe aux protocoles de couche supérieure tels que TCP. Cependant, IP dispose d'un moyen pour envoyer des messages d'alerte et de diagnostic par le biais du module ICMP (*Internet Control Message Protocol* ou "Protocole de message de contrôle Internet"). Ces messages peuvent permettre à l'administrateur de réseau de détecter des problèmes potentiels ou réels.

Présentation de ICMP

Comme le protocole IP a volontairement été conçu dans un souci de simplicité, il n'est pas rare que des erreurs surviennent lors de la transmission d'un datagramme IP. L'émetteur envoie le datagramme et s'en remet au réseau sous-jacent (couches OSI 3, 2 et 1) pour la remise du datagramme. L'émetteur n'a aucun moyen de savoir si des problèmes potentiels sont survenus lors de la transmission.

Les erreurs dans la transmission de datagrammes sont généralement détectées au niveau d'un routeur intermédiaire. Ainsi, par exemple, le dernier routeur de la chaîne des routeurs du chemin du datagramme peut se rendre compte que l'hôte de destination n'existe pas. Si le réseau de destination est un réseau local à diffusion, tel qu'un réseau Ethernet, le routeur s'aperçoit que l'hôte n'est pas disponible au moment où il utilise ARP pour trouver l'adresse matérielle de l'hôte de destination à partir de son adresse IP. Dans ce cas, le routeur tente d'en informer l'émetteur en envoyant un message ICMP qui indique que l'hôte de destination n'est pas disponible.

Parmi les autres problèmes susceptibles de survenir, on peut citer :

- l'expiration du champ TTL (durée de vie) du datagramme, à la suite d'une boucle dans le routage ;
- la non-remise d'un datagramme à la suite d'une perte d'un de ses fragments ;
- l'indisponibilité d'un protocole, d'un service ou d'un hôte ;



- l'impossibilité de transmettre un datagramme parce que la fragmentation est interdite et que la taille du datagramme est trop grande pour un des réseaux intermédiaires ;
- une congestion du trafic au niveau d'un routeur, empêchant celui-ci de continuer à traiter les datagrammes entrants.

Dans chacun de ces cas, on utilise ICMP pour avertir l'émetteur du problème. Le protocole IP lui-même ne contient pas de mécanisme permettant de prévenir le réseau. Les concepteurs de IP ont spécifié un protocole particulier qui, utilisé conjointement à IP, permet d'alerter le réseau sur l'existence d'erreurs, de tester le réseau et d'obtenir des informations le concernant. Ce protocole particulier est ICMP.

Toutes les implémentations de IP doivent prendre en charge ICMP, mais, malheureusement, le degré de cette prise en charge varie. Certaines implémentations de ICMP ne renvoient pas de message d'erreur pour des problèmes spécifiques, et d'autres génèrent un message erroné. Même si des messages ICMP sont générés, ils ne sont pas toujours pris en compte par l'implémentation IP de l'hôte qui les reçoit.

Toute implémentation IP peut générer un message ICMP, c'est en particulier le cas des hôtes et des routeurs. Les routeurs génèrent des messages d'erreur ICMP lorsqu'ils rencontrent un problème de transmission, tandis que les hôtes en génèrent en cas de problème de remise à une application ou à un protocole particuliers. Les messages ICMP ne concernent pas tous les erreurs. Certains servent spécifiquement pour tester un réseau ou obtenir des informations concernant ce dernier.

Voici les points les plus importants de ICMP :

- Toutes les implémentations de IP doivent implémenter ICMP.
- ICMP s'exécute au-dessus de IP. Autrement dit, ICMP est un "client" de IP, bien qu'il soit implémenté dans le module IP même.
- ICMP signale les conditions d'erreur. Il ne rend pas IP plus fiable.
- ICMP ne signale d'erreurs que pour le premier fragment d'un datagramme. Il s'agit du fragment de datagramme dont le champ Fragment Offset ("décalage de fragment") vaut 0. Cela évite que des messages ICMP soient envoyés pour chacun des fragments d'un même datagramme.
- ICMP est un mécanisme de signalisation d'erreur pour les datagrammes IP. ICMP ne génère aucun message d'erreur concernant des problèmes affectant des datagrammes ICMP.

Enfin, ICMP ne doit pas signaler les problèmes liés aux points suivants :

- **Routage ou remise des messages ICMP.** Si on générât des messages d'erreur ICMP pour les messages d'erreur ICMP, leur nombre croîtrait considérablement, ce qui augmenterait la charge du réseau.
- **Datagrammes IP de diffusion (*broadcast*) ou de multidiffusion (*multicast*).** Si on générât des messages d'erreur ICMP pour ces datagrammes, chaque nœud recevant le datagramme générerait un message ICMP, ce qui ajouterait encore à la charge du réseau.
- **Diffusion ou multidiffusion au niveau de la couche de liaison de données.** Si on générât des messages d'erreur ICMP pour les diffusions ou multidiffusions de couche de liaison de données, chaque nœud recevant le datagramme générerait un message ICMP, ce qui ajouterait encore à la charge du réseau.
- Des datagrammes dont les adresses source ne concernent pas un nœud IP unique (adresses de bouclage 127.x.x.x ou 0.0.0.0, ou une adresse dont le préfixe de réseau vaut 0, par exemple). Le préfixe de réseau est constitué des bits de l'adresse IP qui identifient le réseau de manière non ambiguë.

Les sections qui suivent analysent en détail ICMP.

• RFC traitant de ICMP

Les informations concernant ICMP sont ventilées sur plusieurs RFC.

ICMP est un protocole standard (STD 5), et on le trouve documenté dans la RFC 792, la RFC 950 parlant des extensions relatives aux sous-réseaux. La RFC 1812 contient des informations concernant les types de messages ICMP 13, 14 et 15. La RFC 1122 contient des conseils sur le traitement par les hôtes des messages ICMP. La RFC 1256 contient des informations concernant les types ICMP 9 et 10 permettant d'implémenter le protocole de découverte de routeur. La RFC 1191 contient des extensions aux messages de type 4.

• Détection d'erreur par ICMP

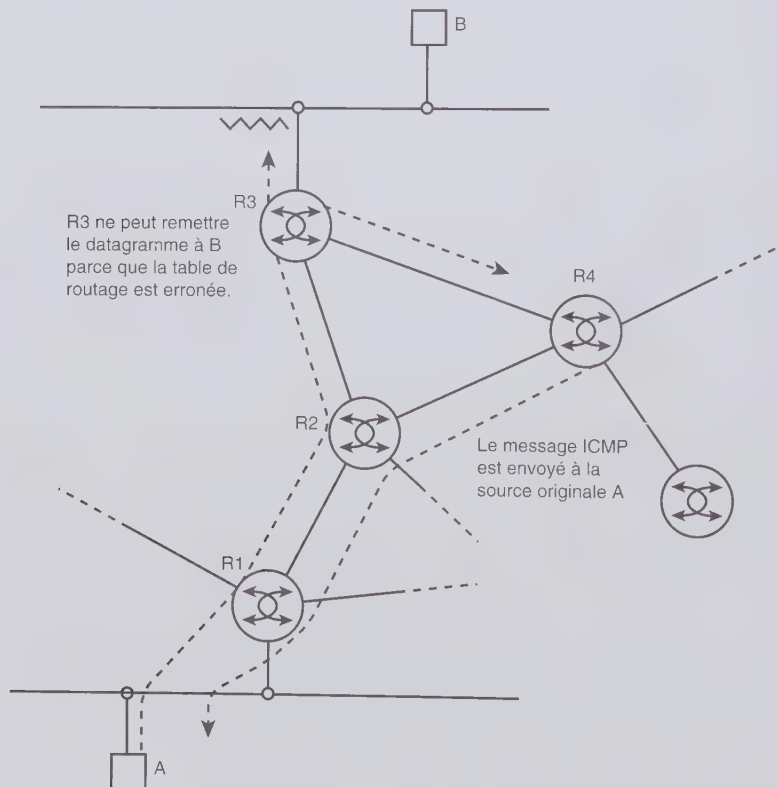
Comme nous venons de le voir, ICMP est un mécanisme signalant les erreurs concernant les datagrammes IP. Les messages d'erreur ICMP concernant un datagramme IP sont envoyés à l'émetteur du datagramme. Cependant, si les spécifications ICMP indiquent à quoi sert le datagramme ICMP, elles ne précisent pas complètement la réaction à avoir face à chacun de ces messages. Il incombe au destinataire du message ICMP d'agir de la manière idoine.



Les messages ICMP sont envoyés à l'émetteur du datagramme IP. On suppose donc que l'émetteur saura comment traiter le problème. C'est le cas la plupart du temps, mais pas toujours. Prenons l'exemple de la Figure 8.1, où un datagramme envoyé par la source A à la destination B suit un chemin incluant les routeurs R1, R2 et R3.

Figure 8.1

Un exemple de message ICMP.



Supposons que la table de routage du routeur R3 contienne des informations erronées concernant la destination, et que, selon ces informations, le datagramme doit être routé vers R4 plutôt que vers B. Dans ce cas, le datagramme sera transmis à R4 plutôt qu'à la destination B. R4 peut envoyer un message ICMP signalant la route incorrecte, mais il ne peut envoyer ce message à R3 qui est la source du problème. Il l'envoie à la source A, qui n'est pas responsable du problème et n'est pas responsable de R4. La plupart du temps, la source, lorsqu'elle reçoit le message ICMP, ne peut pas déterminer quel est le routeur qui a provoqué le problème. Pourquoi ICMP se borne-t-il à envoyer des messages à la source

originelle ? Cela s'explique par le fait que l'en-tête du datagramme IP ne contient que deux champs d'adresses : l'adresse IP source originelle et l'adresse IP de destination finale. Les adresses IP des routeurs intermédiaires ne sont pas enregistrées dans l'en-tête, à moins que l'option IP Record Route ne soit activée par la source. La plupart des datagrammes sont envoyés sans option IP. On évite de faire un trop grand usage de l'option Record Route, car cette dernière nécessite un traitement supplémentaire pour l'enregistrement des adresses intermédiaires et augmente la taille de l'en-tête.

Lorsque le routeur R4 souhaite renvoyer le message ICMP, il n'a connaissance que des adresses IP de la source originelle et de la destination. R4 ne sait rien des routeurs intermédiaires qui ont traité le datagramme et ne peut donc leur envoyer de message ICMP. Il serait absurde pour R4 d'envoyer le message ICMP à la destination, puisque le datagramme initial a sans doute eu un problème pour atteindre cette destination. De plus, la destination n'est pas la cause du problème. La faute en revient à la source ou à l'un des routeurs intermédiaires. Donc, le seul choix raisonnable consiste à envoyer le message ICMP à la source.

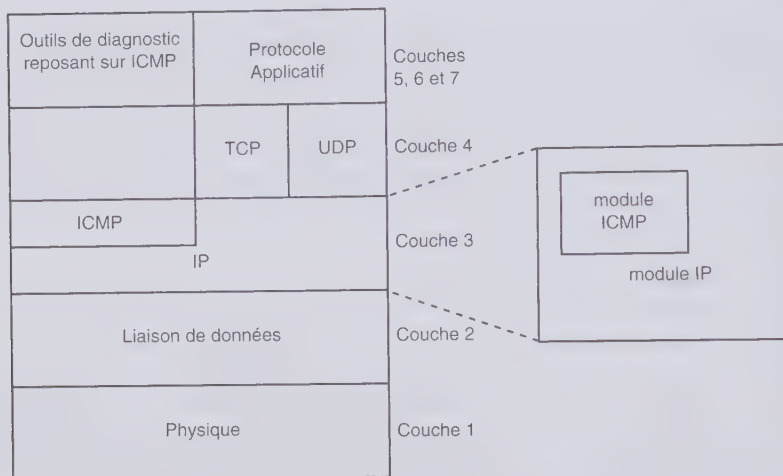
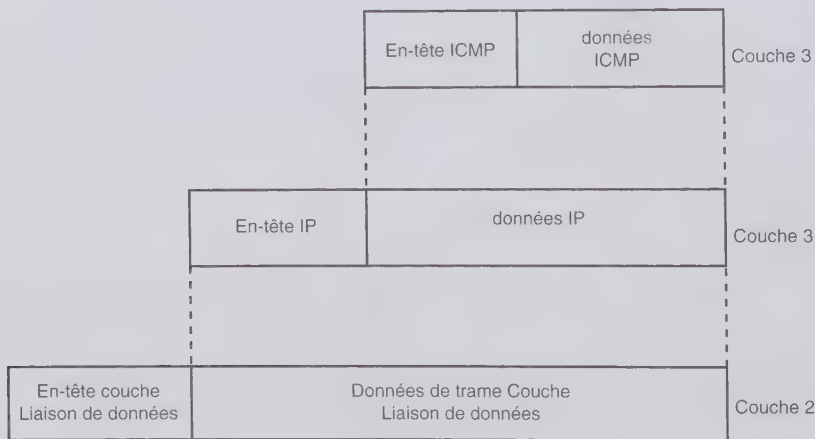
L'envoi d'un message ICMP à la source est de toute façon préférable à l'absence de tout message. Sans message ICMP, l'administrateur de réseau ne dispose pas d'informations suffisantes concernant la cause du problème. Le message ICMP peut permettre à un administrateur de réseau avisé de localiser et de corriger le problème.

• Encapsulation ICMP

Comme les messages ICMP doivent être transportés sur l'interréseau et de routeur en routeur, ils doivent être encapsulés par IP. En effet, on a vu que sans IP, les données de protocoles de couche supérieure ne peuvent franchir la frontière du routeur. Cela signifie donc que le message ICMP est inclus dans la partie données du datagramme IP. L'en-tête IP de ce datagramme a une valeur de champ Protocole de 1, valeur réservée pour ICMP.

Le module IP qui reçoit le paquet démultiplexe les données du datagramme selon la valeur du champ Protocole et envoie les données au module ICMP. La Figure 8.2 montre la structure de couche de IP et ICMP. Elle indique aussi que le module ICMP est implémenté comme partie du module IP. La Figure 8.3 montre l'encapsulation de ICMP dans un en-tête IP.



Figure 8.2*Relation entre ICMP et IP.***Figure 8.3***Encapsulation ICMP.*

La Figure 8.2 montre aussi que bien que le message ICMP soit encapsulé par IP, on ne considère pas ICMP comme un protocole de couche supérieure. ICMP est une partie nécessaire de IP. Il est possible de créer des applications, des utilitaires et des outils de diagnostic spécialisés utilisant ICMP. Ces applications spécialisées peuvent être considérées comme faisant partie de la couche d'application.

Un exemple d'un tel outil de diagnostic spécialisé utilisant ICMP est l'utilitaire PING, disponible sur la plupart des systèmes TCP/IP. L'utilitaire PING permet d'envoyer un message de requête d'écho à une destination. Si le message de requête atteint un module IP distant, ce module doit renvoyer une réponse d'écho.

La réception de la requête d'écho indique que le nœud IP distant est accessible jusqu'à la couche IP au moins. Nous verrons plus en détail les messages ICMP permettant d'implémenter l'utilitaire PING un peu plus loin dans ce chapitre. Dans certaines implémentations IP, l'accès au module ICMP n'est pas disponible pour les applications. Dans ce cas, les applications de type PING émulent le comportement de ICMP en générant eux-mêmes des messages ICMP.

L'en-tête IP utilisé pour transporter un message ICMP a la structure suivante :

Version	4
Longueur d'en-tête (mots de 32 bits)	Généralement cinq, à moins que des options IP ne soient spécifiées.
TOS	0 (Préséance : routine, retard normal, débit normal, fiabilité normale).
Longueur totale	Dépend de la taille du message ICMP.
Identification	Dépend des datagrammes précédemment envoyés.
Flags	Activé pour indiquer une éventuelle fragmentation. Généralement, les datagrammes contenant des messages ICMP sont courts, et leur fragmentation est donc rarement nécessaire.
Décalage de fragment	Dépend du flag de fragmentation, généralement 0.
TTL	Dépend de l'implémentation. Doit être assez élevé pour permettre au datagramme d'atteindre sa destination.
Protocole	1 (valeur indiquant un message ICMP).
Total de contrôle	Dépend des données du message.
IP source	L'adresse IP du routeur ou de l'hôte qui compose le message ICMP. Cette adresse peut être celle de n'importe lequel des ports du routeur.
IP destination	L'adresse IP du routeur ou de l'hôte de destination du message.

Vous pouvez remarquer que l'en-tête IP n'indique aucune priorité spéciale pour la remise de messages ICMP. Les datagrammes IP contenant des messages ICMP sont traités par les nœuds IP (hôtes ou routeurs) comme tout autre datagramme.



Comme IP ne garantit pas la remise des messages, il se peut que les messages ICMP se perdent ou soient écartés du fait de congestion du réseau. De plus, dans un réseau déjà congestionné, la génération de messages ICMP signalant la congestion ne fait qu'ajouter au problème.

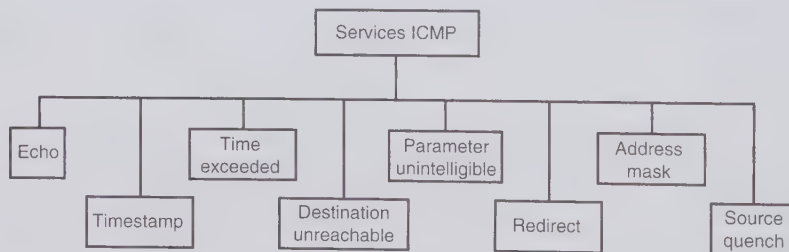
Que se passe-t-il si le datagramme qui transporte un message ICMP rencontre une erreur ? Un message ICMP indiquant une erreur sur le datagramme transportant un message ICMP doit-il être généré ? La réponse est non : aucun message ICMP n'est généré pour des erreurs concernant le traitement de datagrammes IP qui contiennent des messages ICMP. De tels messages risqueraient d'affecter le trafic du réseau et n'auraient de toute façon pas grand intérêt pour l'administrateur.

Services ICMP

ICMP fournit une gamme assez large de services de messages d'erreur, d'information et de diagnostic (voir Figure 8.4).

Figure 8.4

*Les services ICMP
(source : Learning Tree).*



- ICMP transporte les erreurs dans des datagrammes IP
- ICMP ne rend pas IP plus fiable

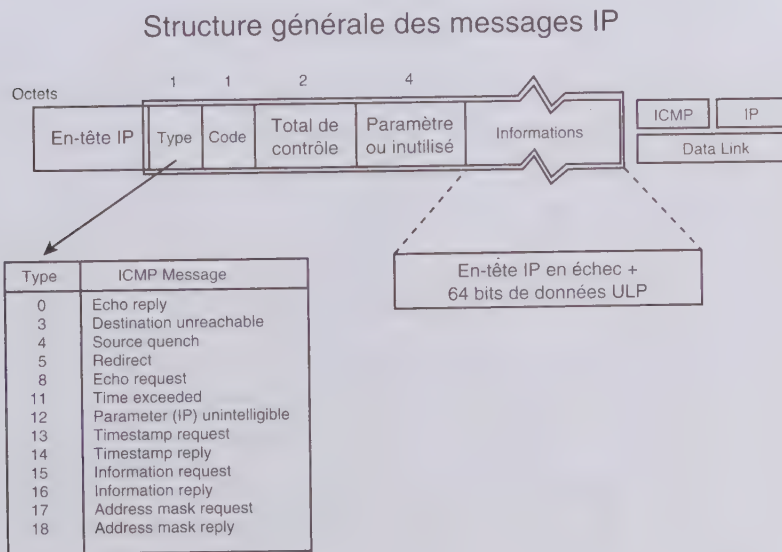
Les principaux services ICMP sont les suivants :

- **Echo.** Utilisé comme outil de diagnostic pour déterminer si un nœud IP est accessible.
- **Destination unreachable** ("destination inaccessible"). Indique qu'un nœud IP de destination n'est pas accessible.
- **Source Quench.** Indique un problème de congestion à la source.
- **Redirect.** Utilisé par les routeurs pour indiquer une route de substitution.
- **Time exceeded** ("temps dépassé"). Utilisé pour indiquer l'expiration de la valeur du champ TTL de l'en-tête IP.

- **Parameter problem** ("problème de paramètre"). Indique un problème dans un datagramme IP.
- **Timestamp** ("marqueur de temps"). Utilisé pour la mesure du temps sur un interréseau.
- **Address mask** ("masque d'adresse"). Permet d'obtenir des informations de masque de sous-réseau pour le réseau.

Quel que soit le message ICMP, le format général reste à peu de choses près le même (voir Figure 8.5). Le premier octet de l'en-tête ICMP est le champ Type, indiquant le type du service ICMP. L'octet suivant contient une valeur Code décrivant plus en détail la nature du type de message. Il faut considérer à la fois les valeurs des champs Type et Code pour déterminer la teneur exacte du message ICMP.

Figure 8.5
Structure générale
d'un message ICMP
(source : Learning Tree).



Le Tableau 8.1 dresse la liste des valeurs de type pour ICMP, et le Tableau 8.2 les valeurs de code pour les types correspondants. Certains types sont réservés pour des protocoles expérimentaux. Leurs codes ne figurent alors pas sur la liste, car ils sont susceptibles de changer.



Tableau 8.1 : Valeurs des champs Type ICMP

Valeur de Type	Signification
0	Echo Reply
1	Unassigned
2	Unassigned
3	Destination Unreachable
4	Source Quench
5	Redirect
6	Alternate Host Address
7	Unassigned
8	Echo
9	Router Advertisement
10	Router Selection
11	Time Exceeded
12	Parameter Problem
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply
19	Reserved (for Security)
20-29	Reserved (for Robustness Experiment)
30	Traceroute
31	Datagram Conversion Error
32	Mobile Host Redirect
33	IPv6 Where-Are-You
34	IPv6 I-Am-Here
35	Mobile Registration Request
36	Mobile Registration Reply
37-255	Reserved

Tableau 8.2 : Valeurs des champs Code ICMP

Type	Codes	Signification
0 (Echo Reply)	0	No Code
1 (Unassigned)		
2 (Unassigned)		
3 (Destination Unreachable)	0	Net Unreachable
	1	Host Unreachable
	2	Protocol Unreachable
	3	Port Unreachable
	4	Fragmentation Needed and Don't Fragment was Set
	5	Source Route Failed
	6	Destination Network Unknown
	7	Destination Host Unknown
	8	Source Host Isolated
	9	Communication with Destination Network is Administratively Prohibited
	10	Communication with Destination Host is Administratively Prohibited
	11	Destination Network Unreachable for Type of Service
	12	Destination Host Unreachable for Type of Service
4 (Source Quench)	0	No Code
5 (Redirect)	0	Redirect Datagram for the Network (or subnet)
	1	Redirect Datagram for the Host
	2	Redirect Datagram for the Type of Service and Network
	3	Redirect Datagram for the Type of Service and Host
6 (Alternate)	0	Alternate Address for Host Host Address
7 (Unassigned)		
8 (Echo)	0	No Code



Tableau 8.2 : Valeurs des champs Code ICMP (*suite*)

Type	Codes	Signification
9 (Router	0	No Code Advertisement)
10 (Router	0	No Code Selection)
11 (Time Exceeded)	0	Time to Live exceeded in Transit
	1	Fragment Reassembly Time Exceeded
12 (Parameter Problem)	0	Pointer indicates the error
	1	Missing a Required Option
	2	Bad Length
13 (Timestamp)	0	No Code
14 (Timestamp Reply)	0	No Code
15 (Information Request)	0	No Code
16 (Information Reply)	0	No Code
17 (Address Mask 0 Request)		No Code
18 (Address Mask 0 Reply)		No Code
19 (Reserved for Security)		

Les deux octets suivants constituent un champ Total de contrôle. Ce total de contrôle ne concerne que le message ICMP et n'inclut pas l'en-tête IP du datagramme encapsulé. L'algorithme de ce total de contrôle est identique à celui qui est utilisé pour IP (voir Chapitre 7). Une implémentation peut définir le total de contrôle ICMP à 0 pour indiquer que cette valeur n'est pas calculée. Cela tient compte du fait que l'en-tête IP contient aussi un total de contrôle prenant en compte le message ICMP, puisque ce dernier est encapsulé dans le datagramme IP. De plus, la trame de couche de liaison de données peut aussi fournir un total de contrôle. En conséquence, le total de contrôle ICMP est une précaution supplémentaire qui ne fait que s'ajouter aux mécanismes existant déjà dans le datagramme IP ou la trame de couche de liaison de données.

Les 4 octets suivants spécifient des paramètres facultatifs. Ces paramètres dépendent du message ICMP. Ce champ est facultatif et peut ne pas figurer dans certains types de messages ICMP. Le champ Information, utilisé par les messages ICMP signalant des erreurs, inclut l'en-tête IP, auquel s'ajoutent 64 bits de données de protocole de couche supérieure (ULP, *Upper Layer Protocol*) appartenant au datagramme posant problème.

Cette inclusion est des plus utiles, car elle permet de déterminer quel est le datagramme qui a posé problème. L'en-tête IP est inclus afin de déterminer les adresses IP de l'émetteur et de la destination. Les 64 bits (8 octets) de données de couche supérieure incluent une partie de l'en-tête TCP ou la totalité de l'en-tête UDP.

Nous verrons plus tard dans cet ouvrage qu'un en-tête TCP a une taille minimale de 20 octets. Comme on ne fait figurer que 8 octets de cet en-tête dans le message ICMP, on ne retrouve dans ce dernier qu'une partie de l'en-tête. Néanmoins, les 8 premiers octets de l'en-tête TCP contiennent les numéros de port de source et de destination et le champ de numéro de séquence. Les numéros de port permettent de déterminer quelles étaient les applications impliquées dans la session TCP. En revanche, si on a utilisé UDP comme protocole de couche de transport, l'en-tête UDP tout entier est renvoyé, car sa taille est de 8 octets.

Les différents messages ICMP sont décrits dans les sections qui suivent.

• Types ICMP 0 et 8 : Echo/PING

Les messages ICMP les plus utilisés sont sans doute les types Echo 0 et 8. Ces messages sont utilisés pour le diagnostic par l'utilitaire PING (*Packet Internet Groper*) qui est disponible sur la plupart des systèmes TCP/IP. PING permet de vérifier qu'un nœud IP est "vivant" et accessible. PING envoie un message ICMP de type Requête d'écho (8) à un nœud IP. A la réception de ce message, le nœud IP renvoie un message ICMP de Réponse d'écho (0). Le message de réponse d'écho contient une copie des données envoyées dans le message de requête d'écho. La réception d'un message de réponse d'écho confirme que la couche IP (et les couches inférieures) du système de transport fonctionne bien entre la source et la destination. Si des routeurs intermédiaires sont rencontrés entre la source et la destination, le message de réponse d'écho indique aussi que les routeurs sont en mesure de transmettre des datagrammes entre la source et la destination.

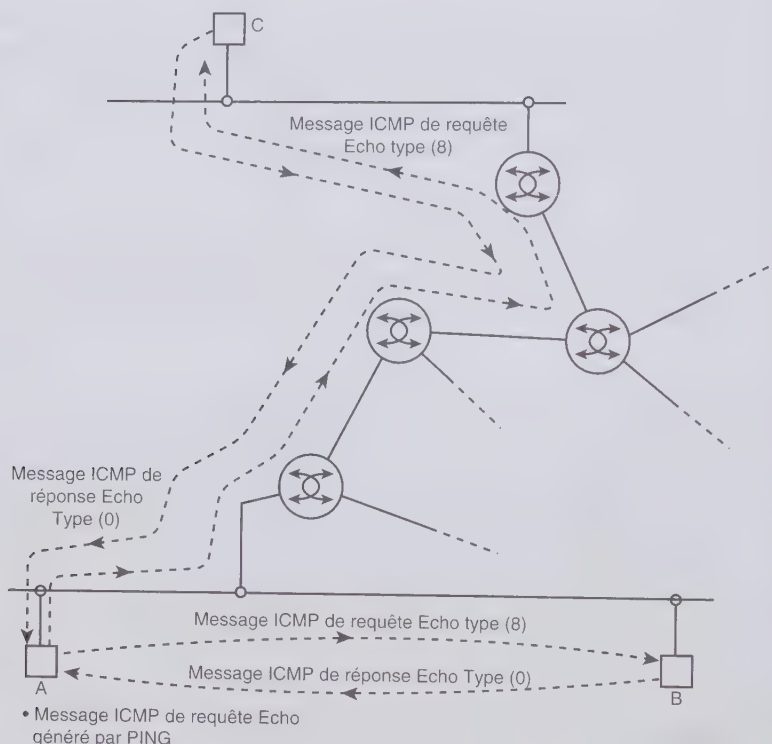
Le nœud IP est généralement un hôte ou un routeur. La Figure 8.6 montre un exemple d'utilisation de l'utilitaire PING.

• 337



Figure 8.6

Utilisation de messages Requête d'écho/Réponse d'écho.



Cette figure montre une situation où PING permet de tester l'accessibilité d'hôtes locaux et distants. Dans la Figure 8.6, l'hôte B est local pour l'hôte A, et l'hôte C est distant pour ce même hôte A. Les messages ICMP envoyés entre A et C sont traités comme des datagrammes IP normaux par les routeurs intermédiaires. Si des messages ICMP sont perdus, aucune tentative de renvoi du message n'est effectuée.

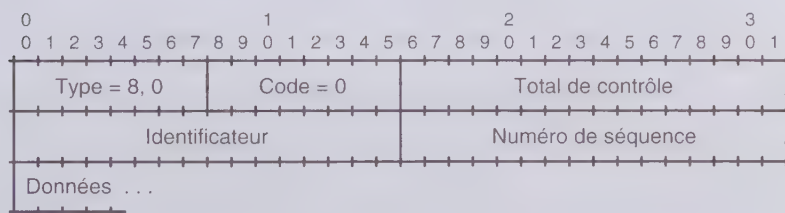
L'utilitaire PING est utilisé pour générer le message de Requête d'écho ICMP. Le module IP qui s'exécute sur le nœud de destination répond automatiquement aux messages ICMP. Il n'est pas nécessaire d'exécuter un logiciel spécialisé au niveau des nœuds de destination pour générer le message ICMP de réponse d'écho. Autrement dit, tous les nœuds IP d'un interrèseau sont capables de répondre aux messages ICMP de requête d'écho générés par PING.

Il n'existe pas de standard codifiant les options et fonctionnalités de PING, mais la plupart des implémentations proposent des services similaires. La syntaxe permettant d'invoquer PING varie selon le système d'exploitation, mais elle est souvent similaire d'un système à un autre, en particulier sur les différents systèmes

d'exploitation Unix. La plupart des implémentations offrent une version en ligne de commande de PING, appelée *ping*, mais il existe aussi des implémentations graphiques. La plupart des implémentations fournissent une estimation du temps d'aller-retour, mesuré en secondes, ainsi que des statistiques concernant la perte de datagrammes. D'autres options existent aussi. Ainsi, vous pouvez spécifier la taille des données envoyées dans chaque message de requête d'écho et déterminer la fréquence d'envoi des requêtes. Certains PING permettent de spécifier des options IP, telles que le routage par la source, qui permet d'imposer le chemin suivi par le message ICMP.

La Figure 8.7 montre le format des messages ICMP de requête d'écho et de réponse d'écho.

Figure 8.7
Format ICMP Requête d'écho/
réponse d'écho.



Le champ Type a comme valeur 8 pour la requête d'écho et 0 pour la réponse d'écho. Le champ code vaut 0 pour les deux messages. Le champ Total de contrôle est commun à tous les types de messages ICMP et a été décrit précédemment.

Les champs Identificateur et Numéro de séquence servent à l'identification non ambiguë des messages de requête et de réponse d'écho, ce qui permet de les faire correspondre deux à deux sans risque de confusion. Cette correspondance est nécessaire, car le réseau IP ne garantit pas que les messages de réponse d'écho seront envoyés dans le bon ordre.

Le champ Données permet d'envoyer des messages de taille arbitraire. Cela est utile pour tester la capacité d'un réseau à transporter de gros datagrammes. Il arrive ainsi qu'un routeur ou qu'un hôte puisse traiter sans problème des datagrammes de petite taille et soit incapable de traiter des datagrammes plus importants. Ce genre de problème peut être détecté en utilisant des champs de données PING de grande taille.

Le message de réponse d'écho renvoie les données reçues dans la requête d'écho.



• Type ICMP 3 : Destination inaccessible

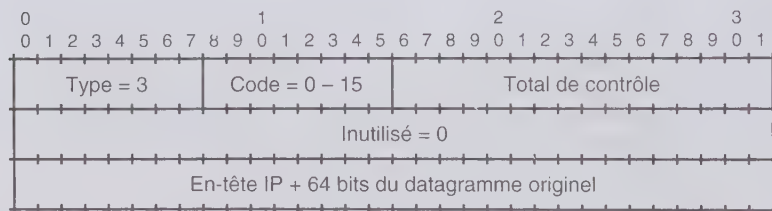
Le réseau IP fournit un service de *remise au mieux* et n'est pas toujours en mesure de remettre le datagramme IP. La non-remise d'un datagramme est généralement détectée par les routeurs, le plus souvent par le dernier routeur de la chaîne des routeurs du chemin emprunté par le datagramme.

Lorsqu'un routeur est incapable de remettre un datagramme, il en avertit l'émetteur au moyen d'un message ICMP de destination inaccessible. Les protocoles de couche supérieure, tels que TCP, transmettent le message perdu sans connaître la cause de l'échec de la remise. Le message ICMP permet dans certains cas d'indiquer à l'émetteur cette cause.

La Figure 8.8 montre le format du message ICMP de destination inaccessible.

Figure 8.8

Message ICMP Destination inaccessible.



Le champ Type vaut 3 et indique qu'il s'agit d'un message ICMP de destination inaccessible. Le champ Code précise *pourquoi* la destination est inaccessible. Les valeurs du champ Code sont détaillées dans le Tableau 8.3.

Tableau 8.3 : Codes pour le message Destination inaccessible

Code	Signification
0	Net Unreachable
1	Host Unreachable
2	Protocol Unreachable
3	Port Unreachable
4	Fragmentation Needed and Don't Fragment was Set
5	Source Route Failed
6	Destination Network Unknown
7	Destination Host Unknown
8	Source Host Isolated

Tableau 8.3 : Codes pour le message Destination inaccessible (suite)

Code	Signification
9	Communication with Destination Network is Administratively Prohibited
10	Communication with Destination Host is Administratively Prohibited
11	Destination Network Unreachable for Type of Service
12	Destination Host Unreachable for Type of Service
13	Communication Administratively prohibited because of firewalls
14	Host Precedence Violation
15	Precedence cut-off in effect

Les sections qui suivent traitent de ces codes plus en détail.

Type ICMP 3, Code 0 : Réseau inaccessible

La valeur de code 0 (réseau inaccessible) est générée lorsque le réseau spécifié dans l'adresse IP de destination est introuvable. Ce message n'est généré que par les routeurs. La cause peut être une erreur dans l'adresse IP de destination ou dans la table de routage du routeur générant le message ICMP. L'adresse source contenue dans l'en-tête IP transportant ce message ICMP identifie le routeur ayant généré l'erreur. L'étape suivante consiste à examiner les tables de routages du routeur concerné.

Type ICMP 3, Code 1 : Hôte inaccessible

La valeur de code 1 (hôte inaccessible) est générée par le routeur qui est directement connecté au réseau de destination. Elle indique que le datagramme a été remis avec succès par les routeurs, mais que le dernier routeur ne peut communiquer avec l'hôte. Cela peut venir du fait que le routeur n'est pas parvenu à déterminer l'adresse matérielle de destination à l'aide du protocole ARP. L'hôte peut être inaccessible parce qu'il est hors-service, mal configuré, ou parce qu'une adresse IP incorrecte a été spécifiée. L'adresse source contenue dans l'en-tête IP transportant ce message ICMP identifie le routeur qui a généré l'erreur. Les messages d'erreur ICMP Hôte inaccessible impliquent un échec dans la remise, tandis que les messages Réseau inaccessible impliquent un échec lors du routage.



Type ICMP 3, Code 2 : Protocole inaccessible

Le code 2 (Protocole inaccessible) est généré par l'hôte de destination. Ce message indique que le datagramme a bien atteint l'hôte de destination, mais que le protocole de couche supérieure, transporté par le datagramme IP et identifié par le champ Protocole de l'en-tête IP, n'est pas disponible. Les protocoles transportés par IP figurent dans le Tableau 6.5 du Chapitre 6. Les protocoles de couche supérieure les plus courants sur les hôtes TCP/IP sont TCP, UDP et OSPF. L'écrasante majorité des hôtes TCP/IP prennent au moins en charge TCP et UDP. Un message ICMP indiquant qu'un tel protocole est inaccessible est signe d'un grave problème de configuration au niveau de l'hôte. Les autres protocoles sont plus souvent inaccessibles, et le message ICMP Protocole inaccessible signale ce fait à l'émetteur.

La plupart des systèmes TCP/IP mettent en place des données spécifiques pour la configuration – c'est le cas du fichier `/etc/protocols` des systèmes Unix – spécifiant les protocoles pris en charge par IP. Ces données de configuration doivent être consultées au niveau de l'hôte de destination pour vérifier s'ils sont en accord avec le message Protocole inaccessible.

Type ICMP 3, Code 3 : Port inaccessible

Le code 3 (Port inaccessible) est généré lorsque le protocole de couche de transport indiqué (TCP ou UDP) est incapable de démultiplexer le datagramme et ne dispose pas d'autre mécanisme de protocole permettant d'en informer l'émetteur. Il se peut que le protocole de couche de transport ne soit pas en mesure de démultiplexer le datagramme du fait de l'absence d'un service d'application spécifique au niveau de l'hôte de destination.

Les services d'application sont identifiés par des numéros de port TCP ou UDP. Sur la plupart des systèmes, les services d'application sont lancés après lecture de fichiers de configuration dédiés – tels que le fichier `/etc/inetd.conf` sous Unix – lors de l'initialisation du système TCP/IP.

Si un message Type 3 Code 3 est reçu, cela indique que le service d'application désiré n'est pas disponible. Il convient alors de consulter les données de configuration des services d'application pour voir si le service concerné y figure. On peut aussi examiner le mécanisme d'initialisation et de lancement du service d'application au niveau de l'hôte de destination. Ainsi, par exemple, la plupart des systèmes, et Unix en particulier, emploient le concept de processus de serveur (appelé le superdémon *inetd*), qui écoute les requêtes arrivant sur les numéros de port de service d'application. Si le processus de serveur n'est pas activé, il se peut que les services d'application soient inaccessibles.

Type ICMP 3, Code 4 : Fragmentation nécessaire et flag Ne pas fragmenter activé

Le code 4 (Fragmentation nécessaire et le flag Ne pas fragmenter est activé) est généré par les routeurs qui doivent fragmenter le datagramme parce que la taille de MTU de l'interface de réseau par laquelle doit transiter le datagramme est inférieure à celle du datagramme. Si dans l'en-tête IP du datagramme, le flag Ne pas fragmenter (DF) est mis à 1, le routeur ne peut fragmenter le datagramme. Dans un tel cas de figure, le routeur ne peut transmettre le datagramme et doit donc l'écarter. Il envoie alors à l'émetteur initial un message ICMP de type 3 signalant que le datagramme n'a pu être envoyé, le code 4 en indiquant la raison.

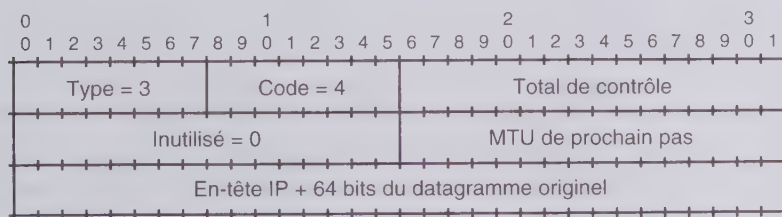
INFO



Le flag DF est mis à 1 par les stations de travail sans disque, qui effectuent un transfert de fichier TFTP pour télécharger l'image de démarrage système. TFTP envoie les données par blocs de 512 octets. En ajoutant la taille de l'en-tête IP (20 octets), celle de l'en-tête UDP (8 octets) et celle des données propres au protocole de transport utilisé par TFTP, la taille du message est en moyenne de 540 octets. Selon la RFC 791 concernant IP, tous les hôtes et routeurs doivent accepter des datagrammes IP de taille inférieure à 576 octets. De ce fait, le datagramme IP TFTP n'aura jamais besoin d'être fragmenté, à moins que l'hôte-routeur ne respecte pas la taille minimale de 576 octets ou que la taille de MTU d'une interface de réseau soit inférieure à 540 octets.

La RFC 1191 décrit une légère modification dans la forme des messages Destination inaccessible de code 4. La RFC 1812 affirme qu'un routeur doit utiliser cette forme modifiée lorsqu'il émet des messages Destination inaccessible de code 4. La Figure 8.9 décrit la forme modifiée de ce message.

Figure 8.9
Message Type 3, Code 4
modifié selon la RFC 1812.



Le format code 4 modifié est nécessaire pour la prise en charge de la technique de découverte de MTU du chemin spécifiée dans la RFC 1191. Le routeur doit inclure la taille de MTU de ce réseau de prochain pas dans les 16 bits de poids faible du



champ d'en-tête ICMP baptisé "inutilisé" dans la Figure 8.8. Les 16 bits de poids fort restent inutilisés et doivent être mis à zéro.

La valeur indiquée dans le champ MTU de prochain pas correspond à la taille en octets du plus grand datagramme pouvant être transmis le long du chemin du datagramme initial, sans qu'une fragmentation ne survienne au niveau de ce routeur. La taille inclut l'en-tête IP et les données IP, mais pas les en-têtes de couches inférieures, telles que la couche de liaison de données ou la couche physique.

Type ICMP 3, Code 5 : Echec du routage par la source

Le code 5 (échec du routage par la source) est généré par un routeur. Il n'est généré que pour des datagrammes IP utilisant l'option IP Source Route (voir Chapitre 6). Cette option permet de spécifier le chemin complet des routeurs utilisés pour la transmission du datagramme.

Si un routeur n'est pas en mesure de transmettre le datagramme en respectant le chemin de routage par la source, il doit écarter le datagramme, puis envoyer à l'émetteur un message ICMP type 3 code 5, indiquant que le routage par la source a échoué. On peut déterminer l'identité du routeur qui a généré ce message en examinant le champ d'adresse source de l'en-tête IP du message ICMP.

Le message ICMP Echec routage par la source doit contenir une option Source Route de même type (strict ou loose), créée en prenant dans l'option Source Route du datagramme initial la portion du champ d'option située avant le pointeur de route enregistrée, et en l'inversant. La seule occasion où les champs de routage par la source du datagramme fautif ne sont pas utilisés se rencontre lorsqu'un message ICMP Problème de paramètre concernant un routage par la source est envoyé, ou lorsque le routeur a connaissance de raisons qui empêcheraient la remise de message ICMP utilisant les informations de routage par la source du datagramme initial.

Type ICMP 3, Code 6 : Réseau de destination inconnu

Le code 6 (Réseau de destination inconnu) était supposé être généré par un routeur lorsque celui-ci aurait détecté, en consultant sa table de routage, un réseau de destination inconnu. En fait, ce message ne sera jamais généré sur un réseau réel, car le routeur préférera générer un message Type 3 code 0 (Réseau inaccessible). Autrement dit, le message ICMP Type 3, code 6 est obsolète.

Type 3 Code 7 : Hôte de destination inconnu

Le code 7 (hôte de destination inconnu) est généré par un routeur lorsque celui-ci peut détecter, par le biais du logiciel de couche de liaison situé au niveau de son interface de réseau, que l'hôte de destination n'existe pas. Ainsi, par exemple, le logiciel de couche de liaison de données peut déterminer qu'un hôte n'existe pas

s'il est connecté à l'hôte par une liaison point à point. Le protocole de couche de liaison de données pour les liaisons point à point peut en effet détecter l'existence de périphériques à chacune des extrémités de la liaison.

Si un routeur est connecté à l'hôte de destination par le biais d'un réseau local à diffusion, tel qu'Ethernet, le routeur ne peut déterminer que l'hôte n'existe pas. Dans le cas d'un réseau local, le routeur envoie une requête ARP pour déterminer l'adresse matérielle, et s'il ne reçoit pas de réponse ARP, il suppose que l'hôte de destination est inaccessible. Il ne peut toutefois en conclure que l'hôte n'existe pas. Il peut être simplement mal configuré.

Les routeurs doivent utiliser des messages Hôte inaccessible (code 1) ou Hôte de destination inconnu (code 7) lorsque d'autres hôtes du même réseau de destination sont accessibles. Autrement, l'émetteur pourrait conclure, à tort, que tous les hôtes du réseau sont inaccessibles.

Type ICMP 3, Code 8 : Hôte source isolé

STD3, qui est décrit par la RFC 1122, définit un message de code 8. Ce message est généré lorsqu'un routeur peut détecter que l'hôte source est isolé du reste du réseau. Toutefois, la RFC 1812 déclare que les routeurs ne doivent pas générer des messages de code 8, mais doivent utiliser plutôt un message Réseau inaccessible (code 0) ou Hôte inaccessible (code 1), selon la circonstance. Les messages de code 8 sont donc obsolètes.

Type ICMP 3, Codes 9 et 10 : Communication avec le réseau/hôte de destination interdit par l'administrateur

De nombreux routeurs permettent à l'administrateur d'interdire le routage vers des destinations spécifiques, et ce, même si ces destinations sont accessibles par le routeur. Ces interdictions peuvent être motivées par la politique de réseau de l'organisation. Une politique de réseau est déterminée par des considérations administratives, des considérations liées aux coûts, ou à la sécurité.

Un code 9 indique que la communication avec le réseau de destination est interdite administrativement, tandis que le code 10 indique que c'est la communication avec l'hôte de destination qui est interdite.

Les codes 9 et 10 furent conçus pour être utilisés par des périphériques de chiffrement point à point des agences militaires américaines. La RFC 1812 déclare que les routeurs doivent utiliser le code 13 nouvellement défini (communication interdite par l'administrateur) dans le cas d'un filtrage administratif des paquets.



Type ICMP 3, Codes 11 et 12 : Réseau/hôte de destination inaccessible pour le type de service

Ces messages sont générés si un routeur est incapable de transmettre le datagramme à un réseau ou à un hôte de destination parce que les routes de sa table de routage ne peuvent trouver le type de service (TOS) demandé, ou la valeur de TOS par défaut. La valeur de TOS est spécifiée dans l'en-tête IP du datagramme et permet de spécifier la préséance, le retard, le débit et le coût du chemin suivi par le datagramme (voir Chapitre 6).

Un code 11 indique que le réseau de destination est inaccessible pour le TOS spécifié, un code 12 indique que c'est l'hôte de destination qui est inaccessible.

Type ICMP 3, Code 13 : Communication interdite par l'administrateur du fait d'un coupe-feu (*fire-wall*)

La plupart des routeurs récents disposent d'un logiciel de filtrage des paquets permettant de mettre en place des règles de filtrage interdisant la transmission de paquets au vu de l'adresse IP, du type de protocole, des numéros de port spécifiés, etc. Ce filtrage est une mesure de sécurité et constitue la première ligne de défense contre des intrusions hostiles.

L'implémentation d'un routeur peut envoyer des messages ICMP indiquant que le datagramme n'a pu être transmis pour des raisons administratives. Le routeur peut être configuré pour ne pas envoyer de tels messages. Dans ce cas, aucun message d'erreur ICMP ne sera renvoyé pour chaque paquet refusé administrativement. De cette manière, le routeur ne risque pas de donner des informations supplémentaires à l'émetteur des paquets suspects.

Type ICMP 3, Code 14 : Violation de préséance d'hôte

Le message ICMP est généré par le routeur de premier pas. Le routeur de premier pas est le premier routeur que le datagramme doit franchir pour atteindre sa destination. Ce message ICMP est généré pour indiquer que la préséance demandée n'est pas permise pour l'adresse source-destination spécifiée, pour le protocole de couche supérieure (TCP ou UDP par exemple) ou pour les numéros de port source-destination.

Un routeur peut être configuré de manière à ne pas renvoyer de messages de code 14. Dans ce cas, aucun message ICMP ne sera généré en cas de refus d'un paquet pour cause de violation de la préséance.

Type ICMP 3, Code 15 : Precedence cut-off in effect

Ce message ICMP est généré par un routeur. Un administrateur de réseau peut spécifier un niveau de préséance minimale, requis pour une route spécifique de la table de routage. Si le datagramme a été envoyé avec une préséance inférieure, il

ne peut être transmis par cette route. Si le datagramme doit être écarté, un message ICMP Type 3, code 15 est envoyé à l'émetteur.

Le routeur peut être configuré pour ne pas générer de messages de code 15. Dans ce cas, aucun message ICMP ne sera généré à la suite d'un paquet rejeté pour cause de préséance trop basse.

• Type ICMP 4 : Source Quench

Le réseau IP est sans connexion. Les routeurs d'un réseau IP transmettent les datagrammes IP unité par unité et ne réservent pas de mémoire pour les datagrammes d'une session particulière. Le concept même de session est étranger au processus de routage. Il est donc possible que des routeurs connaissent une congestion temporaire.

Qu'est-ce que la congestion pour un routeur ? Il s'agit d'un état dans lequel il n'y a plus assez de mémoire ou de place dans le tampon pour que le routeur puisse stocker les datagrammes entrants. Cela peut survenir à la suite d'une augmentation du trafic du réseau à laquelle ne peuvent faire face les capacités de stockage du réseau et des routeurs en particulier. Lorsqu'une congestion survient, plusieurs routeurs du réseau peuvent être affectés. Les datagrammes entrants peuvent ne pas être traités immédiatement parce que le routeur est occupé à traiter d'autres datagrammes. Les tampons de mémoire sont utilisés par les routeurs pour stocker temporairement les datagrammes, avant traitement et transmission.

Les routeurs connectés à des liaisons lentes sont souvent la proie de congestion. Mais ce peut être aussi le cas de routeurs disposant de peu de mémoire ou de puissance de calcul limitée. Lorsqu'un routeur subit une congestion et que ses tampons sont pleins, ils doit écarter les datagrammes entrants. Le routeur génère alors des messages ICMP Source Quench pour demander aux émetteurs de ralentir la fréquence de leurs envois. Le message Source Quench est un mécanisme rudimentaire de régulation du flux, destiné à éviter que les routeurs ne soient submergés par les datagrammes entrants.

A quelle fréquence convient-il d'envoyer le message ICMP Source Quench ? Une implémentation simplifiée pourrait le faire pour chaque datagramme écarté à la suite de la saturation des tampons mémoire du routeur. Cette méthode est loin d'être la meilleure, car le message ICMP Source Quench ne fait alors qu'ajouter à la congestion du réseau et affecte les autres routeurs du réseau. Des implémentations plus subtiles peuvent plutôt surveiller le débit en datagrammes de différentes sources et envoyer des messages ICMP Source Quench aux sources générant le



plus grand nombre de datagrammes. De plus, l'envoi de messages Source Quench peut commencer au moment où les tampons menacent d'être saturés, afin d'éviter une saturation totale.

Comment la source sait-elle que la congestion a cessé, et qu'elle peut augmenter sa fréquence de transmission ? Il n'existe malheureusement pour l'instant pas de message ICMP permettant d'annuler ou d'inverser le message Source Quench pour indiquer que la congestion a cessé. Se pose donc la question de savoir comment l'émetteur décide qu'il est temps d'augmenter la fréquence d'envoi de ses datagrammes. En fait, au niveau de IP, la source ne connaît pas le moment favorable à l'augmentation de la fréquence, mais elle le sait au niveau du protocole TCP. En effet, TCP implémente un mécanisme sophistiqué de contrôle du flux qui peut détecter les congestions et moduler la fréquence d'envoi de datagrammes en conséquence. En fait, les problèmes de contrôle de flux sont mieux traités par les protocoles de couche de transport – tels que TCP justement –, responsables de la remise fiable des données. La RFC 1122 spécifie ainsi que la réception d'un message ICMP Source Quench doit être notifiée à la couche de transport, à qui il incombe alors d'implémenter un mécanisme en tenant compte.

La RFC 1106 décrit un mécanisme de traitement des messages Source Quench, mais cette implémentation n'est pas généralisée.

De plus, l'expérience montre que, dans la plupart des cas, les messages ICMP Source Quench n'ont que peu d'incidence sur l'amélioration des performances d'un réseau congestionné, et qu'ils constituent même une gêne en augmentant le traitement. La plupart des implémentations préfèrent donc ignorer tout simplement les messages ICMP Source Quench.

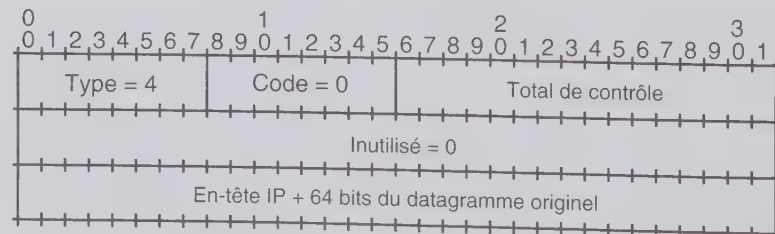
La RFC 1812 spécifie même que les routeurs ne doivent pas envoyer de messages ICMP Source Quench. Les hôtes sont tenus de recevoir tout message ICMP qui leur est adressé, mais n'ont pas l'obligation d'agir en conséquence.

La Figure 8.10 montre le format du message Source Quench.

Le champ Type vaut 4, indiquant qu'il s'agit d'un message ICMP Source Quench. Le champ Code vaut 0. Le champ Paramètre long de 4 octets et figurant après le total de contrôle reste inutilisé et vaut 0.

Figure 8.10

Message ICMP Source Quench.



• Type ICMP 5 : Redirection

Un message ICMP Redirection est envoyé par un routeur chaque fois que celui-ci détecte qu'un autre routeur propose un chemin plus favorable pour le datagramme. Les routeurs échangent des informations de routage sur le réseau à l'aide d'un protocole de routage. A chaque instant, les routeurs disposent d'informations optimales ou presque, concernant les routes possibles dans le réseau. Quant aux hôtes, ils ne disposent que d'informations de routage minimales. Un hôte ne connaît généralement que quelques réseaux et le routeur de prochain pas permettant d'accéder à chacun de ces réseaux. Les routeurs de prochain pas sont les routeurs situés à un pas (*hop*) de la source. Autrement dit, l'hôte source et ses routeurs de prochain pas sont connectés directement (voir Chapitre 7). De plus, les hôtes disposent généralement d'un routeur par défaut, utilisé pour transmettre les datagrammes vers toutes les destinations qui ne figurent pas explicitement dans la table de routage de l'hôte.

L'initialisation de la table de routage d'un hôte se fait généralement au démarrage du système. Les informations de cette table figurent dans la zone de données de configuration de l'hôte, la plupart du temps sous forme de fichiers. Si la topologie du réseau change, les routeurs prennent connaissance des nouvelles routes permettant d'atteindre les différentes destinations, car les protocoles de routage qu'ils utilisent leur permettent d'échanger des informations sur la topologie du réseau. Les hôtes ne participent généralement pas à cet échange, ce qui fait qu'ils n'ont pas connaissance de la modification de topologie et conservent des tables de routage qui ne sont plus favorables.

La Figure 8.11 montre un exemple dans lequel on a ajouté un nouveau routeur dans le réseau. Ce nouveau routeur, R2, contient un chemin optimal pour l'hôte de destination B, mais l'hôte est encore configuré pour transmettre les datagrammes par le truchement du routeur R1. Si R1 détecte que R2 dispose d'un chemin optimal, il envoie un message ICMP Redirection à l'émetteur et transmet le datagramme à R2.

A la réception d'un message ICMP Redirection, les hôtes doivent mettre à jour leur table de routage en tenant compte des nouvelles informations concernant le nouveau routeur. Le prochain datagramme envoyé à la même destination devra utiliser la route optimale mentionnée dans le message Redirection. Toutefois, certaines implémentations d'hôte TCP/IP ignorent purement et simplement les messages ICMP de redirection, ce qui a pour conséquence une augmentation du trafic du réseau, liée à l'envoi d'un message Redirection chaque fois qu'un datagramme est envoyé au routeur non optimal.

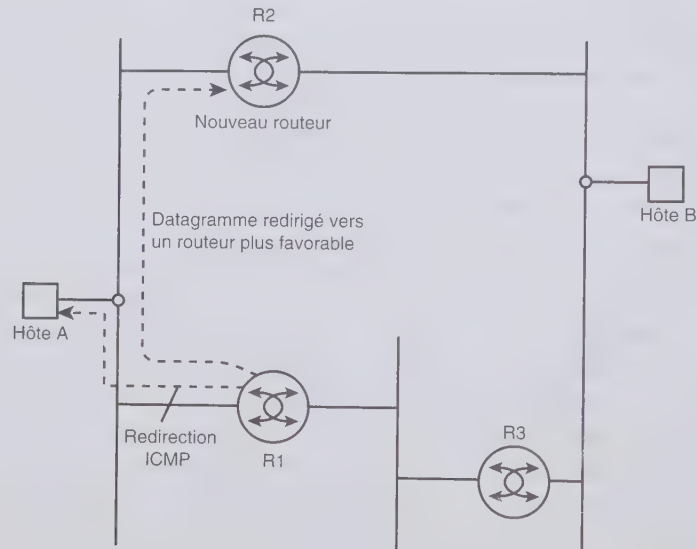
Le routeur qui génère le message ICMP de redirection transmet aussi le datagramme au routeur optimal. Comme cette transmission peut se faire sur le réseau



qui a reçu le datagramme, le nombre de datagrammes transmis est doublé (voir Figure 8.11).

Figure 8.11

Un exemple d'envoi de messages ICMP Redirection.



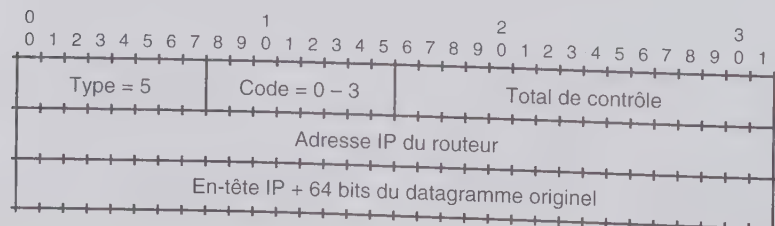
La présence de messages ICMP Redirection doit alerter l'administrateur de réseau sur la possibilité que des routes ne soient pas optimales. Si les messages ICMP de redirection continuent d'arriver sur une longue période, il convient de vérifier les tables de routage pour débusquer les entrées non optimales. Si des messages ICMP de redirection sont continuellement envoyés à l'un des hôtes donnés, il est nécessaire de mettre à jour le logiciel TCP/IP de cet hôte, car ce dernier ne doit pas ignorer les messages de redirection.

Les routeurs doivent aussi implémenter un paramètre de limite de fréquence, indiquant le nombre maximal de messages de redirection envoyés au même hôte. Ce paramètre doit être configurable en même temps que les autres paramètres de configuration du routeur.

La Figure 8.12 montre le format du message ICMP de redirection.

Figure 8.12

Message ICMP Redirection.



Le champ Type vaut 5, indiquant qu'il s'agit d'un message de redirection. Les valeurs du champ Code figurent dans le Tableau 8.4.

Tableau 8.4 : Valeurs des codes pour le message ICMP de redirection

Code	Signification
0	Rediriger le datagramme pour le réseau (ou sous-réseau)
1	Rediriger le datagramme pour l'hôte
2	Rediriger le datagramme pour le Type de Service et le réseau
3	Rediriger le datagramme pour le Type de service et l'hôte

L'adresse du routeur optimal est placée dans le champ Adresse Internet du routeur. La RFC 1812 spécifie que les routeurs ne doivent pas générer les messages Rediriger le datagramme pour le réseau (code 0), ni Rediriger le datagramme pour le type de service et le réseau (code 2). Au lieu de cela, ils doivent pouvoir générer le message Rediriger le datagramme pour l'hôte (code 1) et le message Rediriger le datagramme pour le type de service et l'hôte (code 3).

Les messages de code 0 et de code 2 s'appliquent à tous les hôtes du réseau spécifié. Ils sont plus économiques que les messages de code 1 et 3 qui sont destinés à des hôtes spécifiques. Dans le cas d'un adressage en sous-réseaux, les messages de code 0 et 2 sont difficiles à traiter parce que l'ambiguïté du masque de sous-réseau utilisé pour interpréter un message de redirection de réseau dans un environnement CIDR fait qu'il est difficile de spécifier précisément les cas dans lesquels on peut utiliser des redirections de réseau.

Le code 3 (Rediriger le datagramme pour l'hôte et le type de service) est généré lorsque le paquet qui a provoqué le message de redirection ICMP doit être envoyé par un chemin qui dépendrait, en partie, du TOS demandé. Les routeurs qui peuvent générer des messages de code 3 doivent proposer une option de configuration, activée par défaut, permettant de substituer des redirections de code 1 aux redirections de code 3. Si un routeur n'est pas capable de générer des redirections de code 3, il doit générer des redirections de code 1 dans les situations où une redirection de code 3 serait nécessaire.

Les routeurs ne doivent générer de messages de redirection que lorsque l'ensemble des conditions suivantes sont vérifiées :

- Le datagramme est transmis vers l'extérieur par l'interface physique utilisée pour sa réception.



- L'adresse IP source figurant dans les paquets est sur le même (sous-)réseau IP que l'adresse IP de prochain pas.
- Le datagramme ne contient pas une option IP Source Route.

L'adresse IP source utilisée dans le message ICMP de redirection doit appartenir au même sous-réseau logique que l'adresse de destination. Un routeur qui n'utilise pas de routes statiques, mais emploie un protocole de routage, ne doit pas prendre en compte les chemins mentionnés par les messages IP de redirection lorsqu'il transmet un paquet. En effet, les messages de redirection servent à transmettre des informations de routage aux hôtes. Les routeurs utilisent des mécanismes plus efficaces, tels que les protocoles de routage, pour obtenir leurs informations de routage. De plus, si on accepte systématiquement les messages ICMP de redirection, il se peut que des informations correctes soient remplacées dans la table de routage par des informations incorrectes, fournies par les messages de redirection, ce qui pourrait avoir comme conséquences des boucles et divers autres problèmes. La prise en compte des messages ICMP rend aussi le routeur vulnérable aux attaques d'utilisateurs mal intentionnés, qui enverraient de faux messages ICMP de redirection afin de perturber le réseau.

Quoi qu'il en soit, si un routeur n'utilise pas un protocole de routage, on peut le configurer pour qu'il prenne en compte les routes mentionnées par les messages de redirection.

Récapitulatif

ICMP est une composante obligatoire de IP et permet d'envoyer des messages d'alerte et de diagnostic par le biais du module ICMP. Le module ICMP est implémenté au sein du module IP.

Les messages ICMP sont très utiles pour la détection de problèmes réels ou potentiels sur un réseau. On peut générer des messages ICMP en réponse à des conditions d'erreur très variées, comme nous l'avons vu en détail dans ce chapitre. Les messages ICMP servent aussi à diagnostiquer un réseau et à établir des statistiques concernant celui-ci.

Testez vos connaissances

1. Citez au moins cinq exemples de problèmes que peuvent signaler des messages ICMP.
2. ICMP est-il obligatoire dans toute implémentation de IP ? Qu'en est-il du degré de détail des messages ICMP pris en charge par les différentes implémentations ?
3. Sur un réseau, quels sont les périphériques qui peuvent générer des messages ICMP ?
4. L'utilisation de ICMP rend-elle IP plus fiable ? Justifiez votre réponse.
5. Si un datagramme IP est fragmenté, génère-t-on des messages ICMP pour chaque fragment ? Justifiez votre réponse.
6. ICMP génère-t-il des erreurs concernant les messages contenant un message ICMP ? Justifiez votre réponse.
7. Quelles sont les erreurs que ne signalent pas ICMP ?
8. A partir d'un message ICMP, comment déterminer le périphérique responsable de la génération de ce message ? Comment déterminer le type de problème signalé par le message ?
9. Pourquoi les messages ICMP sont-ils envoyés à l'émetteur du datagramme concerné par le message ? Si c'est une erreur dans la table de routage d'un routeur qui est responsable de la génération du message ICMP, pourquoi ne pas envoyer le message ICMP au routeur lui-même ?
10. Quel protocole encapsule ICMP ? ICMP encapsule-t-il un autre protocole ? Dessinez un schéma indiquant toutes les couches de l'encapsulation ICMP sur un réseau Ethernet.
11. Indiquez les valeurs qui doivent être placées dans les champs d'en-tête IP pour l'envoi d'un message ICMP.
12. La remise des messages ICMP est-elle garantie ? Justifiez votre réponse.
13. Tracez le schéma du format général des messages ICMP. Décrivez les différents champs et composants de ce format.
14. A quoi servent les champs Type et Code dans un message ICMP ?
15. A quoi sert le champ Total de contrôle dans un message ICMP ?



16. A quoi servent les champs Identification et Numéro de séquence dans un message ICMP ? Donnez des exemples de messages ICMP où ces champs sont utilisés.
17. Quand peut-on mettre le champ Total de contrôle du message ICMP à zéro ? Que signifie alors ce champ ?
18. Quelles informations sont contenues dans le champ Information d'un message ICMP ?
19. Indiquez quelles sont les informations concernant les protocoles de transport TCP et UDP, qui sont fournies dans le message ICMP. Ces informations sont-elles complètes ? Combien d'octets supplémentaires devrait contenir le message ICMP pour que l'en-tête du protocole de couche de transport soit repris entièrement ?
20. Montrez en quoi les messages ICMP de types 8 et 0 permettent de diagnostiquer un réseau.
21. Quelles sont les options généralement disponibles avec l'utilitaire PING pour le diagnostic ?
22. Dressez la liste des différents types de conditions d'erreur/de réseau qui peuvent être détectées par un message ICMP de type 3.
23. A quelle occasion un message ICMP de type 3, Protocole inaccessible, sera-t-il envoyé ?
24. A quelle occasion un message ICMP de type 3, Port inaccessible, sera-t-il envoyé ?
25. A quelle occasion un message ICMP de type 3, Fragmentation nécessaire et flag Ne pas fragmenter activé, sera-t-il envoyé ?
26. Indiquez de quelle façon on peut utiliser le message ICMP de type 3, Fragmentation nécessaire et flag Ne pas fragmenter activé, pour estimer la taille minimale de MTU sur un chemin.
27. A quelle occasion un message ICMP de type 3, Echec du routage par la source, sera-t-il envoyé ? Ce type de message inclut-il des options IP de Source Route ?
28. A quelle occasion un message ICMP de type 3, Communication interdite administrativement du fait d'un coupe-feu, sera-t-il envoyé ?
29. A quelle occasion un message ICMP de type 4 (Source Quench) sera-t-il envoyé ? Pourquoi recommande-t-on que ce message ne soit pas envoyé par les routeurs ? Quelle couche OSI est la plus adaptée pour traiter le problème signalé par ce type de message ?
30. Le message d'erreur ICMP doit-il être signalé à la couche de transport ?
31. Qu'est-ce qui provoque une congestion dans un routeur ?

32. A quoi sert le message ICMP de redirection ?
33. Sous quelles conditions les routeurs doivent-ils générer des messages ICMP ?
34. Présentez les messages ICMP servant pour la découverte des routeurs. Quels sont les avantages de cette méthode par rapport à d'autres techniques ?
35. A quelle occasion un message ICMP Temps dépassé sera-t-il envoyé ?
36. A quelle occasion un message ICMP Problème de paramètre sera-t-il envoyé ?
37. A quoi servent les messages ICMP de types 13 et 14 ?
38. Pourquoi n'utilise-t-on pas le message ICMP de Requête/Réponse d'information ? Quels autres protocoles peuvent être utilisés à la place ?
39. A quoi servent les messages ICMP de Requête/réponse de Masque d'adresse ? Quels autres protocoles peuvent être utilisés à la place ?
40. Si aucune réponse n'est reçue à la suite d'une requête ICMP de masque d'adresse, que doit faire un hôte ?
41. Décodez le dump hexadécimal ci-après. Nommez les champs du protocole de couche supérieure dans ce paquet. Décrivez la fonction de ce paquet.

```

0: 00 00 C0 A2 0F 8E 00 80 C7 2D FD 05 08 00 45 00 |.....E.
10: 00 30 01 4E 00 00 FF 01 C1 C0 C7 F5 B4 09 C7 F5 |.0.N.....
20: B4 C9 0D 00 2E FF F6 00 96 00 01 65 98 8D 00 00 |.....e...
30: 00 00 00 00 00 00 00 00 00 00 9B 6C 02 A0 |.....l..

```

42. Décodez le dump hexadécimal ci-après. Nommez les champs du protocole de couche supérieure dans ce paquet. Décrivez la fonction de ce paquet.

```

0: 00 80 C7 2D FD 05 00 00 C0 A2 0F 8E 08 00 45 00 |.....E.
10: 00 28 00 62 00 00 40 01 81 B5 C7 F5 B4 C9 C7 F5 |(.b..@.....
20: B4 09 0E 00 43 8B F6 00 96 00 01 65 98 8D 02 98 |...C.....e...
30: 41 A8 02 98 41 A8 6C 74 72 65 65 63 |A...A.ltreec

```

43. Décodez le dump hexadécimal ci-après. Nommez les champs du protocole de couche supérieure dans ce paquet. Décrivez la fonction de ce paquet.

```

0: 00 80 C7 2D FD 05 00 00 C0 DD 14 5C 08 00 45 00 |.....\..E.
10: 00 38 20 61 00 00 80 01 22 65 C7 F5 B4 0A C7 F5 |.8 a...."e.....
20: B4 09 03 01 18 00 00 00 00 00 45 10 00 3C 00 E3 |.....E..<..
30: 00 00 40 06 34 C8 C7 F5 B4 09 C8 01 01 01 04 03 |...@.4.....
40: 00 17 30 D0 B0 14 |...0...

```



44. Décodez le dump hexadécimal ci-après. Nommez les champs du protocole de couche supérieure dans ce paquet. Décrivez la fonction de ce paquet.

```

0: 01 00 5E 00 00 01 00 00 C0 DD 14 5C 08 00 45 00 |...^.....\..E.
10: 00 24 ED AD 00 00 01 01 70 2A C7 F5 B4 0A E0 00 |.$......p*.....
20: 00 01 09 00 72 F5 01 02 07 08 C7 F5 B4 0A 00 00 |....r.....
30: 00 00 00 00 11 02 00 F0 00 00 99 00          |.....

```



Sous-réseaux et surréseaux

Les formats d'adresse d'IPv4 ont été conçus pour s'adapter à des réseaux de différentes tailles. Ces formats convenaient bien dans les premiers temps de l'Internet. Cependant, à mesure que l'Internet se développait, le format a montré ses limites. La principale faiblesse du format d'adresse IP réside dans le gâchis d'espace d'adressage consécutif à la conception d'un interrésseau.

Pour pallier ce gâchis d'espace d'adressage IP, la RFC 950 a défini et formalisé le concept de sous-réseau en 1985. Les sous-réseaux permettent d'utiliser un numéro de réseau unique pour construire plusieurs réseaux interconnectés. Le préfixe de numéro de réseau est partagé entre plusieurs réseaux interconnectés, appelés "sous-réseaux". Cette mise en sous-réseaux est généralement utilisée lorsque le même numéro de réseau est utilisé pour plusieurs réseaux interconnectés.

Il y a peu est apparu le concept de surréseau, aussi appelé "CIDR" (*Classless Internet Domain Routing* ou "routage de domaine Internet sans classe"). La mise en surréseau consiste à combiner plusieurs numéros de réseaux en un seul numéro de réseau logique qui peut être utilisé pour construire un réseau de bien plus grande taille. La mise en surréseau est utilisée le plus souvent pour combiner plusieurs adresses de classes C en une classe logique plus grande qu'une classe C, mais plus petite qu'une classe B en termes de nœuds IP pouvant être connectés.

Sous-réseaux

Dans le plan d'adressage IPv4, on sélectionne un nombre de 32 bits pour former une adresse IP quelconque. L'adresse IP est composée de deux parties :

- une adresse de réseau (aussi appelée "identification de réseau", "ID de réseau" ou "netid") ;
- une adresse d'hôte (aussi appelée "identification d'hôte", "ID d'hôte" ou "hostid").

L'adresse de réseau est commune à tous les hôtes d'un réseau donné. Généralement, le concepteur du réseau affecte une adresse de réseau distincte pour un réseau donné, puis les hôtes se voient affecter des valeurs d'ID d'hôte pour former l'adresse IP. Cette manière de procéder fait que tous les hôtes du réseau partagent un même préfixe de réseau.



Le nombre de bits utilisés pour le préfixe d'adresse de réseau dépend du format de l'adresse IP. Dans le cas d'un réseau de classe A, 8 bits sont utilisés pour l'adresse de réseau et 24 bits pour l'adresse d'hôte. Dans un réseau de classe B, 16 bits sont utilisés pour l'adresse de réseau et 16 pour l'adresse d'hôte. Dans un réseau de classe C, 24 bits sont utilisés pour l'adresse de réseau et 8 bits pour l'adresse d'hôte. Le format d'adresse de classe D est utilisé pour la multidiffusion (multicast) et la classe E est réservée à un usage ultérieur.

Ce plan d'adressage IP distinguant adresse de réseau et adresse d'hôte a pour avantage de minimiser le nombre d'entrées dans la table d'un routeur. Au lieu d'avoir une entrée pour chaque hôte d'un réseau, on peut ainsi prendre en compte tous les hôtes d'un même réseau en utilisant une entrée unique ne contenant que la partie adresse de réseau, qui est le préfixe commun à tous les hôtes de ce réseau. Les routeurs comparent les préfixes qui figurent dans leur tables de routage avec le préfixe de l'adresse de destination du datagramme IP à router. Si les préfixes correspondent, la route correspondante est sélectionnée.

Les sections qui suivent détaillent la mise en sous-réseau, qui consiste à étendre cette notion de préfixe commun pour permettre de subdiviser les réseaux.

• Pourquoi des sous-réseaux ?

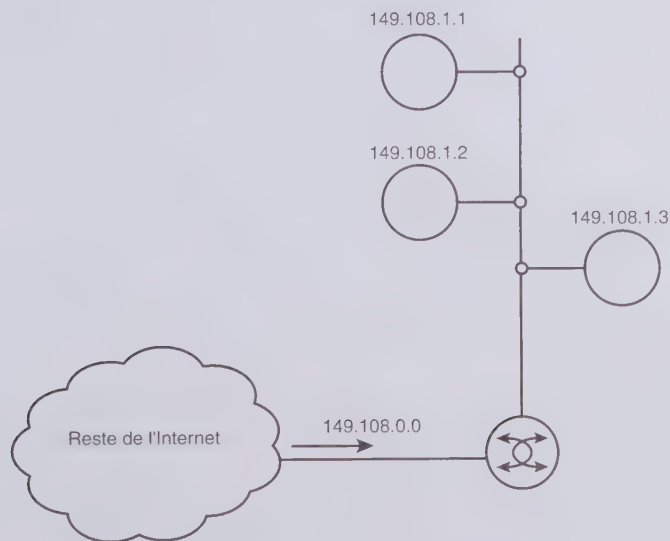
Lorsqu'un routeur envoie un datagramme à un réseau, il n'examine que la partie réseau de l'adresse IP de destination du datagramme. Il n'examine pas la partie hôte de cette adresse. Autrement dit, dans le cas d'une adresse de classe A, le routeur n'examine que le premier octet de l'adresse. Dans le cas d'une adresse de classe B, seuls les deux premiers octets sont examinés. Dans le cas d'une adresse de classe C, seuls les trois premiers octets sont examinés.

Prenons l'exemple d'un réseau IP de numéro 149.108.0.0, disposant de 16 bits à affecter. Ces 16 bits donnent un total de 2^{16} combinaisons possibles pour les hôtes, soit 65 536. Sur ces 65 536 combinaisons, vous ne pouvez pas utiliser le motif constitué exclusivement de 1, car ce motif est réservé pour l'adresse de diffusion. Vous ne pouvez de même pas utiliser la motif constitué exclusivement de 0, car ce motif est réservé pour le réseau lui-même. Sur ce total de 65 536 numéros d'hôtes potentiels, seuls deux numéros sont interdits. Cela laisse 65 534 numéros à affecter (voir Tableau 2.12).

Prenons maintenant l'exemple d'un réseau de classe B (voir Figure 9.1), dans lequel le réseau numéro 149.108.0.0 (appelé réseau 1) est connecté à l'Internet par à l'aide d'un routeur. Tout le trafic pour ce réseau de numéro 149.108.0.0 est envoyé au routeur de ce réseau. Le réseau peut contenir 65 534 hôtes. Toutefois, si

le réseau doit tenir dans un unique bâtiment, il comportera bien moins que 65 534 hôtes sur le même réseau physique. Même si la place risque de manquer, ce réseau conserve la capacité de s'étendre à 65 534 hôtes.

Figure 9.1
Un réseau de classe B connecté
à l'Internet.



Supposons maintenant que l'entreprise souhaite construire un second réseau (dans un bâtiment séparé, par exemple) et décide de connecter ce nouveau réseau à l'Internet. Quel numéro de réseau faudra-t-il utiliser ? Si l'on utilisait le numéro de réseau 149.108.0.0 pour ce second réseau, le routeur IP ne pourrait distinguer entre les deux réseaux en examinant le champ netid de l'adresse IP. Il serait possible d'affecter un nouveau numéro de réseau correspondant à des classes A, B ou C. Mais il faudrait pour cela faire une demande de nouveau numéro, alors même que des nombreux hostids potentiels n'ont pas été utilisés dans le réseau 1. L'utilisation d'un nouveau numéro constitue, on le voit, une piètre gestion de l'espace d'adressage IP.

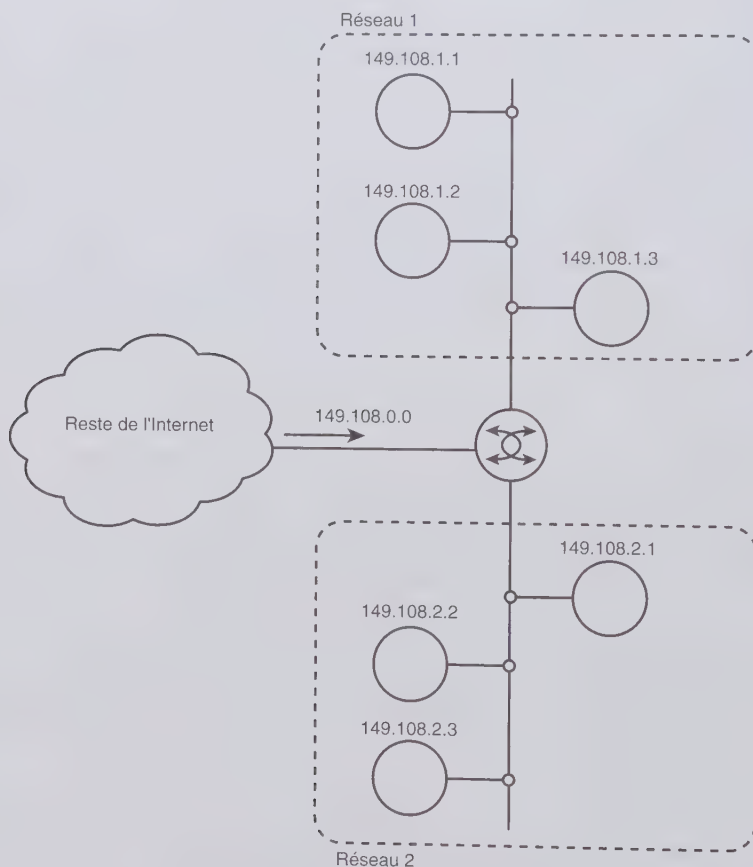
Une meilleure méthode consiste à utiliser une partie des bits du champ hostid pour effectuer une distinction entre les deux réseaux, le reste des bits servant pour affecter des numéros d'hôtes. Cette méthode s'appelle la *mise en sous-réseau*, et les réseaux résultants sont appelés des *sous-réseaux*. La méthode de mise en sous-réseau est documentée dans la RFC 950.



La Figure 9.2 montre que l'on peut connecter un second réseau (réseau 2) au premier et au reste de l'Internet en utilisant le même routeur qu'à la Figure 9.1. Seule condition, ce routeur doit disposer d'un autre port inutilisé. A la Figure 9.2, on utilise le premier octet du champ hostid pour faire la distinction entre les deux réseaux. Dans le champ hostid, les bits permettant de faire cette distinction sont appelés les *numéros de sous-réseau*. Ici, le réseau 1 a un numéro de sous-réseau valant 1 et le réseau 2 un numéro de sous-réseau valant 2.

Figure 9.2

Un réseau de classe B connecté à l'Internet à l'aide de sous-réseaux.



• Définition et avantages de la mise en sous-réseau

La mise en sous-réseau est un mécanisme permettant de fractionner un réseau en plusieurs réseaux plus petits tout en utilisant le même numéro de réseau. Outre une utilisation optimisée de l'espace d'adressage IP, la mise en sous-réseau présente trois avantages :

- une administration simplifiée ;
- la possibilité de restructurer un réseau interne sans affecter des réseaux externes ;
- une sécurité améliorée.

L'administration est simplifiée parce qu'on utilise des routeurs pour partitionner les réseaux à l'aide de frontières logiques. Ce partitionnement permet d'administrer les sous-réseaux de manière indépendante et plus efficace. Les sous-réseaux peuvent même être administrés par une entité indépendante. Ce type de configuration permet d'éviter les problèmes de gestion qui surviennent lorsque différentes entités d'une même entreprise ont des exigences contradictoires concernant la gestion du réseau.

Les sous-réseaux permettent aussi de structurer le réseau de façon interne sans que le reste du réseau connecté n'ait conscience de ces changements. A la Figure 9.2, le réseau interne a été divisé en deux sous-réseaux, mais le trafic externe provenant du réseau interne est toujours envoyé à l'adresse de réseau 149.108.0.0. Le routeur qui appartient à l'organisation responsable du réseau doit faire la distinction entre les adresses IP appartenant à ses différents sous-réseaux. Un des aspects les plus importants de cette "invisibilité" des sous-réseaux est qu'une organisation peut restructurer entièrement son réseau interne sans qu'il soit nécessaire d'obtenir un nouveau numéro de réseau. Cela n'est pas un mince avantage maintenant que les numéros de réseaux se font rares.

Comme la structure du sous-réseau interne reste invisible pour les réseaux externes, la sécurité du réseau s'améliore indirectement.

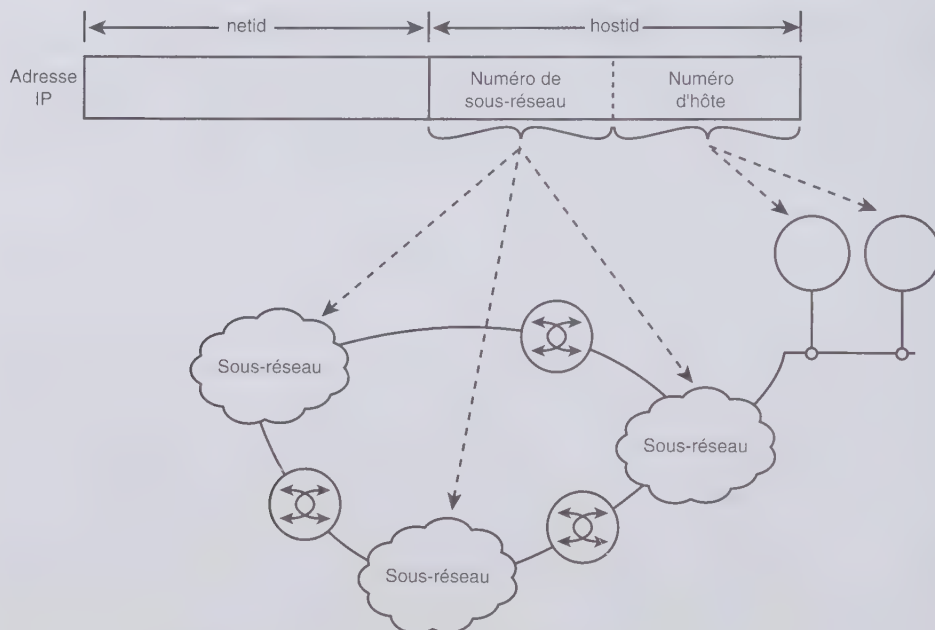
La Figure 9.3 montre les relations existant entre les sous-réseaux et les différents champs d'une adresse IP. Si les sous-réseaux de la Figure 9.3 sont connectés, il est nécessaire d'utiliser des routeurs pour les relier. De plus, les routeurs doivent avoir conscience qu'il s'agit là de sous-réseau et doivent savoir combien de bits du champ *hostid* sont utilisés pour distinguer un sous-réseau d'un autre.

On doit faire en sorte que chaque routeur de la Figure 9.3 comprenne que le champ *hostid* de l'adresse IP doit être traité de manière particulière, autrement dit, qu'une partie de ce champ représente le numéro de sous-réseau et que le reste constitue le numéro d'hôte. Ces informations sont généralement présentées au routeur sous forme d'un *masque de sous-réseau*.



Figure 9.3

*Sous-réseaux
et numéros de
sous-réseaux.*



• Masques de sous-réseau

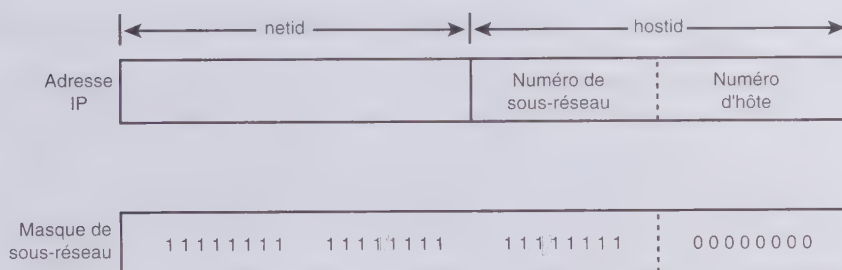
Le masque de sous-réseau est utilisé par les routeurs et les hôtes d'un sous-réseau pour interpréter le champ *hostid* afin de déterminer combien de bits sont utilisés pour la mise en sous-réseau. Le masque de sous-réseau divise le champ *hostid* en un numéro de sous-réseau et un numéro d'hôte. Le masque de sous-réseau est un nombre à 32 bits dont les valeurs sont obtenues à l'aide des règles suivantes :

- Les chiffres "un" (1) dans le masque de sous-réseau correspondent à la position de l'ID de réseau et du numéro de sous-réseau dans l'adresse IP.
- Les zéros (0) dans le masque de sous-réseau correspondent à la position du numéro d'hôte dans l'adresse IP.

La Figure 9.4 illustre ces règles. Dans la figure, un numéro de réseau de classe B est utilisé pour une mise en sous-réseau. Huit bits du champ *hostid* sont utilisés pour le numéro de sous-réseau. Le masque de sous-réseau résultant apparaît à la Figure 9.4. Le masque de sous-réseau est un motif de 32 bits exprimé par convention sous une forme décimale à points. Comme un groupe de huit 1 correspond à la valeur décimale 255, la masque de sous-réseau de la Figure 9.4 peut être écrit de la manière suivante :

255.255.255.0

Figure 9.4
Représentation d'un masque
de sous-réseau
(source : *Learning Tree*).



Les 1 du masque de sous-réseau correspondent aux parties netid et Numéro de sous-réseau de l'adresse IP

Les 0 du masque de sous-réseau correspondent au numéro d'hôte de l'adresse IP

Si on utilise une valeur de masque de sous-réseau de 255.255.0.0 pour une adresse de classe B, cela signifie qu'il n'y a pas de sous-réseau. Le champ netid d'une adresse de classe B fait 16 bits. Le champ netid correspond aux deux premiers 255 (255.255) du masque de sous-réseau 255.255.0.0. Le 0.0 restant correspond donc au numéro d'hôte. Comme aucun 1 ne figure dans le masque de sous-réseau pour le numéro de sous-réseau, il n'y a pas ici de mise en sous-réseau.

Si le même masque de sous-réseau 255.255.0.0 est utilisé pour une adresse de classe A, cela indiquerait cette fois la présence de sous-réseaux. Pour une adresse de classe A, le champ netid fait 8 bits. Ce champ netid correspond au premier 255 du masque de sous-réseau. Le 255 suivant correspond donc au numéro de sous-réseau, qui fait 8 bits de long.

Si on utilise le masque de sous-réseau 255.255.255.0 pour une adresse de classe C, on signifie ainsi qu'il n'y a pas de sous-réseau. Le champ netid d'une adresse de classe C fait 24 bits. Ce champ correspond aux trois premiers 255 (255.255.255) du masque de sous-réseau. Le 0 final correspond au numéro d'hôte. Comme aucun 1 ne figure dans le masque de sous-réseau pour le champ de numéro de sous-réseau, aucun sous-réseau n'est utilisé ici.

Vous ne pouvez pas utiliser le masque de sous-réseau 255.255.0.0 pour une adresse de classe C, car cette adresse n'est pas un masque de sous-réseau valide (nous verrons par la suite que cette adresse correspond en fait à un surréseau). Une classe adresse comporte un champ netid de 24 bits, mais les deux premiers 255 du masque 255.255.0.0 ne correspondent qu'aux 16 premiers bits du champ

* 30 - 1



LE MACMILLAN

netid. Au moins un autre 255 serait nécessaire pour prendre en compte les 8 bits restants de netid.

Le masque de sous-réseau est généralement nécessaire au moment où vous spécifiez l'adresse IP d'un hôte ou d'un routeur. Cette valeur peut être exprimée sous forme décimale à points comme nous l'avons vu jusqu'ici. Certains logiciels TCP/IP utilisent un motif hexadécimal ou une notation hexadécimale à points. Pour résumer, on peut exprimer le masque de sous-réseau 255.255.255.0 des manières suivantes :

FFFFFF00 (motif hexadécimal)

0xFF.0xFF.0xFF.0x00 (notation hexadécimale à points)

Un numéro de sous-réseau composé exclusivement de 0 ou de 1 n'est pas autorisé.

Dans la plupart des systèmes d'exploitation, le masque de sous-réseau est stocké dans une zone de configuration interne. Ainsi, dans les systèmes Unix, le masque est stocké dans les fichiers `/etc/rc.local`, `/etc/netd.cf`, `/etc/rc.d/rc.inet1`, ou `/etc/netstart`.

Rétrospectivement, on se rend compte qu'il aurait pu être préférable d'utiliser une représentation différente du masque de sous-réseau afin de spécifier le nombre de bits à utiliser pour la mise en sous-réseau. On aurait ainsi pu créer un sous-réseau représenté par un nombre unique appelé "bit de sous-réseau". Ce bit aurait correspondu à la taille du champ de numéro de sous-réseau, en commençant par la partie la plus significative du champ `hostid`. En fait, certaines implémentations de TCP/IP permettent précisément de spécifier les choses de cette manière. Le logiciel TCP/IP se fonde alors sur ces informations pour déterminer les valeurs de masque de sous-réseau.

Cette façon de procéder aurait pu épargner à des générations d'administrateurs de réseaux de s'arracher les cheveux devant les masques de sous-réseau. Le problème des masques de sous-réseau est qu'ils obligent souvent l'administrateur à manipuler des motifs binaires ou hexadécimaux. Or cette activité ne lui est pas forcément familière. Un programmeur système travaillant en assembleur ou en C/C++ a l'habitude de manipuler des motifs de bits. Mais ce n'est pas le cas de tout le monde.

Si vous éprouvez des difficultés à comprendre ce qu'est un masque de sous-réseau, ne désespérez pas. Le principe est simple. Seule la mise en œuvre risque de vous dérouter. Les exemples de ce chapitre vous habitueront aux diverses techniques.

• Les masques de sous-réseaux dans les RFC

Dans la littérature des RFC, les masques de sous-réseaux sont représentés sous forme du triplet suivant :

{<numéro de réseau>, <numéro de sous-réseau>, <numéro d'hôte>}

La valeur -1 représente un masque exclusivement composé de 1. Ainsi, le masque de sous-réseau 255.255.255.0 pour un réseau de classe B est représenté de cette manière :

{-1, -1, 0}

Le premier -1 correspond à un <numéro de réseau> dans lequel tous les bits sont à 1. Le second -1 correspond à un <numéro de sous-réseau> dans lequel tous les bits sont à 1. Le dernier 0 correspond à un <numéro d'hôte> valant 0.

• Conversions entre nombres décimaux et binaires

Si vous manipulez des masques de sous-réseau et des adresses IP, vous serez amené à convertir des nombres décimaux en valeurs binaires et inversement. Cette section se propose de vous montrer comment procéder.

Nous verrons tout d'abord comment convertir un nombre binaire en un nombre décimal. Nous verrons l'opération inverse plus tard.

Prenons l'exemple d'un motif binaire à 8 bits valant 10101000. On cherche à connaître la valeur décimale correspondante pour résoudre un problème concernant des adresses IP. Comme les adresses IP sont généralement représentées en notation décimale à points, selon laquelle on convertit 8 bits de l'adresse IP en une valeur décimale, nous ne traiterons ici que des motifs à 8 bits.

Le nombre binaire 10101000 est en base 2 (un nombre décimal, tel que 143, est en base 10). En fait, le mot *bit*, utilisé pour décrire un élément pouvant prendre la valeur 1 ou 0, est une contraction des mots *binary digit*, c'est-à-dire "chiffre binaire".

Dans un nombre décimal tel que 143, le 1 représente la position des centaines, le 4 celle des dizaines, et le 3 celle des unités. Le chiffre 1 représente la position des centaines, car deux chiffres se trouvent à sa droite. Ces deux chiffres correspondent à un facteur de 10 puissance 2 (10^2), soit 100. De même, le chiffre 4 dans le nombre 143 représente les dizaines, car il n'y a qu'un chiffre à sa droite. Ce chiffre correspond à un facteur de 10 à la puissance 1, soit 10. Enfin, le chiffre 3 dans



le nombre 143 représente la position des unités, car aucun chiffre ne se trouve à sa droite. Le facteur correspondant est donc 10 puissance 0, soit 1.

Reprenons notre motif binaire 10101000. Le 1 situé le plus à gauche a 7 bits à sa droite. Ce 1 correspond donc à un facteur de 2 à la puissance 7 ou 2^7 (128). Le Tableau 9.1 indique les valeurs décimales des puissances de 2. Vous pouvez aussi utiliser une propriété mathématique simple pour déterminer la valeur décimale. 2^7 peut s'exprimer comme le produit de 2^3 , 2^3 et 2, comme le montre cet exemple :

$$\begin{aligned} 2^7 &= 2^{(3+3+1)} \\ &= 2^3 \times 2^3 \times 2^1 \\ &= 8 \times 8 \times 2 = 128 \end{aligned}$$

Dans l'équation précédente, on a exprimé l'exposant 7 comme la somme $3 + 3 + 1$.

Le 1 suivant dans le motif 10101000 a cinq bits à sa droite. Le facteur multiplicatif est 2^5 , soit 32 (voir Tableau 9.1). Le dernier 1 dans 10101000 a trois bits à sa droite. Le facteur multiplicatif est 2^3 soit 8. Le motif 10101000 peut donc s'écrire :

$$\begin{aligned} 10101000 &= 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 \\ &\quad + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2 + 0 \times 1 \\ &= 128 + 32 + 8 \\ &= 168 \end{aligned}$$

Tableau 9.1 : Table des puissances de 2

Puissance de 2	Valeur décimale
2^0	1
2^1	2
2^2	4
2^3	8
2^4	16
2^5	32
2^6	64
2^7	128
2^8	256

Essayons de convertir 145 en binaire. Une des méthodes consiste à utiliser la Tableau 9.1 pour exprimer 145 comme une somme de puissances de 2. On peut procéder ainsi :

$$\begin{aligned} 145 &= 128 + 16 + 1 \\ &= 2^7 + 2^4 + 2^0 \end{aligned}$$

On a vu que 2^7 correspond au motif binaire composé d'un 1 suivi de sept 0, que 2^4 peut s'écrire en binaire avec un 1 suivi de quatre 0 et que 2^0 peut s'écrire en binaire avec un 1 suivi d'aucun 0.

$$\begin{aligned} 2^7 &= 10000000 \\ 2^4 &= 00010000 \\ 2^0 &= 00000001 \end{aligned}$$

En ajoutant ces nombres binaires, on obtient :

$$\begin{aligned} 145 &= 128 + 16 + 1 \\ &= 2^7 + 2^4 + 2^0 \\ &= 10010001 \text{ (motif binaire)} \end{aligned}$$

Vous pouvez ainsi utiliser le Tableau 9.1 pour effectuer la conversion du décimal en binaire. Mais vous pouvez recourir à une autre technique :

1. Divisez le nombre par 2. On appelle le quotient (le résultat entier) de la division Q et le reste R.
2. On place le reste R (valant 0 ou 1 puisque l'on est en base 2) dans le motif binaire, en commençant à la position la plus à droite puis en avançant d'un cran vers la gauche à chaque étape.
3. On utilise Q comme nombre à diviser dans l'étape 1. On répète la procédure jusqu'à ce que le quotient soit nul.

Voyons comment convertir 145 en nombre binaire à l'aide de cette technique :

1. On divise 145 par 2. Le quotient vaut 72. Le reste vaut 1.

Le motif binaire est 1.

2. On divise 72 par 2. Le quotient vaut 36. Le reste vaut 0.

Le motif binaire est 01.



3. On divise 36 par 2. Le quotient vaut 18. Le reste vaut 0.

Le motif binaire est 001.

4. On divise 18 par 2. Le quotient vaut 9. Le reste vaut 0.

Le motif binaire est 0001.

5. On divise 9 par 2. Le quotient vaut 4. Le reste vaut 1.

Le motif binaire est 10001.

6. On divise 4 par 2. Le quotient vaut 2. Le reste vaut 0.

Le motif binaire est 010001.

7. On divise 2 par 2. Le quotient vaut 1. Le reste vaut 0.

Le motif binaire est 0010001.

8. On divise 1 par 2. Le quotient vaut 0 (arrêt de l'algorithme). Le reste vaut 1.

Le motif binaire est 10010001.

On a obtenu un quotient nul, on arrête donc la conversion. Le motif 10010001 est la réponse recherchée.

Essayez maintenant de convertir 11001001 et 11110000 en valeurs décimales. Vérifiez votre résultat en convertissant ces valeurs décimales en binaire.

Essayez aussi de convertir 212 et 254 en motifs binaires. Vérifiez votre résultat en convertissant ces valeurs binaires en décimal.

Les sections qui suivent contiennent quelques exemples qui vous aideront à mieux comprendre l'utilisation des sous-réseaux.

• Un exemple d'utilisation de sous-réseaux : utilisation d'une adresse de classe B avec un masque de sous-réseau aligné à l'octet

Dans les questions qui suivent, on utilisera l'adresse IP et le masque de sous-réseau suivant :

adresse IP = 128.12.34.71

masque de sous-réseau = 255.255.255.0

Quelles sont les valeurs des :

numéro de sous-réseau = ?

numéro d'hôte = ?

adresse de diffusion dirigée = ?

En examinant le masque de sous-réseau 255.255.255.0, on peut voir que la division des 1 et des 0 dans le masque de sous-réseau est alignée sur un octet. Cet alignement à l'octet simplifie le calcul.

L'adresse IP 128.12.34.71 est une adresse de classe B, car 128 est compris entre 128 et 191 (voir Tableau 4.4). Les deux premiers 255 du masque de sous-réseau 255.255.255.0 correspondent aux 16 bits du champ netid pour une classe B. Le 255 suivant doit donc correspondre au numéro de réseau de l'adresse IP. En conséquence, le numéro de sous-réseau est 34. On peut l'exprimer en notation décimale à points :

Numéro de sous-réseau = 0.0.34.0

Le 0 dans le masque de sous-réseau 255.255.255.0 correspond au numéro d'hôte. Le numéro d'hôte dans l'adresse IP doit être 71. On peut l'exprimer en notation décimale à points :

Numéro d'hôte = 0.0.0.71

L'adresse de diffusion dirigée (*directed broadcast*) pour le réseau doit n'avoir que des 1 dans le champ de numéro d'hôte. Autrement dit, le dernier octet du sous-réseau doit être entièrement composé de bits à 1, ce qui s'exprime de la manière suivante en notation décimale à points :

Adresse de diffusion dirigée = 128.12.34.255

• Un exemple de sous-réseau : utilisation d'une adresse de classe avec un masque de sous-réseau non aligné à l'octet

Dans les questions qui suivent, on utilisera l'adresse IP et le masque de sous-réseau suivant :

adresse IP = 192.55.12.120

masque de sous-réseau = 255.255.255.240

Quelles sont les valeurs des

numéro de sous-réseau = ?

numéro d'hôte = ?

adresse de diffusion dirigée = ?



Au vu du masque de sous-réseau 255.255.255.240, on s'aperçoit que la division entre 1 et 0 dans le masque de sous-réseau se trouve au milieu du dernier octet. La division n'est pas alignée à l'octet, mais au bit. Afin de faciliter les calculs, la Figure 9.5 contient les différents motifs binaires correspondant aux valeurs données ci-avant.

Figure 9.5

Solution de l'exemple de sous-réseau (source : Learning Tree).

	192	55	12	120
Adresse IP (classe C) =	1 1 0 0 0 0 0 0	0 0 1 1 0 1 1 1	0 0 0 0 1 1 0 0	0 1 1 1 0 0 0 0
	255	255	255	240
Masque de sous-réseau =	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 0 0 0 0
	← Réseau →			↔ Numéro de sous-réseau Numéro d'hôte ↔
Numéro de sous-réseau =	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 1 1 1 0 0 0 0
	0	0	0	112
Numéro d'hôte =	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 1 0 0 0
	0	0	0	8
Adresse de broadcast =	1 1 0 0 0 0 0 0	0 0 1 1 0 1 1 1	0 0 0 0 1 1 0 0	0 1 1 1 1 1 1 1
	192	55	12	127

L'adresse IP 192.55.12.120 est une adresse de classe C, car 192 est compris entre 192 et 223 (voir Tableau 4.1). Les trois premiers 255 du masque de sous-réseau 255.255.255.240 correspondent aux 24 bits du champ netid d'une adresse de classe C. Le 240 restant doit donc correspondre au numéro de sous-réseau de l'adresse IP. La valeur décimale 240 correspond au motif binaire 11110000. Voici le masque de sous-réseau sous forme binaire :

11111111 11111111 11111111 11110000

Les quatre derniers 1 correspondent au champ de numéro de sous-réseau de l'adresse IP. Voici la représentation binaire de l'adresse IP, dans laquelle on a mis en gras le champ de sous-réseau :

11000000 00110111 00001100 **01111000**

Le motif binaire 0111 du champ de sous-réseau correspond à la valeur décimale 7, mais cette valeur fait partie de la valeur des 8 derniers bits de l'adresse IP. Ce numéro de sous-réseau peut s'exprimer ainsi sous forme de motif binaire :

00000000 00000000 00000000 01110000

Le numéro de sous-réseau exprimé en notation décimale à point est le suivant :

0.0.0.112

Le 0 dans le masque de sous-réseau 255.255.255.240 correspond au numéro d'hôte. Le numéro d'hôte dans l'adresse IP est mise en gras dans le motif binaire suivant :

11000000 00110111 00001100 0111**1000**

Ce numéro d'hôte s'exprime ainsi sous forme de motif binaire :

00000000 00000000 00000000 0000**1000**

Voici le numéro d'hôte exprimé en notation décimale à points :

0.0.0.8

L'adresse de diffusion dirigée doit ne contenir que des 1 dans le champ de numéro d'hôte. Les quatre derniers bits du sous-réseau doivent donc être tous égaux à 1.

On obtient donc l'adresse de diffusion dirigée suivante (exprimée sous forme binaire) :

11000000 00110111 00001100 0111**1111**

Les 1 en gras représentent les 1 utilisés dans les champs de numéro d'hôte formant l'adresse de diffusion dirigée. Cette adresse s'exprime ainsi en notation décimale à points :

= 192.55.12.127

• Masques de sous-réseaux non alignés à l'octet

Dans l'exemple précédent, on utilisait le masque de sous-réseau 255.255.255.240. Le dernier nombre décimal 240 correspond au motif binaire 11110000. Il est préférable de connaître les motifs binaires correspondant à un champ de numéro de sous-réseau faisant de 1 à 7 bits. Le Tableau 9.2 indique la taille décimale du sous-réseau et les valeurs décimales du masque de sous-réseau. Dans ce tableau, on suppose que les bits du numéro de sous-réseau sont tous *contigus* (c'est-à-dire regroupés sans aucun interstice), et on retrouve ce cas de figure dans la plupart des réseaux. Toutefois, rien dans les RFC n'interdit l'usage de numéros de sous-réseau non contigus. Autrement dit, il se peut que les bits de numéro de sous-réseau et ceux du numéro d'hôte alternent dans le masque de sous-réseau. Ainsi, on peut avoir le masque de sous-réseau suivant pour une adresse de classe B :

11111111 11111111 **10110101** 00001111



soit, en décimal :

255.255.171.15

Dans ce masque de sous-réseau, on peut voir neuf bits de numéro de sous-réseau (marqués en gras), mais ils sont en alternance avec les bits du numéro d'hôte. Bien que la RFC 950 autorise une telle monstruosité, la plupart des logiciels TCP/IP ne fonctionneront pas avec des bits de numéro de sous-réseau non contigus. L'usage de tels masques n'a aucun intérêt pratique, sinon de dérouter de futurs administrateurs de réseau.

Tableau 9.2 : Taille de sous-réseau et valeurs décimales

Taille de sous-réseau (en bits)	Motif binaire	Valeur décimale
1	10000000	128
2	11000000	192
3	11100000	224
4	11110000	240
5	11111000	248
6	11111100	252
7	11111110	254

- **Espace d'adressage perdu avec un masque de sous-réseau unique**

L'utilisation d'un masque de sous-réseau unique peut avoir pour conséquence un gâchis d'espace d'adressage si les sous-réseaux ne prennent pas en charge le même nombre de nœud sur un réseau. C'est le cas pour deux sous-réseaux séparés par une liaison point à point (voir Figure 9.6).

La figure montre un masque de sous-réseau 255.255.255.0 utilisé pour le numéro de réseau de classe B divisé en ces trois sous-réseaux :

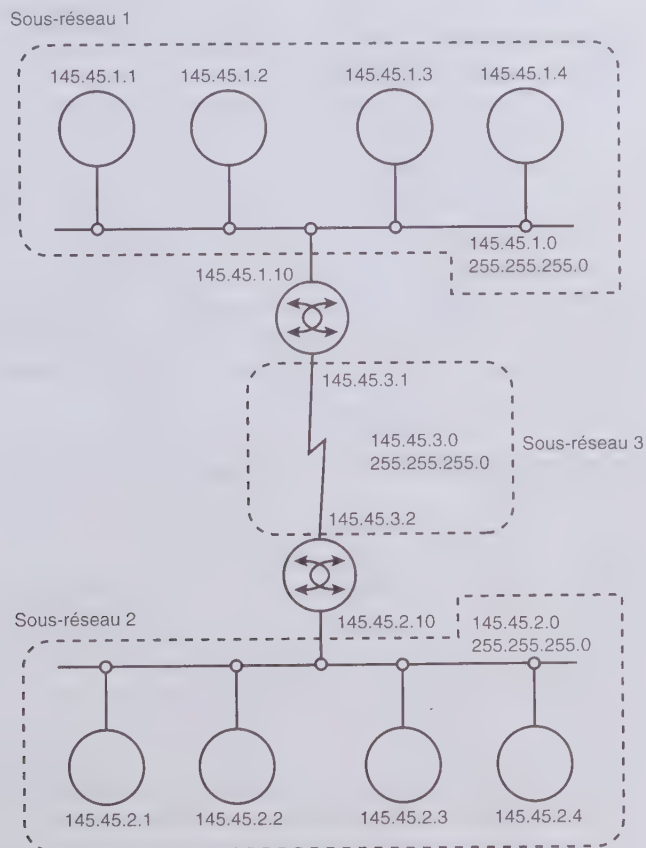
sous-réseau 1 : 145.45.1.0

sous-réseau 2 : 145.45.2.0

sous-réseau 3 : 145.45.3.0

Figure 9.6

Des sous-réseaux séparés par une liaison point à point.



Aussi surprenant que cela puisse paraître, le sous-réseau 3 a été affecté à une liaison point à point qui n'a besoin que de deux adresses IP affectées à ses extrémités. A moins qu'un routeur n'utilise une technique interne spéciale lui permettant de reconnaître cette liaison point à point, il est nécessaire d'affecter un identificateur de réseau distinct à ce sous-réseau de manière que les routeurs puissent envoyer des datagrammes via cette liaison.

Chacun des sous-réseaux peut prendre en charge jusqu'à 254 hôtes. Il s'agit des 2^8 hôtes représentés par les 8 bits, moins les adresses composées exclusivement de bits à zéro ou à un, qui ne peuvent pas être affectés à des hôtes.



Les sous-réseaux 1 et 2 sont des réseaux locaux et sont susceptibles de se développer jusqu'à pouvoir prendre en charge 254 hôtes. Cependant, le sous-réseau 3 est un réseau point à point qui ne peut affecter que deux adresses IP à ses extrémités. Sur ce réseau point à point, les 252 adresses restantes resteront forcément inutilisées et sont donc perdues.

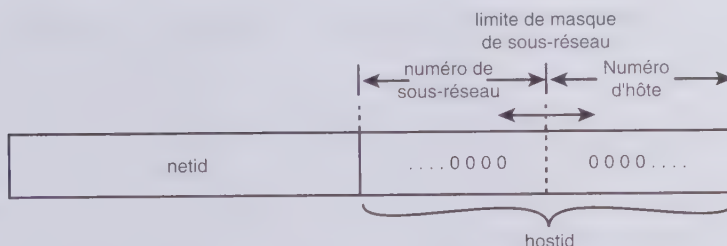
Certains routeurs commerciaux disposent de mécanismes internes leur permettant de reconnaître la présence d'une liaison point à point. Ces routeurs n'exigent pas qu'un numéro de sous-réseau soit affecté à la liaison point à point, ce qui signifie que vous n'êtes pas obligé d'affecter des adresses IP aux ports de routeur connectés à la liaison. Cependant, l'inconvénient de ne pas affecter d'adresses IP pour une liaison point à point est l'impossibilité d'employer des utilitaires tels que PING pour tester l'accessibilité des ports de routeur connectés à la liaison. Sans adresse IP, il est impossible d'identifier la liaison.

- **Affectation optimisée des numéros de sous-réseaux**

Pour la mise en sous-réseau, la partie hôte de l'adresse IP est divisée en deux champs : le numéro de sous-réseau et le numéro d'hôte. Cette division permet d'établir un troisième niveau de hiérarchie sur l'adresse IP. Bien que cette manière de procéder permette de minimiser le routage et d'utiliser un même préfixe de réseau pour plusieurs sous-réseaux, il en résulte souvent une allocation inefficace des adresses. Le problème se situe au niveau de l'administrateur de réseau qui sélectionne une taille de sous-réseau, en prenant en compte le plus grand nombre de sous-réseaux et d'hôtes susceptibles d'exister un jour. Toutefois, une technique permet à un administrateur de sélectionner des numéros de sous-réseaux et d'hôtes de telle façon que l'on puisse facilement modifier si nécessaire le masque de sous-réseau, en réduisant au minimum les modifications d'adresses IP pour les hôtes de chaque sous-réseau.

Pour comprendre de manière intuitive cette technique, examinez la Figure 9.7 dans laquelle la ligne en pointillé marque la frontière entre le numéro d'hôte et le numéro de sous-réseau. Cette frontière est appelée la *frontière de masque de sous-réseau*. A gauche de cette frontière, le masque de sous-réseau est exclusivement composé de 1. A droite de cette frontière, le masque est composé de 0. Si le numéro de sous-réseau et le numéro d'hôte sont tels que des zéros se trouvent près de la frontière de masque de sous-réseau, déplacer la frontière vers la gauche ou la droite affectera le masque de sous-réseau, mais pas les numéros de sous-réseau et d'hôte. En fait, on peut déplacer la frontière aussi loin à gauche ou à droite que l'on veut sans modifier les numéros de sous-réseaux et d'hôtes, jusqu'à la première occurrence d'un 1 dans le numéro de sous-réseau ou d'hôte. Il faut donc affecter des numéros de sous-réseaux contenant des 1 le plus à gauche et des numéros d'hôtes contenant des 1 le plus à droite. La région centrale de l'adresse doit être autant que possible remplie de zéros.

Figure 9.7
Optimisation du masque
de sous-réseau.



L'affectation des numéros de sous-réseaux et d'hôtes s'appelle le *comptage en miroir*, car les numéros de sous-réseaux et d'hôtes sont placés en vis-à-vis comme devant un miroir.

Prenons des exemples de motifs binaires pour les numéros d'hôtes et de sous-réseaux :

Numéro de sous-réseau :

```

0      (réservé, signifie "ce sous-réseau")
10
01
110
001
101
:
:
11...11
111...11 (réservé, signifie "tous les sous-réseaux")

```

Numéro d'hôte :

```

0      (réservé, signifie "cet hôte")
01
10
011
100
101
:
:
11...1110
11...1111 (réservé, signifie "tous les hôtes")

```

Cette technique d'affectation est décrite en détail dans la RFC 1219.



• Masques de sous-réseau de longueur variable

Dans les exemples traités jusqu'ici, on a utilisé le même masque de sous-réseau pour un réseau donné. Avec une adresse de 134.65.0.0 (classe B), le masque de sous-réseau peut être 255.255.255.0. Cela signifie qu'on utilise 8 bits du champ de l'adresse hôte pour la mise en sous-réseau. TCP/IP permet d'utiliser d'autres masques de sous-réseau pour cette même adresse. Ainsi, un réseau subdivisé de 134.65.0.0 pourrait utiliser le masque de sous-réseau 255.255.255.192, tandis qu'un autre utiliserait 255.255.255.0. Lorsque différents masques de sous-réseaux sont utilisés, toutes les combinaisons d'adresses IP d'hôte et de masques de sous-réseaux ne sont pas autorisées.

Supposons que l'on souhaite diviser un réseau de classe C 216.144.108.0 en trois sous-réseaux, avec au maximum 120 hôtes sur un sous-réseaux et 60 hôtes sur chacun des deux autres sous-réseaux. Quel masque de sous-réseau doit-on alors utiliser ? Le nombre total d'hôtes sur les trois sous-réseaux est de $120 + 60 + 60$ soit 240 hôtes. Un réseau de classe C permet d'avoir 254 hôtes. Il est donc possible d'y faire figurer 240 hôtes. Cependant, lorsque vous choisissez un masque de sous-réseau unique, vous devez prendre en compte le plus grand sous-réseau possible, en l'occurrence celui qui comprend 120 hôtes. Pour ce sous-réseau, un seul masque de sous-réseau est possible :

```
11111111 11111111 11111111 1 0000000
255.255.255.128
```

Le numéro de sous-réseau est le **1** en gras et le numéro d'hôte est constitué des sept zéros en gras (**0000000**). Avec sept bits d'identification d'hôte, il est possible de définir deux sous-réseaux, pouvant accepter chacun $2^7 - 2 = 126$ hôtes. Ce type de division en sous-réseaux permet de prendre en compte les besoins du sous-réseau de 120 hôtes, et ceux d'un autre sous-réseau comportant 60 hôtes, mais pas celui du dernier sous-réseau de 60 hôtes. Trois sous-réseaux sont nécessaires, mais seuls deux sont disponibles. En fait, si vous examinez le masque de sous-réseau 255.255.255.128, vous pouvez constater qu'il n'est pas valide en regard des RFC, même s'il est permis par la plupart des routeurs commerciaux. 255.255.255.128 n'est pas un réseau de classe C valide, car le numéro de sous-réseau est alors 0 ou 1 (c'est-à-dire entièrement composé de bits à 0 ou à 1). Ces motifs ne sont pas admis pour les numéros de sous-réseau. Quoi qu'il en soit, nous avons déjà vu que la plupart des routeurs commerciaux autorisent ce masque de sous-réseau.

Supposons que l'on utilise deux bits pour le numéro de sous-réseau, ce qui permet de définir 2^2 soit 4 sous-réseaux. Le masque de sous-réseau devient alors :

```
11111111 11111111 11111111 11 000000
255.255.255.192
```

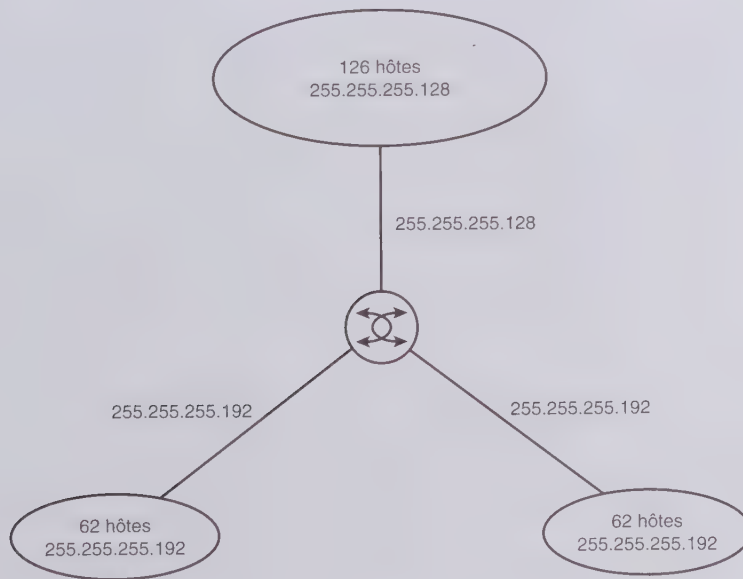
Le numéro de sous-réseau est le **11** en gras, et le numéro d'hôte est constitué des six zéros en gras (**000000**). Avec six bits d'identification d'hôte, il est possible de définir quatre sous-réseaux, pouvant accepter chacun $2^6 - 2 = 62$ hôtes. Ce type de division en sous-réseaux permet de prendre en compte les besoins de deux sous-réseaux comportant chacun 60 hôtes, mais pas ceux du troisième réseau prenant en charge 120 hôtes. En effet, le masque de sous-réseau 255.255.255.192 n'autorise que 62 hôtes. Cela est bien inférieur au nombre d'hôtes requis pour le troisième réseau.

Cet exemple montre que, dans certaines situations, il est impossible d'utiliser un masque de sous-réseau unique pour le réseau de classe C. On a besoin d'un moyen pour diviser le réseau de classe C en sous-réseaux de 126 hôtes, puis de diviser un de ces deux sous-réseaux en deux nouveaux sous-réseaux de $2^6 - 2 = 62$ hôtes. Cette technique, appelée *Variable Subnet Length Mask* (VLSM ou "masque de sous-réseau de longueur variable"), est présentée dans la RFC 1878.

La Figure 9.8 montre un exemple d'utilisation de masques de sous-réseau de longueur variable. Dans cet exemple, on utilise le masque de sous-réseau 255.255.255.128 pour diviser le réseau de classe C en deux sous-réseaux de cent vingt-six hôtes chacun. On utilise ensuite le masque de sous-réseau 255.255.255.192 pour diviser un des sous-réseaux en deux sous-réseaux de soixante-deux hôtes.

Figure 9.8

Un exemple de masque de sous-réseau de longueur variable.



A titre de référence, le Tableau 9.3 dresse la liste du nombre d'hôtes et de sous-réseaux fournis par les routeurs commerciaux. Notez que la colonne Nombre de sous-réseaux inclut les combinaisons ne contenant que des 0 ou des 1, combinaisons interdites par les RFC, mais néanmoins autorisées par la plupart des routeurs commerciaux.

Tableau 9.3 : Capacité des réseaux de classe C, pour les routeurs commerciaux

Masque de sous-réseau	Nombre de sous-réseaux	Nombre d'hôtes
255.255.255.128	2	126
255.255.255.192	4	62
255.255.255.224	8	30
255.255.255.240	16	14
255.255.255.248	32	6
255.255.255.252	64	2

Tous les protocoles de routage ne sont pas en mesure de gérer tous les masques de sous-réseaux utilisés dans un réseau. Ainsi, RIP version 1 ne peut pas traiter un jeu de différents masques de sous-réseaux pour une même adresse de réseau, et RIP ne peut traiter des masques de sous-réseaux de taille variable. Si vous utilisez RIP avec des masques de sous-réseau de taille variable, la topologie de réseau indiquée par RIP ne correspondra pas à celle que vous avez créée. En effet, le format de paquet RIP ne contient pas de champ permettant de communiquer les masques de sous-réseaux à d'autres routeurs. En revanche, des protocoles de routage tels qu'OSPF (*Open Shortest Path First*), RIP version 2 et IS-IS (*Intermediate System to Intermediate System*) permettent l'échange des informations concernant les masques de sous-réseaux. Ils autorisent donc l'emploi de différents masques de sous-réseaux. Néanmoins, VLSM reste encore peu utilisé, et ce pour les raisons suivantes :

- L'affectation d'adresses IP pour des sous-réseaux doit se faire dans les blocs indiqués par les VLSM. Il est nécessaire de modifier les affectations antérieures d'adresses IP, et ces modifications ne sont pas toujours aisées.
- Certains réseaux anciens continuent d'utiliser RIP version 1, qui ne prend pas en charge les VLSM. Un protocole de routage plus récent est nécessaire.
- La technique du VLSM reste mal comprise par la plupart des administrateurs de réseaux. Le principe de sous-réseau et de masque de sous-réseau est souvent nébuleux aux yeux d'une majorité. Espérons qu'un effort de formation et des ouvrages tels que celui-ci permettront d'améliorer la situation.

Le routage dans les sous-réseaux

Au Chapitre 7, on a pu voir l'algorithme de routage IP utilisé par les routeurs et les hôtes. Cet algorithme ne prenait toutefois pas en compte les masques de sous-réseaux. Les informations de base dans une table de routage (omission faite des masques de sous-réseaux) sont constituées des entrées suivantes :

(adresse de destination, adresse de prochain pas)

L'*adresse de destination* est une adresse de réseau ou d'hôte spécifiée dans la table de routage comme une route explicite. L'*adresse de prochain pas* n'est pas le chemin complet jusqu'à destination, mais l'adresse du routeur (routeur voisin) vers lequel on doit transmettre le datagramme. L'*adresse de prochain pas* se trouve toujours sur le réseau directement connecté. Par définition, il ne peut y avoir de routeur entre l'hôte/routeur considéré et le routeur d'*adresse de prochain pas*.

L'utilisation des masques de sous-réseaux dans les tables de routage fait l'objet d'un examen détaillé dans les sections suivantes.

• Utilisation des masques de sous-réseaux dans une table de routage

L'algorithme de routage IP (voir Chapitre 7) compare la partie réseau de l'adresse de destination du datagramme IP avec l'adresse de destination figurant dans l'entrée de table de routage. Si aucune correspondance n'est trouvée, l'adresse de prochain pas sera sélectionnée pour la transmission du datagramme. On considère que l'entrée par défaut (adresse de destination 0.0.0.0) correspond à toutes les adresses de destination, et elle est sélectionnée si aucune autre adresse de destination ne correspond. Comment cet algorithme fait-il pour déterminer la partie réseau de l'adresse IP ? En regardant les bits les plus significatifs de l'adresse IP de destination afin d'en déterminer le format : classe A, B, C ou D (multidiffusion).

Lorsqu'on utilise des sous-réseaux, on ne peut directement déterminer la partie réseau d'une adresse IP à partir du format. Ainsi, le format peut être de classe B, avec les deux premiers octets utilisés pour l'identificateur de réseau. Mais si un masque de sous-réseau 255.255.255.0 est utilisé, l'algorithme de routage doit aussi utiliser le troisième octet pour déterminer la partie réseau de l'adresse. Une table de routage prenant en compte les sous-réseaux doit donc posséder une information supplémentaire dans chaque entrée, pour indiquer les masques de sous-réseau :

(adresse de destination, masque de sous-réseau, adresse de prochain pas)



Lorsque des sous-réseaux sont utilisés, l'algorithme de routage doit utiliser le masque de sous-réseau figurant dans l'entrée de la table de routage, afin d'extraire la partie réseau de l'adresse. On opère à cet effet un ET bit à bit sur l'adresse de destination du datagramme IP et le masque de sous-réseau dans l'entrée. L'algorithme regarde ensuite si le résultat correspond à l'adresse de destination indiquée dans la table de routage. Si c'est le cas, on utilise l'adresse de prochain pas de la table pour router le datagramme. Sinon, on teste l'entrée suivante. Le ET bit à bit permettant d'extraire la partie réseau de l'adresse peut être décrite par l'équation suivante :

Partie réseau extraite = adresse de destination IP ET (bit à bit) masque de sous-réseau

Prenons un exemple. Soit l'entrée de table de routage suivante :

(144.19.74.0, 255.255.255.0, 144.19.74.91)

Soient deux datagrammes dont les adresses IP de destinations sont les suivantes :

144.19.74.12

144.19.75.21

Pour déterminer la partie réseau de ces adresses, on procède comme suit :

Partie réseau extraite de 144.19.74.12 = 144.19.74.12 && 255.255.255.0
= 144.19.74.0

Partie réseau extraite de 144.19.75.21 = 144.19.75.21 && 255.255.255.0
= 144.19.75.0

En comparant la partie réseau avec l'entrée de la table de routage, on trouve une correspondance pour 144.19.74.12, mais pas pour 144.19.75.12.

L'exemple d'entrée de table de routage contient l'adresse de destination de réseau 144.19.74.0. Comment les routes spécifiques aux hôtes et les routes par défaut sont-elles représentées lorsqu'on utilise des masques de sous-réseaux ? Les routes spécifiques aux hôtes ont un masque de sous-réseau de 255.255.255.255 (tous les bits à 1) dans leur entrée de table de routage et les routes par défaut un masque de sous-réseau de 0.0.0.0 (tous les bits à zéro). Au vu de l'exemple précédent, on peut remarquer que la présence d'un 1 dans le masque de sous-réseau signifie que le bit correspondant dans l'adresse IP de destination sera significatif pour le routage. De même, la présence d'un bit à 0 indique que le bit correspondant dans l'adresse de destination n'est pas significatif pour le routage. Un masque de sous-réseau où tous les bits sont à 1 (255.255.255.255) implique que tous les bits de l'adresse de destination sont utilisés. Il s'agit donc d'une route spécifique à un hôte. Un masque de sous-réseau où tous les bits sont à zéro (0.0.0.0) implique

qu'aucun bit de l'adresse de destination n'est utilisé et que toute adresse IP de destination sera valide pour cette entrée. Prenons l'exemple d'une table de routage contenant des routes par défaut et des routes spécifiques à un hôte.

Adresse de destination	Masque de sous-réseau	Adresse de prochain pas
144.19.74.0	255.255.255.0	144.19.74.91
145.12.2.101	255.255.255.255	144.19.74.92
202.33.23.3	255.255.255.255	144.19.74.93
0.0.0.0	0.0.0.0	144.19.74.91
0.0.0.0	0.0.0.0	144.19.74.94

On trouve deux routes spécifiques à un hôte, pour les adresses de destination 145.12.2.101 et 202.33.23.3. On peut le déterminer du fait de la présence du masque de sous-réseau 255.255.255.255 dans l'entrée de table de routage. De même, on trouve deux routes par défaut pointant vers les routeurs de prochain pas 144.19.74.91 et 144.19.74.94.

Soit un datagramme IP d'adresse de destination 202.33.23.3. Lorsqu'on utilise la table de routage ci-avant, la partie réseau utilisée pour la troisième entrée est la suivante :

$$\begin{aligned}\text{Partie réseau extraite de } 202.33.23.3 &= 202.33.23.3 \ \&\& \ 255.255.255.255 \\ &= 202.33.23.3\end{aligned}$$

Cette valeur 202.33.23.3 correspond à la troisième entrée de la table :

(202.33.23.3, 255.255.255.255, 144.19.74.93)

Soit un autre datagramme IP d'adresse de destination 201.3.3.3. Lorsque vous utilisez la table de routage ci-avant, aucune occurrence n'est trouvée avant la première route par défaut. La portion réseau extraite pour cette route est la suivante :

$$\begin{aligned}\text{Partie réseau extraite de } 201.3.3.3 &= 202.33.23.3 \ \&\& \ 0.0.0.0 \\ &= 0.0.0.0\end{aligned}$$

Cette valeur 0.0.0.0 correspond à la première route par défaut, dont l'entrée est :

(0.0.0.0, 0.0.0.0, 144.19.74.91)



Avec une valeur de masque de sous-réseau de 0.0.0.0, l'opération ET bit à bit (&&) aura toujours comme résultat la valeur 0.0.0.0, qui correspond évidemment à la valeur de destination 0.0.0.0.

Si on a toujours une correspondance avec la première route par défaut, à quoi bon spécifier une seconde route par défaut qui ne sera jamais sélectionnée ? En fait, la seconde route sera sélectionnée si le routeur de la première est hors service. La plupart des routeurs commerciaux permettent à l'administrateur réseau de spécifier une valeur hors service pour un port de routeur particulier. Cette valeur hors service permet de désactiver des routes dans la table de routage. D'autres techniques, telles que l'emploi avisé de paquets de requête/réponse ICMP et la communication avec le protocole de couche de Transport au niveau de l'hôte, permettent de déterminer si une route donnée est hors service.

Si la première route par défaut est hors service, le routeur sélectionne une autre route par défaut. Bien que la RFC 1122 suggère l'emploi de plusieurs routes par défaut, de nombreuses implémentations de TCP/IP ne permettent d'utiliser qu'une seule route par défaut.

• L'algorithme de routage IP complet, tenant compte des sous-réseaux

L'algorithme de routage IP prenant en compte les sous-réseaux est décrit par les étapes suivantes :

1. On va à `AdrIPDest`, le champ d'adresse de destination, du datagramme IP. A partir de ce champ, on détermine la partie réseau `PrefixeReseau` en effectuant un ET bit à bit avec le masque de sous-réseau figurant dans l'entrée de table de routage.
2. Si `PrefixeReseau` correspond à l'ID de réseau d'un des réseaux connectés, l'hôte de destination se trouve sur le réseau directement connecté. Dans ce cas, on effectue une remise directe du datagramme IP. La remise directe peut nécessiter la résolution de `AdrIPDest` en son adresse matérielle à l'aide d'un protocole tel que ARP. Le datagramme IP est encapsulé par la trame de couche de liaison de données et est envoyé au réseau directement connecté.
3. Si on n'a pas trouvé de correspondance à l'étape 2, on examine la table de routage pour l'entrée de destination de `PrefixeReseau`. Si cette entrée existe, on transmet le datagramme IP au routeur de prochain pas indiqué dans l'entrée.
4. Si on n'a pas trouvé de correspondance à l'étape 3, on recherche dans la table de routage l'entrée par défaut 0.0.0.0. Si cette entrée existe, on transmet le datagramme IP au routeur de prochain pas indiqué dans l'entrée.

5. Si on n'a pas trouvé de correspondance à l'étape 4, le datagramme IP ne peut être routé. On signale une erreur de routage aux protocoles de couche supérieure.

Autres techniques liées aux sous-réseaux

Avant 1985, les techniques de mise en sous-réseaux décrites dans ce chapitre n'étaient pas encore en usage. Les sous-réseaux ont été introduits pour éviter l'épuisement des numéros de réseaux du fait d'une allocation des adresses mal optimisée.

Après l'apparition des sous-réseaux en 1985, cette technique devint obligatoire pour toute nouvelle implémentation de TCP/IP. Des logiciels TCP/IP plus anciens ont cependant continué leur carrière sans intégrer la notion de sous-réseau. La technique de proxy-ARP a été introduite pour permettre aux sous-réseaux de coexister avec des réseaux plus anciens et incapables de les comprendre. Une autre technique, le routage transparent, fut utilisée pour étendre une adresse – une adresse de classe A par exemple – au-delà de la frontière du routeur. Ces techniques sont présentées dans les sections qui suivent.

• Proxy ARP

Proxy ARP est utilisé dans les réseaux à diffusion pour permettre la coexistence d'hôtes capables de comprendre les sous-réseaux et d'autres incapables de les comprendre. La Figure 9.9 montre un exemple d'utilisation de cette technique dans un réseau constitué de deux sous-réseaux :

Sous-réseau 1 :

réseau = 144.19.74.0

masque de sous-réseau = 255.255.255.0.

Sous-réseau 2 :

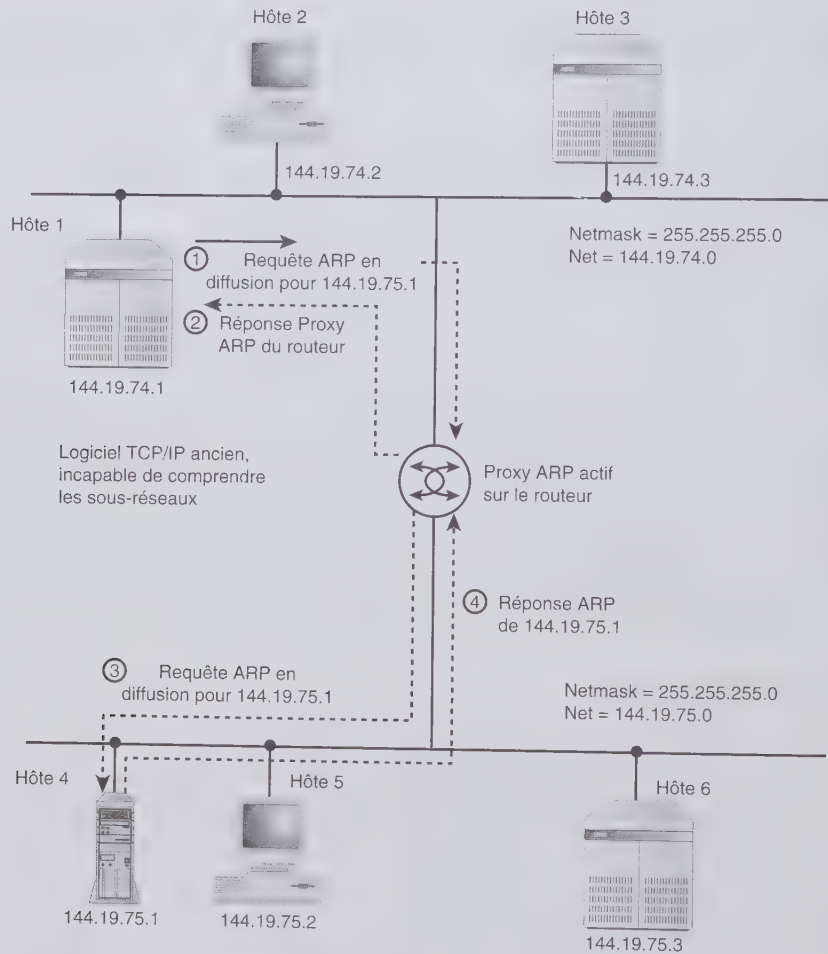
réseau = 144.19.75.0

masque de sous-réseau = 255.255.255.0.



Figure 9.9

Exemple d'utilisation de Proxy ARP.



Dans cette figure, l'hôte 1 en 144.19.74.1 souhaite communiquer avec l'hôte 4 à 144.19.75.1. Ces hôtes se trouvent sur deux réseaux distincts. Si l'hôte 1 utilise un logiciel TCP/IP qui comprend les sous-réseaux, l'algorithme de routage de l'hôte 1 peut détecter que l'hôte 4 se trouve sur un réseau distinct et transmet au routeur les datagrammes destinés à l'hôte 4. Dans cet exemple toutefois, on suppose que le logiciel TCP/IP de l'hôte 1 est trop ancien pour comprendre ce qu'est un sous-réseau. Cet hôte estime donc que le réseau directement connecté est 144.19.0.0 et que son masque de sous-réseau est 255.255.0.0.

S'il est nécessaire d'envoyer un datagramme à 144.19.75.1, l'hôte part du principe que l'hôte de destination se trouve sur le réseau directement connecté

144.19.0.0. Comme il s'agit d'un réseau à diffusion, l'hôte envoie une requête ARP à 144.19.75.1 pour connaître l'adresse matérielle de ce dernier. L'hôte 144.19.75.1 se trouve sur un autre réseau, au-delà de la frontière d'un routeur. Vous devez vous souvenir que les diffusions de couche de liaison de données, telles que la requête ARP, ne franchissent pas les frontières de routeurs, si bien que la requête ARP n'atteint pas l'hôte en 144.19.75.1. Cet hôte ne peut donc répondre à la requête ARP et l'hôte 1 attendra en vain, à moins que le routeur ne soit configuré pour utiliser proxy ARP.

Si le proxy sait utiliser proxy ARP, il détecte que l'hôte 1 demande l'adresse matérielle d'un nœud se trouvant sur l'autre sous-réseau. Le routeur répond à la requête ARP en envoyant sa propre adresse matérielle (c'est-à-dire l'adresse matérielle du port du routeur sur le réseau 144.19.74.0). L'hôte 1 suppose alors que sa requête ARP a généré une réponse appropriée de la part de l'hôte 144.19.75.1 et n'a pas conscience que la réponse ARP contient l'adresse matérielle du port du routeur. Il envoie au port du routeur tous les datagrammes destinés à l'hôte en 144.19.75.1. Le routeur sait que ces datagrammes doivent être envoyés au réseau directement connecté 144.19.75.0. Il accepte les datagrammes et les transmet donc à l'hôte 144.19.75.1 sur le réseau 144.19.75.0.

Notez que si le réseau 144.19.75.0 est un réseau à diffusion, le routeur effectue une résolution ARP classique pour déterminer l'adresse matérielle de 144.19.75.1 avant de transmettre le datagramme à 144.19.75.1.

Le logiciel de proxy ARP qui s'exécute sur le routeur renvoie une réponse proxy ARP à tous les hôtes qui ne reconnaissent pas les sous-réseaux. Dans l'exemple de la Figure 9.9, si tous les hôtes ont des logiciels TCP/IP ne reconnaissant pas les sous-réseaux, le service proxy ARP sera utilisé pour tous les datagrammes destinés à l'autre sous-réseau.

Dans les paragraphes précédents vous avez peut-être noté que le routeur renvoie sa propre adresse matérielle en lieu et place de celles de tous les hôtes de l'autre sous-réseau. En conséquence, les tables de cache ARP des hôtes qui ne reconnaissent pas les sous-réseaux contiendront plusieurs entrées d'adresses IP correspondant à la même adresse matérielle (celle du port du routeur). Certaines implémentations ARP n'y voient pas d'inconvénient, mais d'autres peuvent en conclure qu'un hôte – le routeur en l'occurrence – essaye de se faire passer pour d'autres nœuds en changeant son adresse IP. Ce type d'usurpation d'identité (*spoofing*) constitue une atteinte à la sécurité du système. Les logiciels ARP conçus pour signaler à l'administrateur toute tentative détectée de spoofing enverra un message d'alerte s'il découvre plusieurs entrées de table ARP utilisant la même



adresse matérielle. Si proxy ARP est utilisé dans un réseau, il y a risque d'une multiplication de messages d'alerte concernant des tentatives de spoofing qui n'en sont pas. Cette situation se produit lorsque les conditions suivantes sont remplies :

- L'hôte utilise un vieux logiciel TCP/IP ne reconnaissant pas les sous-réseaux.
- L'implémentation ARP de l'hôte sait détecter le spoofing.
- L'hôte envoie des datagrammes à plusieurs hôtes appartenant à l'autre sous-réseau.

Proxy ARP part du principe que les réseaux à connecter utilisent ARP pour la résolution des adresses IP en adresses matérielles. En conséquence, les réseaux doivent être des réseaux à diffusion. Si plusieurs routeurs sont connectés au même réseau, un seul routeur doit être configuré avec proxy ARP. En effet, si plusieurs routeurs répondent à une requête ARP, on sera confronté à un problème d'adresses dupliquées, puisque deux nœuds différents renverront leur propre adresse IP suite à une requête ARP pour une seule adresse IP.

Proxy ARP est aussi appelé *ARP hack* ("bidouille" ARP) ou *promiscuous ARP* (ARP libidineux). Il est décrit dans la RFC 1027.

• Routeurs transparents

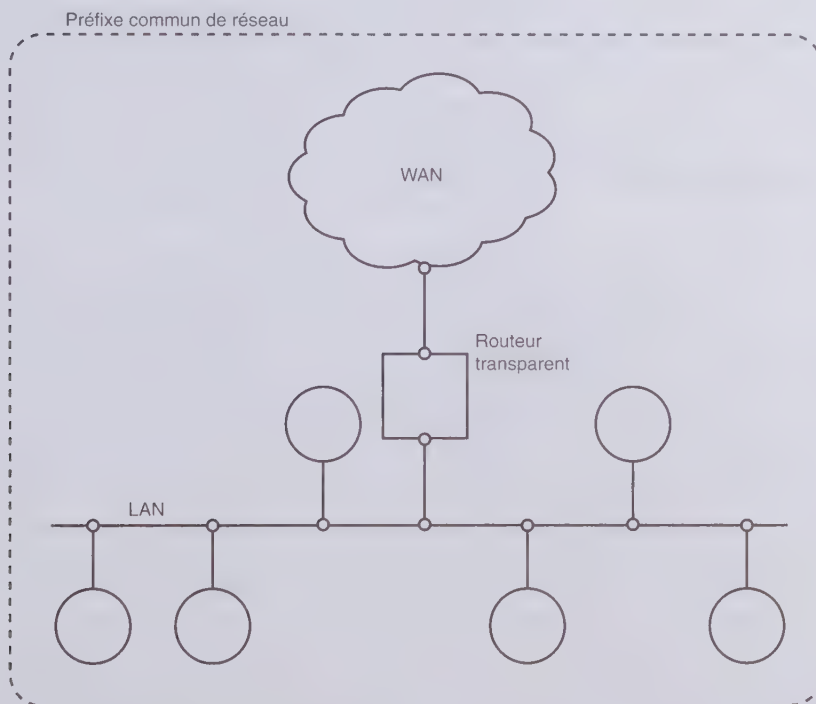
L'utilisation de routeurs transparents est une technique née avec l'ancêtre ARPAnet, qui permettait d'étendre une adresse de classe A au-delà de la frontière d'un routeur (voir Figure 9.10).

A la Figure 9.10, le routeur transparent fait en sorte que les hôtes du réseau local semblent être des hôtes connectés au réseau étendu. Les hôtes qui envoient des datagrammes aux hôtes du réseau étendu n'ont pas conscience de l'existence du "routeur transparent". Comme son nom le suggère, le routeur transparent connecte le réseau local au réseau étendu de manière transparente, comme s'ils faisaient tous deux partie du même réseau. Le réseau étendu et le réseau local partagent un même préfixe de réseau. D'une certaine manière, cette configuration est similaire à celle des sous-réseaux, ces derniers partageant un même ID de réseau avec des numéros de sous-réseaux différents.

Le routeur transparent peut utiliser une table d'adresse des hôtes du réseau local pour déterminer quel hôte de ce réseau doit recevoir le datagramme.

Pour ce qui concerne ARPAnet, les routeurs transparents utilisent une partie de l'adresse IP pour encoder les informations permettant de démultiplexer le datagramme vers l'hôte idoine. ARPAnet utilise l'adresse de réseau de classe A 10.0.0.0. Une adresse sur ce réseau est interprétée comme suit :

10.P.U.I

Figure 9.10*Un routeur transparent.*

Le 10 correspond à l'adresse du réseau, en l'occurrence l'adresse de classe A affectée à ARPAnet. Le P correspond au numéro de port du nœud à commutation de paquets de destination (PSN), lui-même identifié par I, le dernier octet de l'adresse. L'octet U n'est pas interprété par les nœuds PSN, mais par les routeurs transparents afin de déterminer quel hôte du réseau local doit recevoir le datagramme.

Prenons, par exemple, les adresses IP 10.4.1.40 et 10.4.2.40. Ces deux adresses concernent le nœud PSN 40 et au port 4 de ce nœud PSN. Un routeur transparent peut être connecté à ce port. Le troisième octet, U, est utilisé par le routeur transparent pour décider quel hôte doit recevoir le datagramme.

Les routeurs transparents ne peuvent pas héberger tous les services pris en charge par des routeurs classiques. En effet, les routeurs transparents ne possèdent pas d'adresse IP qui permettrait d'accéder à ces services. Ainsi, par exemple, les routeurs classiques peuvent fournir des services de protocole de gestion de réseau tels que SNMP, ou des services de connexion à distance tels que Telnet, ou encore des



services de diagnostic tels que ICMP. Ces services ne peuvent être fournis par des routeurs transparents, car il n'existe pas de méthode simple pour y accéder à distance.

Surréseaux

Le concept de sous-réseau a été conçu en 1985 pour optimiser l'utilisation de l'espace d'adressage IP. Ce concept est bien adapté pour des réseaux disposant d'un vaste espace d'adressage, tels que les réseaux de classes A et B, mais ces formats d'adresse sont très courus et leur nombre ne cesse de diminuer. Il est aussi possible d'utiliser des sous-réseaux pour des adresses de classe C, mais une telle adresse ne peut prendre en charge que 254 hôtes. La plupart du temps, la division d'un réseau de classe C n'est pas praticable, en particulier lorsque le nombre d'hôtes devant être pris en charge par chaque sous-réseau dépasse 126.

Les adresses de classes A et B viennent à manquer, mais les adresses de classe C disponibles sont encore en nombre appréciable. Les organisations de grande taille qui doivent prendre en charge plus de 254 hôtes se voient forcés d'utiliser plusieurs adresses de réseau de classe C.

Supposons qu'une organisation ait besoin d'une adresse de classe B pour prendre en charge 65 534 hôtes. Si le NIC tarde à offrir cette adresse, combien d'adresses de classe C seront nécessaires pour prendre en charge ces 65 534 hôtes ? Environ 256. Ces 256 adresses de classe C peuvent être affectées en bloc. Ainsi, l'intervalle d'adresses de classe C ci-après prend en charge le même nombre d'adresses qu'une seule adresse de classe B :

202.100.0.0 à 202.100.255.255

En regardant le nombre de bits susceptibles de varier dans cette adresse de classe C, on trouve 16 bits de poids faible, c'est-à-dire autant que nécessaire pour une adresse de classe B.

La mise en surréseau, ou adressage en surréseau, consiste à affecter un bloc d'adresses de classe C plutôt qu'une adresse de classe B unique, afin de créer une classe d'adresses virtuelle située à mi-chemin entre un réseau de classe C et un réseau de classe B. Cette technique garantit une utilisation plus rationnelle des adresses. Ainsi, si une organisation a besoin d'un réseau prenant en charge 8 000 hôtes, il est préférable d'affecter un bloc de 32 adresses de classe C plutôt qu'une unique adresse de classe B. L'adresse de classe B peut concerner 65 534 hôtes ; il y aurait donc ici un gâchis de 57 534 adresses (65 534 – 8 000).

L'adressage en surréseau est conçu pour les fournisseurs d'accès Internet (FAI). Généralement, seuls les ISP obtiennent de grands blocs d'adresses de classe C. Il leur est ensuite possible d'allouer de plus petits blocs de ces adresses de classe C à d'autres organisations souhaitant connecter un grand nombre de leurs machines à l'Internet. De nombreuses organisations commerciales utilisent aussi l'adressage en surréseau avec leurs adresses de classe C. Par ailleurs, l'adressage en surréseau réduit la taille des tables de routage.

Classless Internet Domain Routing (CIDR)

L'allocation de blocs d'adresses de classe C évite que les adresses de classe B ne s'épuisent trop vite. Cependant, cette allocation nécessite que des entrées de table de routage supplémentaires soient stockées dans les routeurs. Comme un bloc de 256 adresses de classe C est nécessaire pour prendre en charge l'équivalent d'une seule adresse de classe B, une table de routage doit contenir 256 adresses de réseau pour chacun des réseaux de classe C. Le nombre d'entrées est ici multiplié par 256. Si un routeur dispose de 2 Mo pour sa table de routage en utilisant des adresses de classe B, le remplacement de toutes ces adresses par des adresses de classe C nécessite une mémoire de 2×256 soit 512 Mo !

La technique du Classless Internet Domain Routing (CIDR ou "routage de domaine Internet sans classe") permet de résumer un bloc d'adresses de classe C en une seule entrée de table de routage. Cette entrée de table est ainsi constituée :

(plus basse adresse du bloc, masque de surréseau)

La plus basse adresse du bloc est l'adresse de départ de ce bloc, et le masque de surréseau spécifie le nombre d'adresses de classe C dans le bloc. Le masque de surréseau, aussi appelé *masque CIDR*, contient des 1 pour le préfixe commun à toutes les adresses de classes C, et des 0 pour les parties propres à chacune des classes. Prenons l'entrée de table de routage CIDR suivante :

(200.1.160.0, 255.255.224.0)

L'adresse 200.1.160.0 et le masque CIDR 255.255.224.0 sont représentés en binaire comme suit :

11001000 00000001 10100000 00000000

11111111 11111111 11100000 00000000

les 1 du masque CIDR correspondent au préfixe commun suivant :

11001000 00000001 101

20 1



les 0 du masque CIDR correspondent à la partie variable du bloc d'adresses de classe C. En conséquence, l'intervalle d'adresses de classe C est délimité par les adresses suivantes :

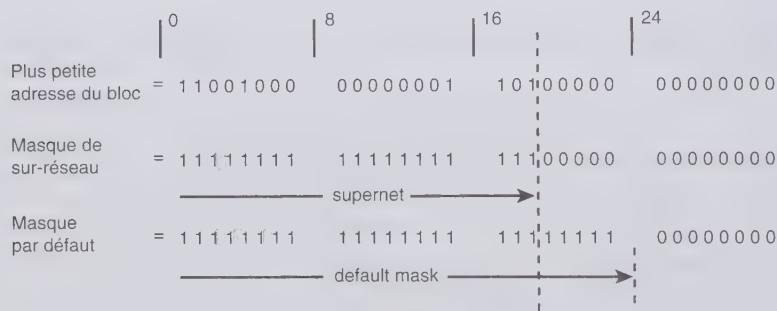
$$11001000\ 00000001\ 10100000\ 00000000 = 200.1.160.0$$

$$11001000\ 00000001\ 10111111\ 11111111 = 200.1.191.255$$

La Figure 9.11 illustre ce bloc d'adresses CIDR.

Figure 9.11

Une illustration du bloc CIDR.



Combien d'adresses de classe C contient ce bloc ? Pour le savoir, examinons la partie réseau de l'adresse de classe C et notons le nombre de bits pouvant varier dans l'intervalle. On en trouve cinq (indiqués en gras ci-après) dans le masque CIDR suivant :

$$11111111\ 11111111\ 111**00000**\ 00000000$$

En conséquence, le nombre d'adresses de classe C dans l'intervalle est de $2^6 = 32$. Les routes d'un bloc CIDR peuvent être résumés en une unique entrée de table de routage appelée un *agrégat*.

Le Tableau 9.4 montre le nombre d'adresses contiguës de classe C qu'il est nécessaire d'affecter pour une taille de réseau donnée. Ce nombre doit être une puissance de 2.

Tableau 9.4 : Nombre d'adresses de classe C en CIDR, en fonction de la taille du réseau

Taille du réseau (en nb d'hôtes)	Nombre d'adresses de classe C
254	1
255 à 508	2
509 à 1016	4
1017 à 2032	8

Tableau 9.4 : Nombre d'adresses de classe C en CIDR, en fonction de la taille du réseau (suite)

Taille du réseau (en nb d'hôtes)	Nombre d'adresses de classe C
2033 à 4064	16
4065 à 8128	32
8129 à 16,256	64

L'autre notation utilisable pour les blocs CIDR est la suivante :

Plus basse adresse du bloc/nombre de bits de préfixe commun

Le nombre de bits de préfixe commun est le nombre de 1 dans le masque de sur-réseau. En conséquence, les exemples suivants sont deux représentations équivalentes du bloc CIDR :

(200.1.160.0, 255.255.224.0)

200.1.1.160.0/19

Le masque de sous-réseau par défaut d'une adresse de classe C est 255.255.255.0, ce qui indique que les hôtes d'une seule adresse de classe C ont 24 bits de préfixe en commun. Lorsque ce nombre de bits est inférieur à 24, c'est l'adressage en sur-réseau qui est utilisé.

On dit que les réseaux qui sont des sous-ensembles d'un bloc CIDR sont *plus spécifiques* par rapport au bloc CIDR. La longueur de préfixe commune aux adresses plus spécifiques est supérieure à celle du bloc CIDR. On peut voir ci-après un exemple de bloc CIDR et de route plus spécifique :

199.22.0.0/16 bloc CIDR avec une longueur de préfixe de 16 ;

199.22.176.0/20 longueur de préfixe plus spécifique de 20.

Les routeurs qui reconnaissent CIDR utilisent l'occurrence la plus longue (*longest match*) pour sélectionner la route. Le masque de CIDR est employé pour déterminer le nombre de bits de préfixe que l'on considère se trouver dans l'occurrence. Toutefois, si plus d'une route vers la destination figure dans la table, le routeur sélectionne la route de plus long préfixe. Prenons l'exemple du réseau de la Figure 9.12. Le routeur contient le routeur R1 qui reçoit deux mises à jour de routage :

199.22.0.0/16 en provenance du routeur R2 ;

199.22.176.0/20 en provenance du routeur R3.

• • •

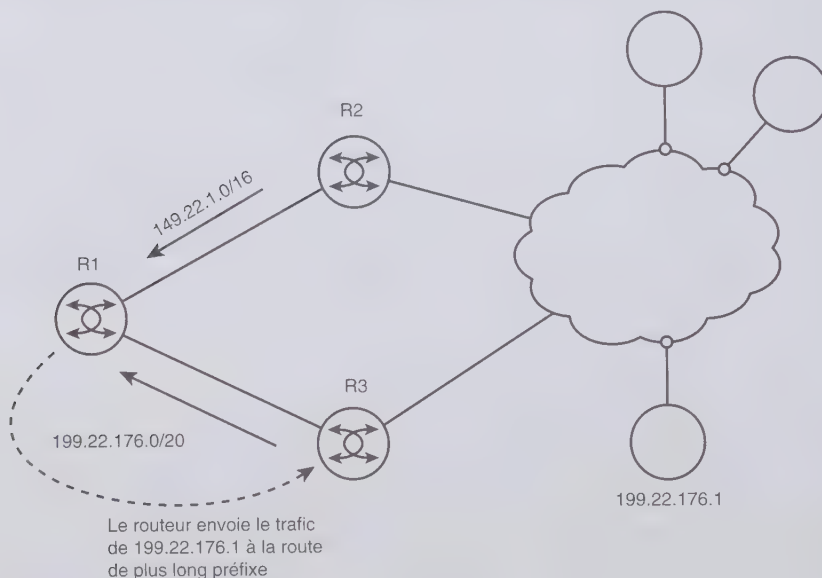


Le routeur R1 sélectionne le chemin passant par R3, car il s'agit du chemin de plus long préfixe.

Le document RFC 1519 traite de cette technique de CIDR.

Figure 9.12

La route du plus long préfixe.



Diffusion et multidiffusion

La *diffusion* (*broadcast*) consiste à envoyer une copie du datagramme à tous les nœuds d'un réseau. La *multidiffusion* (ou *multicast*) consiste à envoyer une copie du datagramme à un groupe de nœuds du réseau. L'existence de sous-réseaux peut compliquer la diffusion et la multidiffusion.

La diffusion peut constituer une façon efficace de communiquer la même information à tous les nœuds d'un réseau. Toutefois, si tous les nœuds n'ont pas besoin de cette information, la diffusion ne fera qu'augmenter le trafic du réseau et la charge de travail des nœuds non concernés par le message ainsi émis. La multidiffusion est à cet égard plus efficace. Avec cette technique, seul un groupe sélectionné de nœuds reçoit les informations. Pour que les multidiffusions soient possibles sur un réseau, la couche de liaison de données de ce réseau doit être capable de les traiter à son niveau. Les réseaux locaux IEEE offrent la possibilité de la multidiffusion au niveau de la couche de liaison de données.

• Diffusion dans des sous-réseaux

Une diffusion est caractérisée par la présence exclusive de 1 dans la partie hôte de l'adresse de destination. Ce type de diffusion est appelé *diffusion dirigée*. Elle doit être envoyée à tous les nœuds du réseau. L'adresse de diffusion peut être représentée sous la forme suivante :

{ numéro de réseau, -1 }

Le *numéro de réseau* est la partie numéro de réseau de l'adresse de réseau de classe A, B ou C. Le -1 correspond à des bits à 1. Lorsqu'on utilise des sous-réseaux, une diffusion vers un sous-réseau spécifique est représentée ainsi :

{ numéro de réseau, numéro de sous-réseau, -1 }

Le *numéro de sous-réseau* est la partie sous-réseau de l'adresse de réseau. Si une diffusion doit être remise à tous les sous-réseaux d'un réseau donné, l'adresse de diffusion est représentée ainsi :

{ numéro de réseau, -1, -1 }

Dans le cas de diffusions remises au sein de sous-réseaux, les routeurs qui interconnectent les sous-réseaux doivent s'accorder pour propager les datagrammes vers tous les réseaux physiques. Un problème peut survenir alors lorsqu'un datagramme de diffusion se retrouve dans une boucle de routage. Cela peut découler de la mauvaise implémentation d'un routeur envoyant sans distinction un datagramme de diffusion sur tous ses ports.

La Figure 9.13 montre un exemple de boucle de routage de diffusion, dans lequel les routeurs transmettent aveuglément le datagramme de diffusion à toutes les interfaces de réseau partageant le même préfixe de numéro de réseau, 132.27 en l'occurrence. En règle générale, les routeurs ne transmettent pas les diffusions à moins d'avoir été spécialement configuré en ce sens. Dans cet exemple, on suppose que les routeurs ont été configurés ainsi. Notez que ces routeurs ne tiennent pas en compte le fait d'avoir ou non précédemment reçu des diffusions. Cette absence de discrimination peut résulter en une boucle de diffusion, qui fera que les sous-réseaux physiques recevront des diffusions multiples. Au bout d'un temps, le champ TTL de l'en-tête IP des datagrammes de diffusion atteindra zéro, et les datagrammes seront écartés du réseau, non sans avoir auparavant provoqué la multiplication des diffusions et l'augmentation sensible du trafic sur le réseau.

Pour éviter l'apparition de boucles de routage, les routeurs doivent utiliser la transmission par chemin inverse. Pour un routeur, cette technique consiste à consulter sa propre table de routage pour déterminer quel port doit être utilisé pour l'envoi

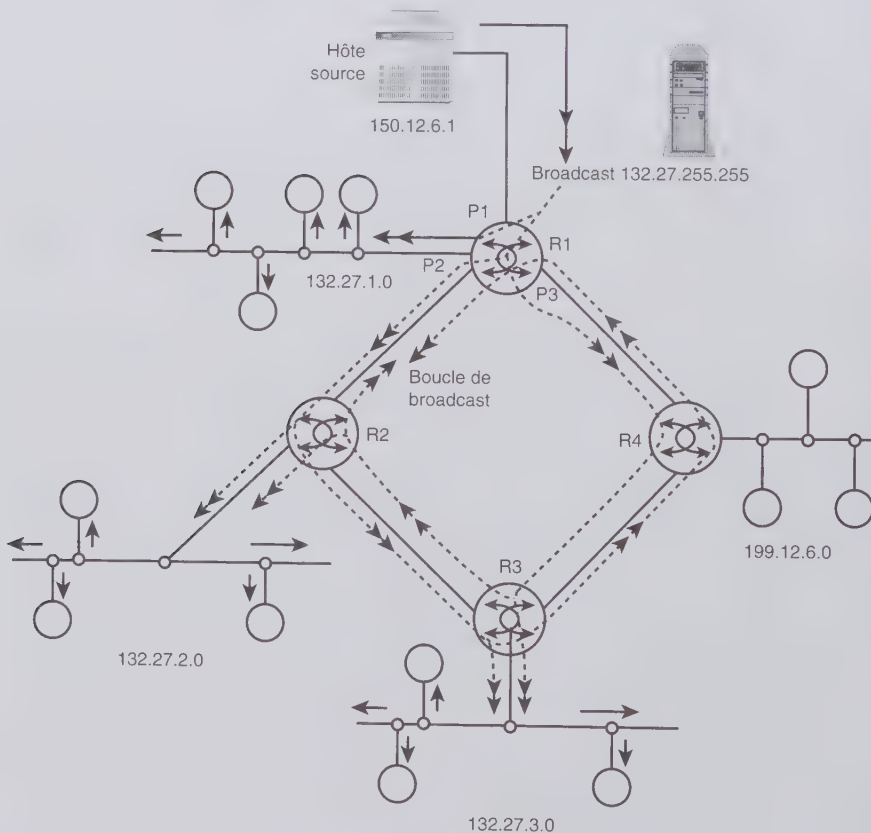


du datagramme vers l'adresse source figurant dans le datagramme de diffusion. Le routeur n'acceptera le datagramme de diffusion que s'il provient de ce port. Le routeur écarte tout datagramme de diffusion provenant d'autres ports. Dans l'exemple de la Figure 9.13, la table de routage du routeur R1 révèle que la source du datagramme de diffusion peut être atteinte par le biais du port P1. Le routeur ignorera donc tout datagramme de diffusion reçu sur les ports P2 et P3, en provenance de la source.

Lorsqu'on utilise le même masque de sous-réseau pour tous les sous-réseaux d'un réseau, l'interprétation de l'adresse de diffusion est sans ambiguïté. En revanche, lorsque plusieurs masques de sous-réseaux sont utilisés, rien ne garantit qu'un routeur donné interprétera correctement l'adresse de diffusion présente dans le datagramme. La diffusion sur les sous-réseaux est traitée dans la RFC 922.

Figure 9.13

Boucle de routage pour les diffusions.



• Multidiffusion

La multidiffusion IP permet d'envoyer un datagramme à un groupe sélectionné de nœuds. Le format d'adresse de classe D est réservé pour les multidiffusions. Les membres d'un groupe de multidiffusion se joignent au groupe de façon dynamique et peuvent le quitter à tout moment. Une adresse de multidiffusion est comparable à une station de radio opérant à une fréquence donnée. Les récepteurs radio peuvent se régler sur la fréquence du canal pour entendre la diffusion de la station à cette fréquence, et peuvent aussi changer de fréquence. Dans les réseaux IP, cette fréquence correspond à une adresse de classe D spécifique.

Il est nécessaire d'appartenir à un groupe de multidiffusion pour recevoir le datagramme de multidiffusion envoyé à l'adresse de classe D correspondant à ce groupe. Un hôte peut être simultanément membre de plusieurs groupes de multidiffusion. N'importe quel hôte peut envoyer une multidiffusion à une adresse de groupe de multidiffusion. Il n'est pas nécessaire de faire partie d'un groupe pour envoyer une multidiffusion.

Certaines adresses de multidiffusion sont attribuées par IANA et sont bien connues (voir Chapitre 4). Les adresses de multidiffusion disponibles pour une utilisation temporaire sont appelées *groupes temporaires de multidiffusion*.

Pour envoyer une multidiffusion à des sous-réseaux ou réseaux connectés par des routeurs, il est nécessaire que les routeurs soient capables de transmettre des datagrammes de multidiffusion. Les routeurs conventionnels peuvent être configurés en ce sens. Lorsque la multidiffusion est remise au réseau physique, le matériel physique sous-jacent est mis à contribution pour envoyer la multidiffusion sur ce réseau physique. L'établissement de correspondance entre les adresses de multidiffusion et les adresses de multidiffusion de couche Physique pour Ethernet est détaillé au Chapitre 4.

Un nœud IP peut participer à une multidiffusion à trois niveaux :

- **Niveau 0.** Le nœud IP ne peut envoyer ou recevoir une multidiffusion IP.
- **Niveau 1.** Le nœud IP peut envoyer, mais ne peut recevoir une multidiffusion IP.
- **Niveau 2.** Le nœud IP peut envoyer et recevoir une multidiffusion IP.

La plupart des implémentations IP peuvent opérer aux niveaux 0 et 1. Les opérations de niveau 2 nécessitent que le nœud IP soit en mesure de recevoir les multidiffusions. La section qui suit s'attache à ce point plus en détail.



• Internet Group Management Protocol (IGMP)

Un hôte rejoint ou quitte un groupe de multidiffusion sur la base de son interface de réseau. En conséquence, un hôte multiconnecté possédant plusieurs connexions de réseau peut se joindre à un groupe pour certaines interfaces de réseau et pas pour d'autres. Généralement, un client de la couche IP, tel qu'un protocole de routage ou une autre application de haut niveau, envoie une requête pour se joindre à un groupe de multidiffusion ou pour le quitter, en précisant une interface de réseau spécifique. Il est utile de spécifier une interface de réseau, car l'application pourrait souhaiter pouvoir envoyer des multidiffusions à des nœuds d'un réseau, mais pas à ceux d'un autre.

Les hôtes qui rejoignent ou quittent un groupe de multidiffusion utilisent le protocole Internet Group Management Protocol (IGMP ou "protocole de gestion de groupe Internet") pour faire savoir aux routeurs avoisinants à quels groupes ils appartiennent. IGMP est implémenté par le module IP, mais utilise ce dernier pour la transmission des messages. IGMP est à cet égard similaire à ICMP (voir Figure 9.14).

La Figure 9.15 montre le format de message IGMP.

Figure 9.14

Schéma conceptuel des couches IGMP.

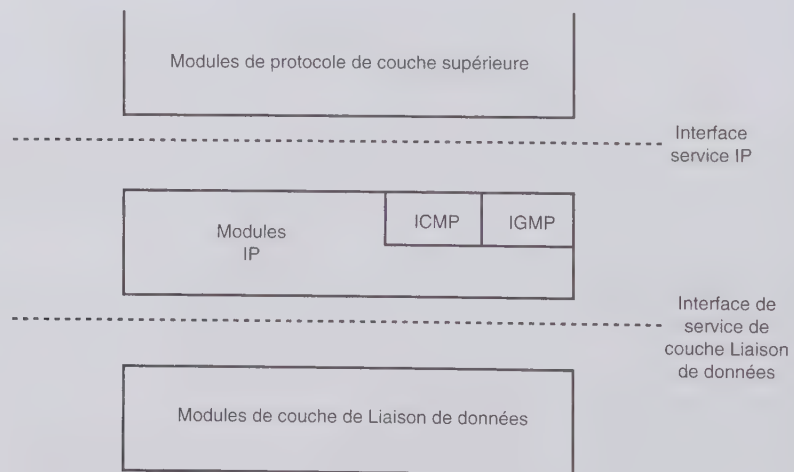
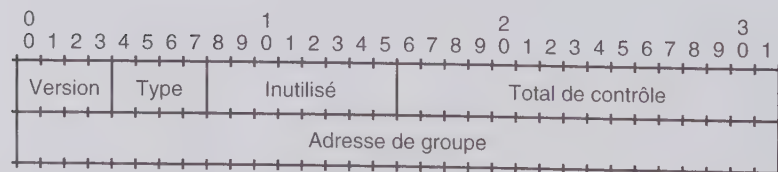


Figure 9.15

Format de message IGMP.



Le champ Version est le numéro de version du protocole et vaut pour le moment

1. Le champ Type indique si le message IGMP est une requête envoyée par un routeur de multidiffusion ou une réponse envoyée par un hôte suite à une requête :

1 = requête d'appartenance de groupe

2 = rapport (réponse) d'appartenance de groupe

Le champ Unused ("inutilisé") doit être mis à 0 lors de l'envoi et est ignoré à la réception. Ce champ est, comme son nom l'indique, inutilisé.

Le champ de total de contrôle est l'habituel complément à 1 de la somme du complément à 1 du message IGMP de 8 octets.

Dans le cas d'un message de requête d'appartenance, le champ Group Address ("adresse de groupe") est à 0 lors de l'émission et ignoré lors de la réception. Dans le cas d'un message de rapport d'appartenance, le champ Group Address contient l'adresse de groupe d'hôte IP du groupe concerné.

Les routeurs de multidiffusion envoient des messages de requête d'appartenance à leurs interfaces de réseau pour déterminer quelles adresses de classe D disposent de membres. La requête IGMP d'appartenance est envoyée à l'adresse de groupe de multidiffusion 224.0.0.1 dans un en-tête IP dont le champ TTL vaut 1. Les hôtes répondent à cette requête en envoyant un message IGMP de rapport d'appartenance. Le message de réponse rapporte les adresses de groupes de multidiffusion dont l'hôte est membre pour l'interface de réseau sur laquelle a été reçue la requête. Pour éviter la simultanéité des messages de réponse et pour en réduire le nombre, on utilise la technique suivante :

- Lorsqu'un hôte reçoit une requête IGMP d'appartenance, l'hôte n'envoie pas immédiatement un message IGMP de rapport. Au lieu de cela, l'hôte lance un temporisateur pour différer le rapport, et ce pour chacune de ses appartenances à un groupe. Les valeurs de temporisateur sont sélectionnées de manière aléatoires, allant de 0 à D secondes (10 secondes, généralement). Lorsqu'un temporisateur expire, un message de réponse est envoyé pour le groupe de multidiffusion correspondant.
- Un message de réponse est envoyé avec une adresse IP de destination égale à l'adresse de groupe concernée par le message, avec un champ TTL égal à 1. Cela permet aux autres membres du même groupe de multidiffusion et du même réseau de recevoir le message de réponse. Si un hôte reçoit un message de réponse pour un groupe auquel il appartient sur ce réseau, l'hôte interrompt son temporisateur pour ce groupe et ne génère pas la réponse correspondante.



Généralement, un seul message de rapport d'appartenance sera généré pour chaque groupe du réseau. Ce message est généré par l'hôte dont le temporisateur expire le premier. Un routeur de multidiffusion n'a pas besoin de savoir quels hôtes appartiennent à un groupe donné. Il faut connaître au moins un hôte appartenant à ce groupe sur un réseau donné. Les routeurs de multidiffusion reçoivent tous les datagrammes de multidiffusion et leur adresse n'a donc pas à être explicitement mise en destination.

Récapitulatif

Les formats d'adresse IPv4 ont été conçus pour s'adapter à des réseaux de différentes tailles. Aux premiers temps de l'Internet, ces formats d'adresse fonctionnaient parfaitement. Toutefois, ce format a montré ses limites au fil du développement de l'Internet.

Le concept de sous-réseau a été conçu et formalisé pour permettre à un numéro de réseau de classe A, B ou C unique d'être subdivisé en plusieurs entités appelées "sous-réseaux". Ces sous-réseaux partagent un même préfixe de réseau. Le nombre de bits utilisés dans la partie hôte de l'adresse IP et permettant d'identifier le sous-réseau est appelé le numéro de sous-réseau. En règle générale, on utilise toujours le même nombre de bits d'hôte pour le numéro de sous-réseau. Cela donne un masque de sous-réseau uniforme. Les surréseaux permettent, quant à eux, de combiner plusieurs adresses de classe C en une adresse unique.

Plutôt que d'affecter des adresses de classe C distinctes qui encombreraient les tables de routage, on utilise une technique d'agrégation des informations de routage. Cette technique a pour nom "CIDR".

On a pu voir le rôle de la multidiffusion dans des réseaux comprenant des sous-réseaux ou des surréseaux. IGMP est le protocole permettant de définir l'appartenance à un groupe de multidiffusion.

Testez vos connaissances

1. Quelle est l'adresse de réseau d'un hôte TCP/IP d'adresse IP 203.23.32.34 ?
2. Quelle est l'adresse de nœud (ou d'hôte) d'un hôte TCP/IP d'adresse 182.23.32.34 ?
3. Convertissez l'adresse IP ci-après en son équivalent en notation décimale à points :

10110101 01011100 10000100 11001100

Combien de masques de sous-réseaux différents sont possibles pour le réseau contenant cette adresse IP ?

4. Quel est le masque de sous-réseau pour 184.231.0.0 si les dix bits d'adresse de nœud sont utilisés pour l'adressage en sous-réseau ?
5. Quels sont les avantages des sous-réseaux ? Quels en sont les inconvénients ?
6. Quelles règles doit respecter la valeur d'un masque de sous-réseau ?
7. Quels sont les masques de sous-réseaux pour les adresses de classes A, B et C indiquant qu'on n'utilise pas d'adressage en sous-réseau ?
8. Dans une RFC, vous trouvez la notation suivante pour décrire un masque de sous-réseau d'une adresse de classe B :

{ -1, -1, 0 }

En utilisant la notation décimale à points, quelle est la représentation de masque de sous-réseau équivalente en partant du principe que trois bits d'hôtes sont utilisés pour les sous-réseaux ?

9. Dans cette question, on suppose que l'adresse IP et le masque de sous-réseau sont les suivants :

- adresse IP = 128.12.34.71 ;
- masque de sous-réseau = 255.255.240.0.

Trouvez les valeurs suivantes :

- numéro de sous-réseau = ?
- numéro d'hôte = ?
- adresse de diffusion dirigée = ?
- adresse de diffusion limitée, pour n'importe quel sous-réseau = ?

10. Dans cette question, on suppose que l'adresse IP et le masque de sous-réseau sont les suivants :

- Adresse IP = 108.77.51.70 ;
- Masque de sous-réseau = 255.255.128.0.

Trouvez les valeurs suivantes :

- Numéro de sous-réseau = ?
- Numéro d'hôte = ?
- Adresse de diffusion dirigée = ?
- Adresse de diffusion limitée, pour n'importe quel sous-réseau = ?



11. Une organisation s'est vu attribuer le numéro de réseau 144.19.0.0. Elle possède trois sites, à Paris, à New York et à Munich. Ces sites doivent être connectés les uns aux autres à l'aide de liaisons WAN (réseau étendu) point à point. Aucun des sites n'aura plus de mille nœuds TCP/IP. Indiquez comment vous utiliseriez des sous-réseaux avec cette affectation d'adresse. Si vous utilisez un masque de sous-réseau uniforme pour toute l'organisation, quel serait-il ? Combien de sous-réseaux sont nécessaires ? Montrez les affectations d'adresses IP pour les ports du routeur et pour au moins deux hôtes de chaque réseau. Comment éliminer la nécessité d'affecter des numéros de sous-réseaux aux liaisons point à point ?
12. Lequel des hôtes ci-après doit utiliser un routeur pour communiquer avec l'hôte 129.23.144.10 si le masque de sous-réseau est 255.255.192.0 ?
129.23.191.21
129.23.127.222
129.23.130.33
129.23.148.127
13. Quelles adresses IP se trouvent sur le même sous-réseau que 130.12.127.231 si le masque de sous-réseau est 255.255.192.0 ?
130.45.130.1
130.22.130.1
130.12.64.23
130.12.167.127
14. Une organisation a demandé puis obtenu le numéro de réseau 133.24.0.0. L'organisation a l'intention d'utiliser des sous-réseaux ne dépassant pas 1 000 nœuds TCP/IP chacun. Quel masque de sous-réseau permettra d'obtenir le maximum de sous-réseaux ? Etapez votre réponse.
15. Une organisation a demandé puis obtenu 181.255.0.0 comme numéro de réseau. Elle souhaite utiliser des sous-réseaux ne dépassant pas deux mille nœuds TCP/IP chacun. Quel masque de sous-réseau lui garantira un maximum de sous-réseaux ? Quel est le nombre maximal de sous-réseaux ? Etapez votre réponse.
16. Une organisation a obtenu le numéro de réseau 190.240.0.0. Elle compte utiliser des sous-réseaux ne dépassant pas cinq cents nœuds TCP/IP chacun. Quel masque de sous-réseau lui garantira un maximum de sous-réseaux ? Quel est le nombre maximal de sous-réseaux ? Pourquoi ?
17. Une organisation a obtenu le numéro de réseau 27.0.0.0. Elle a décidé d'utiliser des sous-réseaux ne dépassant pas cinq mille nœuds TCP/IP chacun. Quel

masque de sous-réseau permettra d'obtenir le nombre maximal de sous-réseaux ? Quel est ce nombre ? Etapez votre réponse.

18. Une organisation a obtenu le numéro de réseau 202.7.17.0. Elle veut utiliser des sous-réseaux ne dépassant pas soixante-quatre nœuds TCP/IP chacun. Quel masque de sous-réseau permettra d'obtenir le maximum de sous-réseaux ? Quel est le nombre maximal de sous-réseaux ? Pourquoi ?
19. Soit une adresse de réseau, 184.84.0.0, devant être divisée en neuf sous-réseaux. Indiquez l'affectation de numéro de sous-réseau et d'hôte qui vous garantira le plus de souplesse pour un éventuel changement du masque de sous-réseau.
20. Une adresse de réseau de 199.233.18.0 comporte quatre bits d'hôtes utilisés pour le numéro de sous-réseau. Pour l'instant, il n'est nécessaire de construire que dix sous-réseaux. Indiquez l'affectation de numéro de réseau et d'hôte qui vous garantira le plus de souplesse pour un éventuel changement du masque de sous-réseau.
21. Supposons que l'on souhaite diviser un réseau de classe C 199.200.30.0 en trois sous-réseaux, avec un maximum de cent hôtes sur un sous-réseau et de cinquante sur les deux autres. Quel masque de sous-réseau utiliser ? Si vous choisissez VLSM, quels seront les masques pour les trois sous-réseaux ?
22. Pourquoi les VLSM ne sont-ils pas plus répandus ?
23. Quelles sont les valeurs des masques de sous-réseau dans une table de routage pour une route spécifique à un hôte et pour une route par défaut ? Pourquoi utilise-t-on ces valeurs ?
24. Si plusieurs routeurs sont connectés au même réseau, seul un d'entre eux doit être configuré avec un logiciel de proxy ARP. Pourquoi ?
25. Supposons que l'on utilise un vieux logiciel TCP/IP sur un réseau Ethernet. Si ce logiciel TCP/IP ne reconnaît pas les sous-réseaux, comment faire pour envoyer des datagrammes de ces hôtes vers d'autres hôtes appartenant à un autre sous-réseau ?
26. Quels sont les avantages et les inconvénients de l'utilisation de proxy ARP ?
27. Pour CIDR, le nombre d'adresses de classe C contiguës doit être une puissance de 2. Pourquoi ?
28. Soit un bloc CIDR décrit par les paramètres suivants :
(200.1.160.0, 255.255.224.0)



Combien d'adresses de classe C seront décrites par ce bloc CIDR ?

29. Soit un bloc CIDR décrit par les paramètres suivants :

203.11.192.0/20

Combien d'adresses de classe C sont-elles décrites par ce bloc ?

30. Un bloc CIDR est composé de l'intervalle d'adresses suivant :

204.23.128.0 à 204.23.255.255

Combien de bits sont utilisés pour le préfixe de réseau commun ? Quel est le masque de surréseau pour ce bloc ?

31. Montrez un exemple d'utilisation de CIDR pour réduire le nombre d'agrégats de routes.

32. Pour chacun de ces nombres d'hôtes devant être présents sur un réseau, donnez la taille de bloc CIDR, en nombre d'adresses de classe C.

- mille hôtes ;
- deux mille hôtes ;
- trois mille hôtes ;
- quatre mille hôtes ;
- cinq mille hôtes.

33. Nommez la technique utilisée pour éviter les boucles de routage lors du traitement des datagrammes de diffusion. Donnez un exemple d'utilisation de cette technique.

34. Quels sont les trois niveaux de participation d'un nœud IP à la multidiffusion ?

35. Pourquoi un hôte rejoint ou quitte un groupe de multidiffusion à l'échelle de l'interface de réseau ?

36. A quoi sert IGMP ? Décrivez le format de message IGMP.

37. Décrivez les techniques permettant de réduire le trafic pour les réponses aux requêtes IGMP.

38. Pourquoi n'est-il pas indispensable pour un routeur de multidiffusion de savoir quels hôtes appartiennent à un groupe de diffusion sur un réseau donné ?

39. Pourquoi le champ TTL est-il fixé à 1 dans l'en-tête IP des messages IGMP ? Est-ce qu'un message ICMP sera généré lorsque le temporisateur TTL deviendra nul ?

10

Protocoles de routage IP

Dans les chapitres précédents, nous avons pu voir les principes à l'œuvre lors du routage ainsi que les algorithmes de routage utilisés au niveau de l'hôte et du routeur. Dans ce chapitre, nous verrons quels sont les protocoles de routage utilisés par les routeurs pour échanger leurs informations. Le concept de système autonome et les protocoles de passerelle intérieure (*Interior Gateway Protocol*, ou IGP) seront présentés. Les mécanismes de protocole de routage à vecteur de distance et à état de liaison seront détaillés à la lumière des protocoles IGP que sont Routing Information Protocol (RIP) et Open Shortest Path First (OSPF).

Systèmes autonomes

Un système d'interréseau est constitué d'un certain nombre de réseaux connectés les uns aux autres. Certains de ces réseaux sont gérés par des organisations distinctes. C'est le cas pour l'Internet, dans lequel différents fournisseurs de services se connectent au réseau dorsal principal. Les fournisseurs de services et les organisations qui se connectent à ces fournisseurs et au réseau dorsal principal sont tous gérés de façon distincte. Dans des organisations de grande taille, il est fréquent que des parties du réseau soient gérées de façon autonome. On peut citer comme raisons :

- différents départements responsables de diverses parties du réseau ;
- une gestion facilitée (réseaux plus petits) ;
- des raisons politiques.

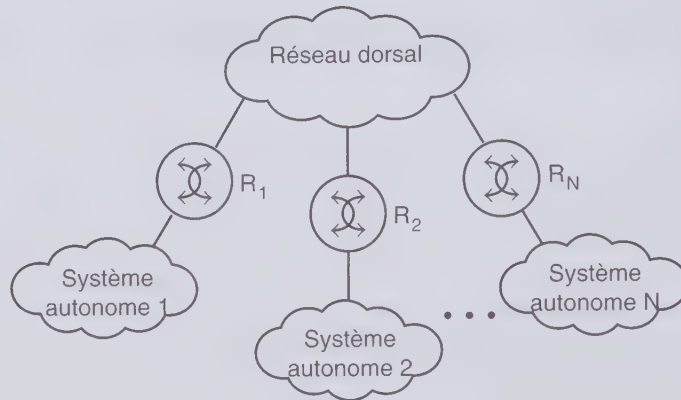
Pour modéliser ce fait, l'architecture Internet définit le concept de *Systèmes autonomes* (SA), consistant en des réseaux gérés indépendamment et connectés à un réseau dorsal (voir Figure 10.1). Du point de vue interne, un SA peut être encore subdivisé (adressage en sous-réseaux) pour faciliter l'administration, mais du point de vue de l'extérieur, le SA semble être un seul réseau logique.

Le réseau dorsal forme lui aussi un système autonome. Les routeurs qui séparent les systèmes autonomes sont appelés des *routeurs de frontière*, car ils se trouvent à la frontière entre les SA.



Figure 10.1

Systèmes autonomes
(source : Learning Tree).



Chaque SA peut sélectionner ses propres protocoles de routage, sur des critères de topologie, de disponibilité de l'équipement, etc. Un SA est libre de changer de protocole de routage à tout moment, sans risquer d'affecter d'autres SA. Le système de réseau dorsal Internet a ainsi plusieurs fois modifié ses protocoles de routage sans jamais avoir affecté les SA qui s'y connectaient.

Selon les époques, on a utilisé sur l'Internet divers protocoles de routage : Gateway to Gateway Protocol (GGP), Exterior Gateway Protocol (EGP) et SPREAD, un protocole SPF (*Shortest Path First*) implémenté par BBN. GGP et EGP sont très peu répandus aujourd'hui et ne seront pas abordés dans ce chapitre. A titre de référence, la RFC 823 présente le protocole GGP et la RFC 904 contient une spécification formelle du protocole EGP. Le protocole SPREAD n'est pas documenté dans les RFC. Les routeurs commerciaux n'implémentent pas SPREAD, car ce protocole n'existe que sur des processeurs spéciaux fabriqués par BBN.

Les routeurs au sein d'un système autonome sont dits intérieurs au système de routeurs. Les protocoles de routage utilisés par les routeurs intérieurs sont appelés *protocoles de passerelle intérieure* (IGP). Les protocoles de routage utilisés entre les SA sont appelés *protocoles de passerelle extérieure* (EGP). Ce dernier acronyme peut prêter confusion, car il correspond aussi à une implémentation de protocole aujourd'hui obsolète. Le terme de protocole Inter-SA (*Inter-AS protocol*) est utilisé dans la RFC 1058 pour décrire le principe de protocole de SA à SA.

Ce chapitre met l'accent sur les protocoles de passerelle intérieure suivants :

- RIP (*Routing Information Protocol* ou "protocole d'informations de routage") ;
- OSPF (*Open Shortest Path First* ou "ouverture du chemin le plus court d'abord").

Routing Information Protocol

Le protocole RIP (*Routing Information Protocol*) est un exemple de la classe de protocoles appelée "protocoles à vecteur de distance". Dans les protocoles à vecteur de distance, le routeur tient à jour dans sa table de routage une liste de toutes les routes connues et diffuse périodiquement le contenu de cette table. Chaque entrée de la table est composée de la destination, de la distance et du routeur de prochain pas.

La *destination* signifie "réseau" ou "hôte de destination". Le *routeur de prochain pas* représente le routeur directement connecté qui sera utilisé pour transmettre le datagramme à la *destination*. La *distance* est la mesure du coût associé à la route, et se mesure en pas (hops) pour RIP (le franchissement d'un routeur comptant pour un pas). D'autres protocoles à vecteur de distance peuvent utiliser d'autres mesures, tels que le temps ou le coût économique, ou une combinaison de ces facteurs. Ainsi, le protocole HELLO qui a été massivement déployé comme IGP sur le réseau dorsal NSFNET utilisait les millisecondes jusqu'à destination comme mesure de la distance. HELLO n'est plus vraiment répandu de nos jours et son intérêt n'est plus qu'historique.

La plupart des protocoles à vecteur de distance affectent une distance nulle au passage à un réseau directement connecté. RIP affecte un compte de pas de 1 pour un réseau connecté, un compte de pas de 2 pour un réseau atteint par le biais d'un routeur, et ainsi de suite.

Lorsqu'un routeur exécutant un protocole à vecteur de distance reçoit une diffusion de table de routage provenant d'un voisin, il recalcule sa table de routage en fonction de ces nouvelles informations. Il fait la somme du coût pour atteindre le routeur voisin et du coût pour atteindre la destination à partir de ce voisin. Le routeur compare alors cette somme avec le coût qui figure déjà dans sa table. Si le nouveau coût est plus faible, l'entrée courante est remplacée par la nouvelle. Supposons qu'un routeur reçoive les informations suivantes provenant d'un voisin K , situé à la distance Nk de lui :

$(Res1, D1)$

La route calculée en fonction de ces informations est la suivante :

$(Res1, D1 + Nk, K)$

Cette route indique que le réseau *Res1* peut être atteint via le routeur K pour un coût de $D1 + Nk$. Si la table de routage ne comprend pas d'entrée pour *Res1*, cette



entrée est ajoutée. Si une entrée existe déjà, on compare le nouveau et l'ancien coût. Supposons que l'entrée existante soit :

(ResI, C, R)

Si $DI + Nk$ est inférieur à C , on utilise la nouvelle route. Sinon, on ne change pas l'entrée existante.

Voici maintenant un exemple plus concret de calcul des entrées. Pour commencer, une table de routage ne comprendra que des entrées pour le réseau directement connecté.

Destination	Distance	Routeur de prochain pas
144.19.0.0	1	(directement connecté)
130.21.0.0	1	(directement connecté)

Le réseau directement connecté est à une distance de 1 (convention RIP). Après avoir reçu les mises à jour de routage provenant des routeurs voisins, la table de routage est remplie de nouvelles entrées. Voici un exemple de ce à quoi ressemble la table de routage après quelque temps :

Destination	Distance	Routeur de prochain pas
144.19.0.0	1	(directement connecté)
130.21.0.0	1	(directement connecté)
200.1.1.0	5	144.19.1.100
200.2.5.0	3	144.19.1.101
199.23.3.0	4	130.21.5.30
131.200.0.0	7	130.21.5.32

Voyons maintenant comment la table de routage est affectée par une diffusion de mise à jour de routage provenant du routeur en 144.19.1.102 contenant les informations suivantes :

Destination	Distance
144.19.0.0	2
200.1.1.0	3
200.2.5.0	1
199.23.3.0	5
131.200.0.0	8

Ces informations sont les informations de routage d'un routeur voisin. Comme un routeur voisin se trouve à un pas du routeur recevant cette mise à jour, les routes calculées correspondent donc à ces valeurs augmentées d'un pas :

Destination	Distance
144.19.0.0	3
200.1.1.0	4 (Meilleure route)
200.2.5.0	2 (Meilleure route)
199.23.3.0	6
131.200.0.0	9

On remarque que les routes ci-après, provenant de la mise à jour, sont meilleures que celles qui figurent actuellement dans la table de routage :

200.1.1.0	4
200.2.5.0	2

Le protocole à vecteur de distance met à jour la table de routage en faisant figurer ces meilleures routes. La table de routage est alors la suivante :

Destination	Distance	Routeur de prochain pas
144.19.0.0	1	(directement connecté)
130.21.0.0	1	(directement connecté)
200.1.1.0	4	144.19.1.102 (Route mise à jour)
200.2.5.0	2	144.19.1.102 (Route mise à jour)
199.23.3.0	4	130.21.5.30
131.200.0.0	7	130.21.5.32

Les protocoles à vecteur de distance ne communiquent leurs informations que périodiquement. En conséquence, il se peut qu'il y ait des retards dans la propagation des mises à jour de routage sur l'ensemble des routeurs du réseau. Les problèmes que posent les protocoles à vecteur de distance sont les suivants :

- Lorsque les routes changent rapidement, le calcul des routes est incessant.



- Du fait de rapides changements dans le réseau et un taux de propagation trop lent, les routeurs peuvent se retrouver avec des informations incorrectes. La conséquence peut en être des routes non optimisées ou des paquets perdus.
- Le calcul des routes se base sur des informations de seconde main. Des informations erronées concernant une route peuvent se propager avec la même efficacité que les informations correctes et les routeurs disposent vite de routes non optimisées ou incorrectes.
- La totalité de la table de routage est remplacée. A mesure que croît le nombre de réseaux, il en va de même de la taille des paquets concernant les informations de routage. De plus, chaque routeur envoie périodiquement ses informations de routage. L'augmentation du nombre de routeurs a pour conséquence celle des paquets contenant des informations de routage. Ces considérations font que les protocoles à vecteur de distance ne sont pas bien adaptés à des réseaux de grande taille, et le trafic du réseau augmente parallèlement à l'augmentation de la taille des informations de routage.

L'algorithme qui calcule la table de routage dans les protocoles à vecteur de distance est parfois appelé algorithme Ford-Fulkerson, du nom de ses créateurs. Cet algorithme est depuis peu baptisé Bellman-Ford parce que le concept de calcul de la table de routage est une application de l'équation de Bellman appartenant à un champ des mathématiques appliquées : la *programmation dynamique*.

Dans la littérature, on utilise aussi le terme *distance vector* ("vecteur de distance") pour les protocoles à vecteur de distance. On trouve aussi l'algorithme de Bellman-Ford sous le nom de *distance vector algorithm* (DVA). Les sections qui suivent traitent des questions soulevées par l'implémentation de RIP.

• Implémentations des protocoles de routage

Malgré son ancienneté, le protocole RIP de TCP/IP reste encore très répandu comme protocole IGP. RIP est issu des travaux du PARC (*Palo Alto Research Center*) de Xerox pour XNS. XNS RIP est devenu la base de développement de nombreux autres protocoles de routage qui ont conservé le nom de RIP. Au début des années 80, de nombreux éditeurs de systèmes d'exploitation – tels que 3COM, Banyan et Novell – développèrent aussi des protocoles de routage basés sur XNS RIP. Malgré quelques différences dans l'implémentation, ces protocoles ont tous été appelés RIP.

Lorsque TCP/IP fut intégré au noyau de la distribution de Unix BSD, les développeurs inclurent une implémentation de XNS RIP modifié pour IP. Le module logiciel qui implémentait le protocole RIP dans Unix BSD reçut le nom de *routed*. L'implémentation de *routed* devint l'implémentation standard de RIP sur de nombreuses autres plates-formes TCP/IP. Les administrateurs de réseau commencèrent à déployer RIP comme IGP sur des réseaux locaux TCP/IP. Alors que les

réseaux grandissaient, les administrateurs continuèrent à utiliser RIP sans se demander si ce protocole était bien adapté à des réseaux de grande taille. On vit ainsi de nombreuses organisations adopter RIP, pour le remplacer par OSPF lorsque ce dernier fit son apparition. C'est pour ces raisons historiques que RIP est encore aujourd'hui très répandu.

Le cas de TCP/IP RIP est intéressant dans la mesure où il a été adopté avant qu'une RFC ne soit proposée pour le protocole. Ce dernier fut adopté comme protocole de facto en raison de la distribution répandue du logiciel Unix BSD. Pendant longtemps, la seule définition de TCP/IP RIP a été l'implémentation de référence du protocole dans le code source de Unix BSD. Cela a entraîné de nombreux problèmes d'interopérabilité sur d'autres plates-formes. Enfin, en 1988, la RFC 1058 (STD 34) fournit une spécification pour le protocole RIP afin d'éliminer ces problèmes. Cela montre bien tout l'intérêt de disposer d'une spécification.

Dans certaines implémentations d'Unix, un programme appelé *gated* est utilisé à la place de *routed*. Le programme *routed* implémente plusieurs protocoles de routage – RIP, HELLO et EGP. Comme les métriques employées par ces divers protocoles sont différentes, il est nécessaire de préciser un facteur de conversion dans le réseau utilisant ces protocoles, afin de convertir le coût de routage d'un système à un autre. Généralement, les fichiers de configuration pour *gated* et *routed* figurent dans */etc/gated.conf* et */etc/routed.conf*, respectivement.

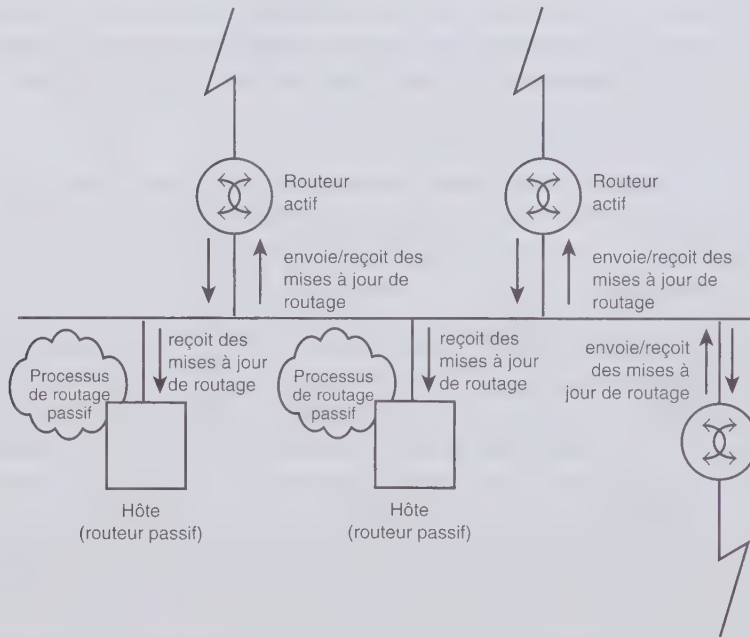
• Routeurs RIP actifs et passifs

Les routeurs RIP sont de deux types : *actif* et *passif*. Les routeurs actifs peuvent transmettre les routes et aussi les recevoir. Les routeurs passifs ne peuvent que recevoir les informations de routage : ils ne les transmettent pas. Les routeurs passifs ne font qu'attendre la réception des informations. En fonction de ces informations, ils calculent leurs tables de routage. Ils ne partagent pas les résultats de leurs calculs (la nouvelle table de routage) avec d'autres routeurs. Seuls les périphériques de routage peuvent tenir lieu de routeurs actifs. Les hôtes ne sont généralement pas des routeurs actifs (voir Figure 10.2).

Si un hôte agit comme un routeur, il effectue les opérations d'un routeur. Le Chapitre 7 traite le cas des hôtes tenant le rôle de routeurs.

Un routeur RIP actif diffuse ses informations de routage toutes les 30 secondes. Dans la plupart des implémentations, l'intervalle entre les diffusions est paramétrable. On utilise généralement un multiple de 30 secondes. Chaque diffusion RIP contient des paires adresse de réseau/nombre de pas.



Figure 10.2*Routeurs actifs et passifs.*

En temps normal, si une mise à jour de routage n'est pas reçue au bout de trois intervalles de diffusion (soit 90 secondes en configuration standard) on considère que le réseau ne peut être atteint, et l'entrée de routage correspondante est supprimée.

• Problèmes posés par le calcul en nombre de pas

Les coûts de chemin basés sur le compte de pas ne sont pas toujours optimaux. En effet, le compte de pas ne prend en compte que le nombre de routeurs nécessaire pour joindre une destination, et laisse de côté d'autres facteurs tels que le coût de la liaison, le délai, la fiabilité, le débit de la liaison, etc. La Figure 10.3 montre deux chemins possibles entre les routeurs R1 et R3. Le chemin direct entre R1 et R3 passe par une liaison lente et coûte un pas.

Le chemin de R1 à R3 via R2 passe par une liaison à haut débit, mais coûte 2 pas. Dans ce cas, c'est le chemin direct entre R1 et R3 qui est sélectionné au détriment de la liaison rapide, car il coûte moins de pas.

Pour résoudre ce problème, de nombreuses implémentations RIP, telles que celle d'Unix, permettent d'attribuer artificiellement un coût très fort à des chemins indésirables. Ainsi, dans la Figure 10.4 on voit que la liaison lente s'est vue artificiellement attribuer un compte de pas de 3. Cette valeur est maintenant supérieure à celle de la liaison à haut débit (valant toujours 2).

Figure 10.3

Sélection du chemin "le moins favorable", en se référant au nombre de pas (source : Learning Tree).

- Les routes basées sur le compte de pas ne sont pas toujours optimales

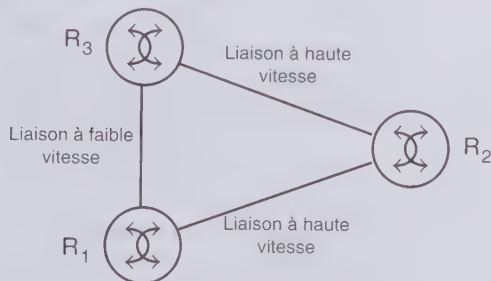
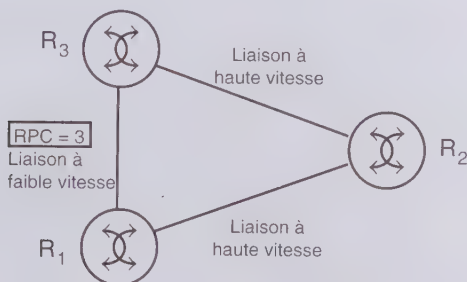


Figure 10.4

Utilisation d'un compte de pas artificiellement élevé pour sélectionner des chemins optimaux (source : Learning Tree).

- On peut artificiellement augmenter le coût de routage des liaisons lentes



- Le coût sur la liaison lente est désormais supérieur à celui de l'autre route

RPC : Route Path Cost

Lorsque RIP doit choisir entre deux chemins de même compte de pas, la sélection se fait arbitrairement.



• Convergence lente avec RIP

Lorsque RIP reçoit une mise à jour de route et met à jour ses entrées de table de routage, il lance un temporisateur pour la route mise à jour. Si la mise à jour n'arrive pas avant expiration du temporisateur, elle est supprimée de la table. Si elle arrive, le temporisateur est relancé.

Avec RIP, les modifications dans les tables de routage peuvent prendre plusieurs cycles avant que tous les routeurs n'aient reçu les informations adéquates. En effet, RIP a un taux de convergence très faible. Ce faible taux de convergence est commun à tous les protocoles à vecteur de distance.

La Figure 10.5 illustre les problèmes liés à la convergence. Dans cette figure (voir Figure 10.5a), on voit le réseau Res1 connecté au routeur R1. Le routeur R2 est connecté au routeur R1 et le routeur R3 au routeur R2. Dans la Figure 10.5b on voit une rupture de la liaison entre R1 et Res1. Lorsque R1 détecte que la liaison vers Res1 est hors service, il affecte 16 au compte de pas pour Res1. Pour RIP, un compte de pas de 16 correspond à "infini" et indique que la destination correspondante n'est pas accessible.

En temps normal, le routeur R1 finirait par diffuser sa table de routage à son voisin R2, à la fréquence de son intervalle de diffusion. Cependant, l'intervalle de diffusion de R2 peut expirer avant que R1 ne puisse envoyer sa mise à jour de routage. R2 était au départ à 2 pas de Res1 avant la mise hors service de la liaison entre Res1 et R1. Si R2 envoie sa mise à jour avant R1, R1 reçoit alors la mise à jour suivante :

(Res1, 2)

La précédente entrée de route est donc remplacée par la suivante (voir Figure 10.5c) :

(Res1,3)

A ce moment, les routeurs R1 et R2 ont les entrées de route suivantes pour Res1 :

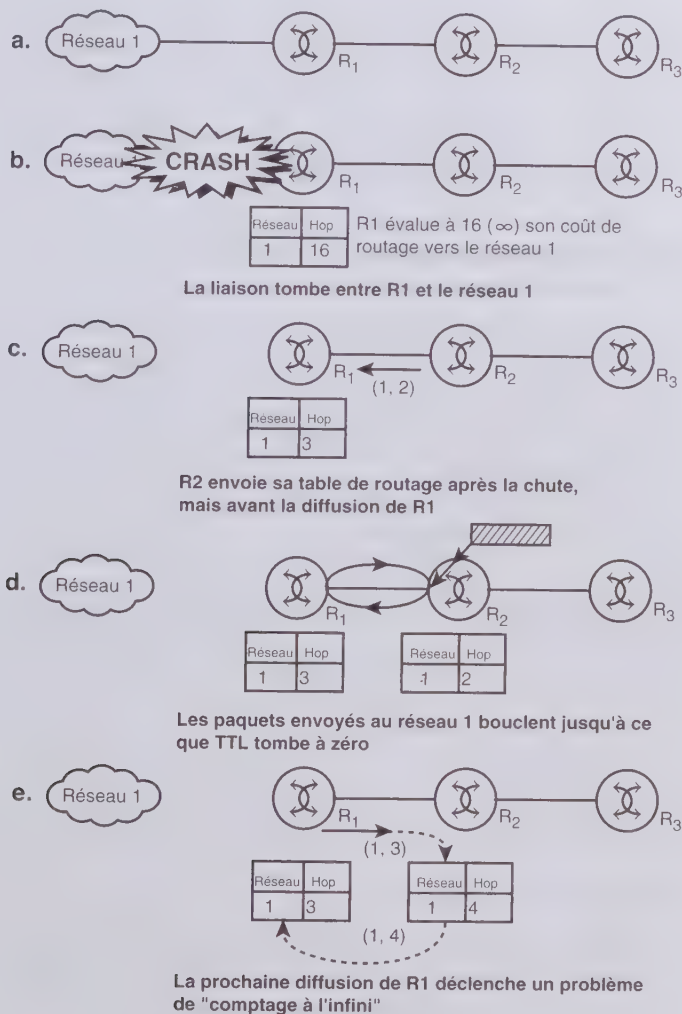
Routeur R1 : (Res1, 3)

Routeur R2 : (Res1, 2)

Si un datagramme IP arrive en R2 pour Res1, R2 le transmet à R1 en pensant que Res1 est à deux pas de distance. Lorsque R1 reçoit ce datagramme pour Res1, il transmet ce datagramme à R2. R2 le retransmet alors à R1. Le datagramme se trouve alors dans une boucle de routage (voir Figure 10.5d) qui se perpétue jusqu'à ce que le champ TTL de l'en-tête IP vienne à expiration. Le datagramme est alors écarté et un message ICMP est envoyé à la source qui a envoyé le datagramme.

Figure 10.5

Le problème du faible taux
de convergence de RIP
(source : Learning Tree).



La diffusion suivante de R1 contient la mise à jour suivante pour R2 :

(Res1, 3)

Le routeur R2 calcule les informations de routage d'après la précédente mise à jour de routage. Comme le routeur voisin R1 est à trois pas de distance de Res1,



R2 doit être à $3 + 1 = 4$ pas de Res1. L'entrée de table de routage de Res1 en R2 devient alors :

(Res1,4)

R2 envoie la route précédente à R1 dans sa prochaine diffusion. R1 met à jour sa table de routage avec les informations suivantes :

(Res1, 5)

Cette augmentation régulière du compte de pas continue indéfiniment. C'est le syndrome du "comptage à l'infini" (voir Figure 10.5e).

Les concepteurs de RIP ont établi un plafond artificiel de compte de pas à 16 pour éviter que les routeurs ne comptent indéfiniment. Lorsqu'un compte de 16 est atteint, le comptage est interrompu et la destination est considérée comme non accessible. Ce plafond implique que RIP ne permet de calculer que des routes passant au maximum par 15 routeurs.

La section qui suit décrit différentes techniques permettant de résoudre le problème de la convergence lente.

• Résoudre le problème de la convergence lente de RIP

Au fil du temps diverses techniques ont été élaborées pour éliminer le problème de la convergence lente de RIP, à savoir :

- split horizon update ;
- hold down ;
- poison reverse.

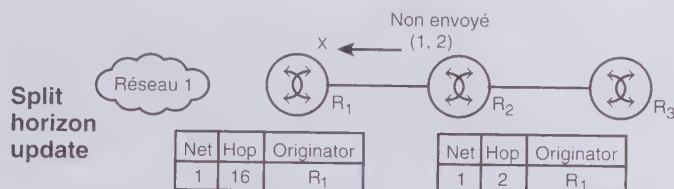
Split horizon update

Dans la technique de *split horizon update*, le routeur enregistre l'interface sur laquelle est reçue une nouvelle route. Cette interface est appelée l'*interface source*, car il s'agit de la source des informations de routage. Le routeur n'envoie pas les informations de routage sur l'interface source. Prenons le problème de la convergence lente rencontré dans la Figure 10.5. Le routeur R2 envoie à R1 une mise à jour de route pour Res1, sans se soucier du fait que R2 a obtenu ses informations de route par R1. Lorsque le routeur R1 reçoit une mise à jour provenant de R2, il fait entière confiance à ces informations bien qu'elles soient erronées, puisque la seule route vers Res1 passe par R1 et la liaison entre R1 et Res1 est hors service.

Si on empêche que les informations de routage ne soient envoyées par le truchement de l'interface source, le problème décrit dans la Figure 10.5 ne peut plus dégénérer (voir Figure 10.6).

Figure 10.6

Split horizon update (source : Learning Tree).



- Ne pas renvoyer les informations de routage à l'expéditeur

Par cette technique, lorsque la liaison entre R1 et Res1 est rompue, le routeur R1 arrête de rendre publique la route. Quand d'autres routeurs du réseau ne reçoivent pas d'informations au bout de trois cycles de mise à jour (90 secondes si l'intervalle de diffusion est de 30 secondes), les routeurs retirent l'entrée de leurs tables. Si un routeur reçoit entre-temps une autre route vers Res1, c'est cette dernière qui sera désormais utilisée. Il est possible que ces informations concernant Res1 s'appuient sur la route qui vient de disparaître. Le routeur qui indique la mauvaise route devra découvrir que la route n'est plus valide et cesser d'en faire mention avant que d'autres routeurs n'agissent en fonction de ces informations erronées.

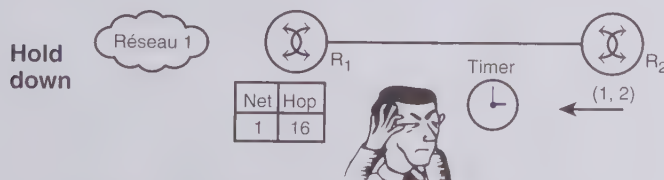
Généralement, après plusieurs cycles de mise à jour RIP, tous les routeurs du réseau prennent conscience de la modification de la topologie du réseau.

Hold down

Dans la technique de *hold down*, les routeurs ignorent pendant un certain intervalle de temps les informations de routage concernant une destination s'ils reçoivent un message indiquant que la destination n'est plus accessible (voir Figure 10.7).

Figure 10.7

Hold down (source : Learning Tree).



- Ignorer les mises à jour de routage concernant des connexions réseau inaccessibles pour une période de latence (60 secondes)



L'intervalle de temps pendant lequel les routeurs ignorent les messages concernant une destination est appelé *hold down time*. Ce temps est généralement égal à deux intervalles de mise à jour (soit 60 secondes). Il permet à la plupart des routeurs de ne pas prendre en compte des informations concernant des routes erronées. Tous les routeurs utilisent la même valeur de hold down time.

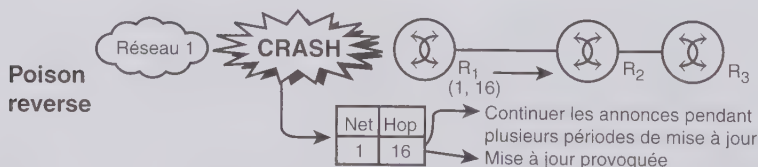
Un des problèmes liés à cette technique est qu'elle ne permet pas de différencier les bonnes informations des mauvaises. Pendant l'intervalle de hold down time, le routeur ignore les informations erronées concernant un routeur aussi bien que les informations valides. En conséquence, les informations erronées sont conservées pendant une longue période même s'il existe des routes valides de remplacement.

Poison reverse

Avec la technique du *poison reverse*, le routeur informant de la non-disponibilité d'une liaison le fait plusieurs fois de suite (voir Figure 10.8). Lorsqu'une destination devient inaccessible du fait de la non-disponibilité d'une liaison, RIP envoie une valeur de 16 ("infini") pour le compte de pas pour cette destination. Cela revient à dire que la route vers la destination passant par un routeur donné est "empoisonnée".

Continuer à spécifier un coût infini pour une destination garantit que tous les routeurs finissent par recevoir le message indiquant que la destination n'est plus accessible.

Figure 10.8
Poison reverse
(source : *Learning Tree*).



Une variante de cette technique est le *poison reverse with triggered update*. Dans ce cas, le routeur n'attend pas l'expiration de son intervalle de diffusion pour faire savoir qu'une destination n'est plus accessible. Le routeur envoie une mise à jour de route dès qu'il détecte qu'une destination n'est plus accessible. Avec cette technique, dans la Figure 10.5, le routeur R1 n'aurait plus besoin d'attendre la fin de l'intervalle de diffusion pour envoyer sa diffusion et R2 recevrait la diffusion et mettrait à jour sa table de routage en conséquence.

Dans certains réseaux, l'utilisation de cette technique peut provoquer une tempête de diffusion (ou *broadcast storm*). Cette technique peut provoquer la modification

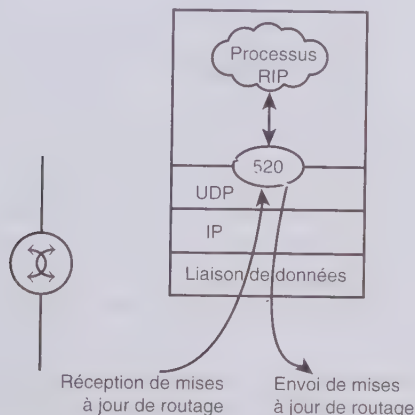
immédiate des tables de routage des routeurs recevant la diffusion. Si ces routeurs utilisent eux-mêmes cette technique, ils répercuteront l'information en envoyant des diffusions vers d'autres routeurs qui réagiront à leur tour, et ainsi de suite, créant au final une tempête de diffusion.

• Format de message RIP

RIP utilise UDP comme protocole de transport. Les routeurs RIP envoient et reçoivent des mises à jour et des requêtes d'informations de routage sur le port UDP numéro 520 (voir Figure 10.9).

Figure 10.9

RIP et son mécanisme de transport.

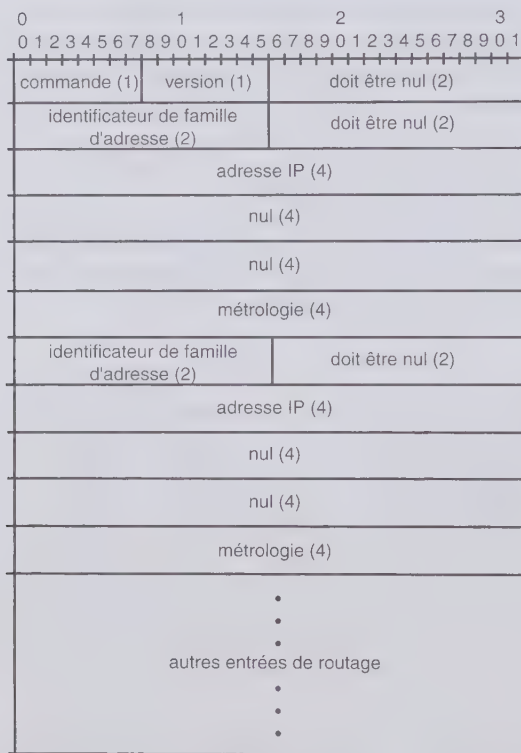


Les messages RIP sont de deux types :

- message d'information de routage ;
- requête pour des informations de routage.

La Figure 10.10 montre le format des messages RIP.



Figure 10.10*Format des messages RIP.*

Le champ Commande spécifie le type d'opération RIP. Le Tableau 10.1 montre les différents types d'opérations RIP reconnus.

Tableau 10.1 : Types de commandes RIP

Type de commande	Description
Requête RIP	Une requête au système cible pour un envoi de tout ou partie de sa table de routage.
Réponse RIP	Un message contenant tout ou partie de la table de routage de l'émetteur. Ce message peut être envoyé en réponse à une requête ou à une interrogation (<i>pool</i>), ou il peut s'agir d'un message de mise à jour généré par l'émetteur.
Trace on	Obsolète. Les messages contenant cette commande doivent être ignorés.

Tableau 10.1 : Types de commandes RIP (suite)

Type de commande	Description
Trace off	Obsolète. Les messages contenant cette commande doivent être ignorés.
Reserved	Cette valeur est utilisée par Sun Microsystems pour son usage propre. Au cas où d'autres commandes seraient ajoutées dans une nouvelle version, elles devront commencer par 6. Les messages contenant cette commande peuvent être ignorés sans problème par les implémentations.

Si les routeurs actifs et passifs peuvent tous deux faire la demande d'informations, ils reçoivent généralement leurs informations par le biais de messages non sollicités envoyés en diffusion par d'autres routeurs.

Le champ Version indique la version du protocole RIP (1).

Le reste du datagramme contient une liste de destinations accompagnées de leur coût. La Figure 10.10 ne montre que deux entrées, mais ces entrées sont répétées pour chaque destination. Souvenez-vous ; une destination peut être un réseau ou bien un hôte spécifique. Quatre octets sont utilisés pour la destination et les 8 octets suivants sont nuls. Ce format de paquet permet à RIP de convoyer des informations pour plusieurs protocoles différents. On a pu voir que RIP est un dérivé de XNS RIP.

Dans XNS RIP, les destinations peuvent avoir des adresses allant jusqu'à 14 octets. Chaque entrée est précédée par un identificateur de famille d'adresse indiquant quel type d'adresse est spécifié dans cette entrée. L'identificateur est suivi par 14 octets réservés pour les valeurs d'adresses. L'identificateur de famille d'adresse pour IP est 2. L'adresse IP est l'adresse Internet classique, stockée avec 4 octets dans l'ordre de réseau et correspond à l'adresse de destination. L'adresse IP est alignée à 32 bits, ce qui explique pourquoi les deux octets d'adresse suivant immédiatement l'identificateur de famille d'adresse valent 0 pour IP. Le champ Métrique doit contenir une valeur comprise entre 1 et 15 inclus, et représente le coût pour atteindre la destination spécifiée. Une valeur de 16 indique que la destination n'est pas accessible.

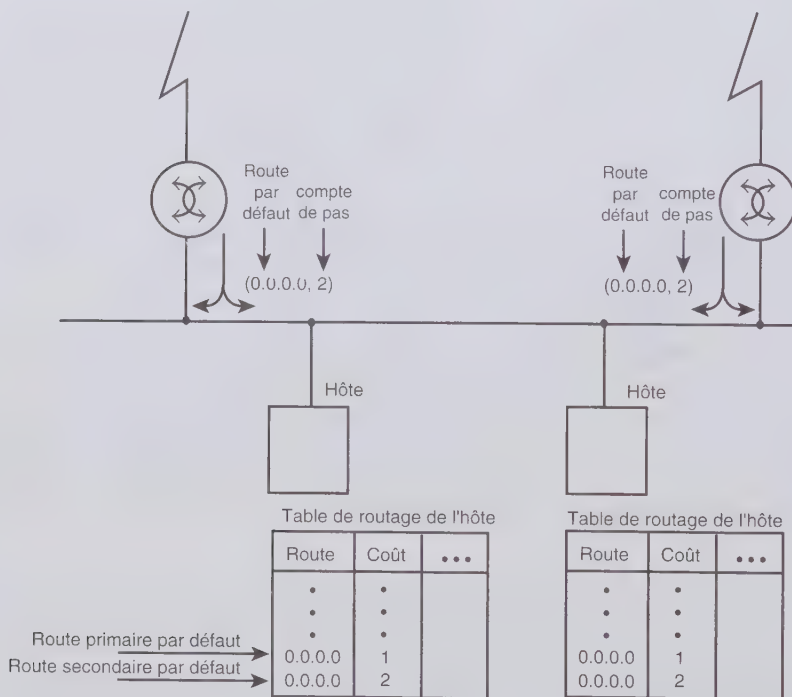
Par convention, une adresse IP de valeur 0.0.0.0 dans le champ d'adresse indique une route par défaut. Si plusieurs routeurs d'un réseau diffusent des routes par défaut, les autres routeurs/hôtes peuvent s'en servir pour faire figurer dans leurs tables des entrées de route par défaut. La route par défaut de coût minimal est utilisée



comme route par défaut principale, les autres routes servant de route de secours (voir Figure 10.11). Cela suppose bien sûr que les routeurs/hôtes sont en mesure de traiter plus d'une route par défaut. Ce n'est pas le cas de toutes les implémentations de TCP/IP.

Figure 10.11

Route par défaut avec RIP.



La taille maximale d'un message RIP est de 512 octets. Cela ne concerne que le message RIP décrit dans la Figure 10.10 et pas les en-têtes IP ou UDP qui l'accompagnent. Si les informations de routage RIP sont trop longues pour figurer dans un seul message, les informations sont réparties sur plusieurs messages. Il n'est pas nécessaire de mettre en place un mécanisme pour gérer le découpage des messages RIP puisque chaque message peut être traité indépendamment.

Lors du routage d'un datagramme, le routeur RIP procède dans un ordre défini pour regarder si l'adresse de destination reçue figure dans sa table :

1. On regarde si l'adresse de destination figure dans la liste des hôtes.
2. On regarde si l'adresse de destination correspond à un numéro de réseau ou de sous-réseau connu.

3. Si on ne trouve aucune correspondance, on utilise la route par défaut.

Lorsqu'un hôte reçoit des mises à jour de route via RIP, sa façon d'interpréter cette adresse varie selon qu'il connaît ou non le masque de sous-réseau s'appliquant au réseau. Si le masque est connu, l'hôte peut déterminer la signification de l'adresse. Supposons que l'hôte reçoive les informations de route suivantes :

(134.3.0.0, 2)

Si l'hôte a un masque de sous-réseau de 255.255.255.0, alors 134.3 est un numéro de réseau. Si c'est l'adresse 134.3.5.0 qui est reçue, alors le numéro de réseau est 0.0.5.0. Si c'est l'adresse 134.3.5.2 qui est reçue, il s'agit alors d'une adresse d'hôte.

Si l'hôte ne connaît pas le masque de sous-réseau, la signification de l'adresse peut être ambiguë. Si l'adresse contient une partie hôte non nulle, il n'y a pas de moyen simple pour déterminer si l'adresse représente un numéro de sous-réseau ou une adresse d'hôte. En l'absence d'un masque de sous-réseau, on considère que les adresses contenant une partie hôte non nulle sont des adresses d'hôte.

• Trace de message RIP

Dans cette section on examinera plusieurs décodages de messages RIP. La bonne compréhension des décodages de ce protocole permettra de mieux vous familiariser au format des messages RIP.

La Figure 10.12 montre un décodage de paquet d'une requête RIP, et la Figure 10.13 montre le décodage de paquet d'une diffusion RIP.

Comme nous l'avons mentionné auparavant, les messages RIP sont encapsulés dans des messages UDP (le Chapitre 12 traite en détail des en-têtes UDP). Un en-tête UDP fait 8 octets de long. Les deux premiers octets de l'en-tête UDP indiquent le numéro de port source et les deux octets suivants le numéro de port de destination. Pour un message de requête ou de réponse RIP (voir Figures 10.12 et 10.13), on peut voir que les ports UDP source et destination valent 520. Il s'agit du numéro de port UDP usuel affecté à RIP.

Vous pouvez constater que la requête RIP (voir Figure 10.12) contient une requête générale demandant toutes les routes possibles. On voit que cette requête est envoyée par 144.19.74.201 sous forme de diffusion dans le réseau 144.19.0.0. L'adresse de diffusion dirigée est 144.19.255.255.



Figure 10.12

Message de requête RIP.

```

Packet Number : 1          10:57:08 PM
Length : 70 bytes
ether: ===== Ethernet Datalink Layer =====
      Station: 00-00-C0-A2-0F-8E ----> FF-FF-FF-FF-FF-FF
      Type: 0x0800 (IP)
ip: ===== Internet Protocol =====
      Station: 144.19.74.201 ----> 144.19.255.255
      Protocol: UDP
      Version: 4
      Header Length (32 bit words): 5
      Precedence: Routine
          Normal Delay, Normal Throughput, Normal Reliability
      Total length: 52
      Identification: 24
      Fragmentation allowed, Last fragment
      Fragment Offset: 0
      Time to Live: 64 seconds
      Checksum: 0x0FB2(Valid)
udp: ===== User Datagram Protocol =====
      Source Port: ROUTER
      Destination Port: ROUTER
      Length = 32
      Checksum: 0x8F9D(Valid)
rip: ===== Routing Information Protocol =====
      Command: Request
      Version: 1
          Family ID: 0
          Address: 0x0
          Distance: Not Reachable

```

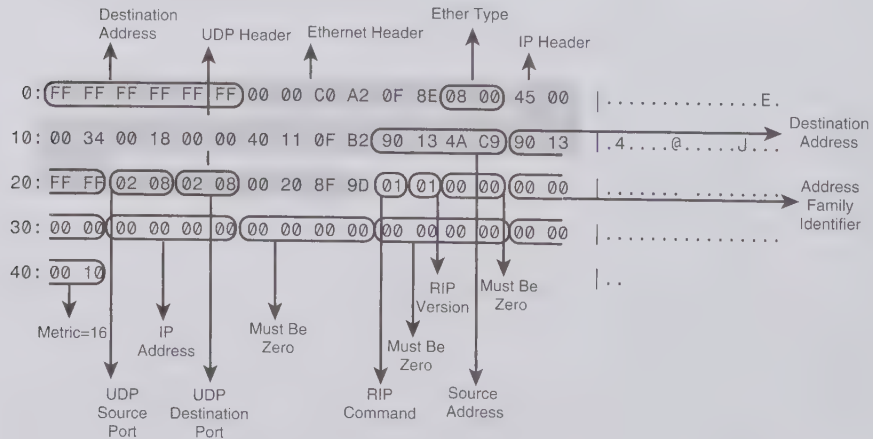


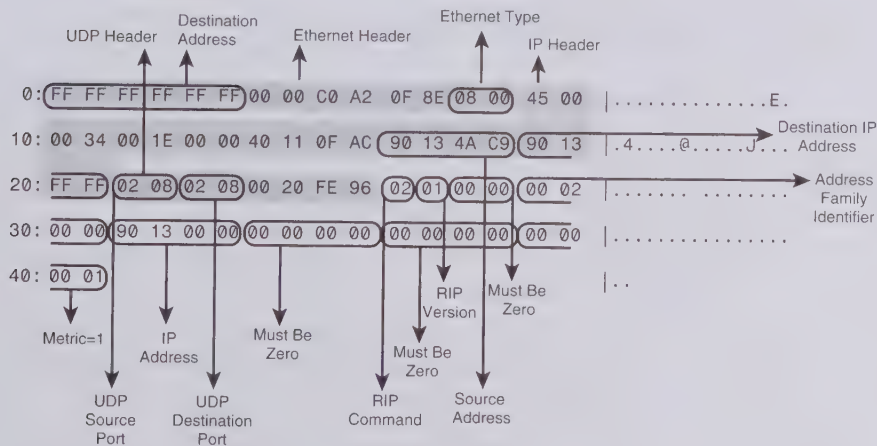
Figure 10.13

Message de réponse RIP
(en diffusion).

```

Packet Number : 2          10:57:57 PM
Length : 70 bytes
ether: ===== Ethernet Datalink Layer =====
      Station: 00-00-C0-A2-0F-8E ----> FF-FF-FF-FF-FF-FF
      Type: 0x0800 (IP)
ip: ===== Internet Protocol =====
      Station: 144.19.74.201 ----> 144.19.255.255
      Protocol: UDP
      Version: 4
      Header Length (32 bit words): 5
      Precedence: Routine
          Normal Delay, Normal Throughput, Normal Reliability
      Total length: 52
      Identification: 30
      Fragmentation allowed, Last fragment
      Fragment Offset: 0
      Time to Live: 64 seconds
      Checksum: 0x0FAC(Valid)
udp: ===== User Datagram Protocol =====
      Source Port: ROUTER
      Destination Port: ROUTER
      Length = 32
      Checksum: 0xFE96(Valid)
rip: ===== Routing Information Protocol =====
      Command: Response
      Version: 1
      Family ID: IP
          IP Address: 144.19.0.0
          Distance: 1

```



Les valeurs des champs du message RIP sont les suivantes :

Command: 1

Version: 1

Family ID: 0

Address: 0x0

Distance: 16 (inaccessible)

En regardant l'en-tête IP de la Figure 10.13, on s'aperçoit que la réponse RIP est envoyée par le routeur 144.19.74.201 sous forme de diffusion dirigée vers le réseau 144.19.0.0. L'adresse de diffusion dirigée est 144.19.255.255.

Les valeurs des champs du message RIP sont les suivantes :

Command: 2

Version: 1

Family ID: 2

IP Address: 144.19.0.0

Distance: 1

Open Shortest Path First

OSPF (*Open Shortest Path First*) est un protocole d'état de liaison qui apporte une solution à la plupart des problèmes posés par RIP. Comme nous l'avons vu lors de notre présentation des protocoles à vecteur de distance, dont le représentant le plus répandu est RIP, ce type de protocole n'est pas bien adapté à des réseaux de grande taille. La taille des messages de routage est proportionnelle au nombre de réseaux, et le nombre de ces messages augmente parallèlement à l'augmentation du nombre de routeurs.

Une technique alternative et maintenant largement répandue est représentée par les protocoles de routage à état de liaison. Les sections qui suivent s'attachent à présenter ces protocoles.

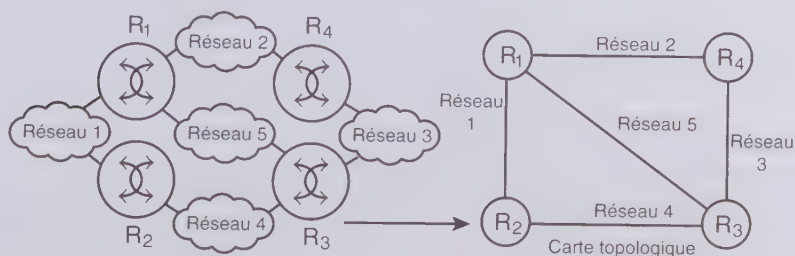
• Protocoles d'état de liaison

Les *protocoles d'état de liaison* sont aussi appelés Shortest Path First (SPF ou *chemin le plus court d'abord*). SPF renvoie en fait à l'algorithme SPF utilisé par les routeurs pour calculer la table de routage. Cet algorithme a été conçu par Dijkstra, un chercheur en informatique néerlandais.

L'algorithme SPF est utilisé par chaque routeur pour calculer une carte interne représentant la topologie de la totalité du réseau. Cette carte contient des informations sur tous les routeurs et réseaux connectés. La Figure 10.14 montre un exemple de carte. Chaque routeur dispose de sa carte complète du réseau. Cette carte est représentée sous forme d'un graphe dans lequel les routeurs sont les nœuds et les réseaux sont les arcs liant les nœuds. Les routeurs sont des voisins s'ils partagent un même arc. Ainsi, dans la Figure 10.14, les voisins du routeur R1 sont R2, R3 et R4, par les liaisons Res1, Res5 et Res2.

Figure 10.14

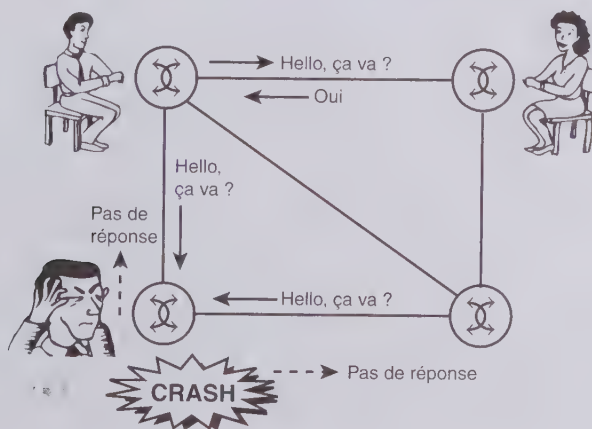
Carte de la topologie du réseau
(source : Learning Tree).



Les routeurs échangent périodiquement des informations concernant l'état de leurs liaisons (voir Figure 10.15). L'état de la liaison peut être soit *up* (en activité), soit *down* (hors service), selon qu'une liaison est opérationnelle ou non. Les messages d'état de liaison sont propagés à travers le réseau de façon que chaque routeur ait connaissance des mêmes informations d'état que les autres. Contraire-

Figure 10.15

Détermination de l'état
de la liaison (source :
Learning Tree).



ment à ce qui se produit dans le cas des protocoles à vecteur de distance, les informations de liaison sont reçues en provenance de la source et non suite à un calcul intermédiaire effectué par un autre routeur. Autrement dit, les messages d'état de liaison sont reçus sans modification. L'éventualité d'un mauvais traitement du message par un routeur est donc réduite. En outre, il est plus facile de mettre en évidence les problèmes qui génèrent des entrées incorrectes dans les tables de routage.

Les messages d'état de liaison contiennent des informations concernant les connexions directes vers un routeur donné. Les connexions indirectes n'y figurent pas. La taille d'un tel message dépend du nombre de connexions directes d'un routeur et pas de la taille du réseau. Donc, les protocoles de routage à état de liaison peuvent bien mieux s'adapter à des réseaux de grande taille que les protocoles à vecteur de distance.

Généralement, la plupart des protocoles à état de liaison ne diffusent pas leurs messages. Ils utilisent plutôt une adresse de multidiffusion de classe D, plus efficace. Si la couche matérielle de liaison de données ne prend pas en charge la diffusion ou la multidiffusion, comme c'est par exemple le cas des liaisons point à point, une copie du message est individuellement transmise aux routeurs concernés.

Dans certains protocoles à état de liaison, les messages d'état de liaison sont envoyés périodiquement. Dans d'autres, ces messages ne sont envoyés que lors de modifications de l'état de la liaison. L'état de la liaison est déterminé en interrogeant périodiquement (*polling*) les routeurs voisins. Dans certains cas, l'état de la liaison peut passer rapidement de *up* à *down*. La plupart des protocoles en tiennent compte et envoient plusieurs interrogations et mesurent le rapport entre messages d'état *up* et *down* pour une liaison donnée. A moins qu'une majorité des réponses n'indique que la liaison est *down*, on considère qu'elle est *up*.

Il est important de bien comprendre que les informations échangées ici par les routeurs ne sont pas les routes elles-mêmes, mais l'état des liaisons. Connaître l'état des liaisons permet au routeur de savoir s'il existe un chemin de communication direct entre deux routeurs. Les tables de routage sont calculées sur la base des messages d'état de liaison. Les routeurs tiennent à jour une carte du réseau, qu'ils mettent à jour en fonction des messages d'état de liaison qu'ils reçoivent. En cas de modification dans l'état d'une liaison, la carte est modifiée en conséquence et la table de routage est calculée à l'aide de l'algorithme SPF. Cet algorithme calcule le plus court chemin vers les destinations du réseau en partant du routeur sur lequel s'effectue le calcul. L'algorithme est exécuté sur chaque routeur, et les calculs des uns ne dépendent pas de ceux des autres. En conséquence, l'algorithme SPF permet une convergence rapide.

• Caractéristiques d'OSPF

OSPF est une implémentation répandue de protocole d'état de liaison sur les réseaux IP. Voici quelques-unes des caractéristiques d'OSPF :

- OSPF route les paquets IP en se basant sur l'adresse IP de destination et le TOS IP figurant dans l'en-tête du paquet IP. Il est possible d'avoir plusieurs routes vers une même destination, chacune étant adaptée à un type de service différent. OSPF calcule des routes distinctes pour chaque type de service.
- Lorsque deux routes sont équivalentes en termes de coût, OSPF peut répartir aussi le trafic entre les deux.
- Pour OSPF, le coût d'une route est exprimé avec une métrique unique sans dimension (c'est-à-dire sans unité).
- OSPF permet de regrouper des ensembles de réseau en *zones*. La topologie d'une *zone* (*area* en anglais) est masquée pour le reste du SA. Cette dissimulation d'informations permet de réduire le trafic de routage. Le routage avec zones permet de limiter à la zone la propagation de mauvaises informations.
- OSPF permet de configurer les sous-réseaux IP de façon souple. Les routes distribuées par OSPF incluent les masques de sous-réseau. Il est possible d'utiliser des VLSM. Avec les VLSM, un paquet est routé en utilisant la correspondance la plus longue ou la plus spécifique. Les routes spécifiques à un routeur sont spécifiées en ne faisant figurer que des 1 dans le masque de sous-réseau.
- OSPF utilise un mécanisme de mot de passe simple pour l'authentification. Ce mécanisme est cependant peu sûr, car les mots de passe sont envoyés en clair (sans chiffrement).
- Les routes indiquées par d'autres protocoles inter-SA peuvent être passées de manière transparente dans le SA. Ces données externes sont conservées à l'écart des données d'état de liaison propres à OSPF. Les routes externes sont marquées avec la route ainsi indiquée. Cela permet aux routeurs situés sur la frontière du SA de passer des informations à travers le SA.
- OSPF permet à l'administrateur de réseau de créer des topologies de réseau virtuel. Ainsi, on peut établir une liaison virtuelle entre deux routeurs et ce même s'ils ne sont pas voisins.
- Pour OSPF, chaque réseau multiaccès qui contient au moins deux routeurs a un routeur désigné. Le routeur désigné envoie des messages d'état de liaison pour toutes les liaisons du réseau, à destination des routeurs connectés à ce réseau. Comme seul le routeur désigné envoie ces messages, le trafic de routage est réduit, ainsi que la taille de la base de données topologique.

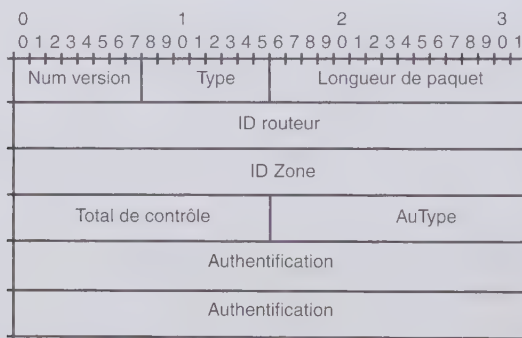


• Format de message OSPF

Tous les paquets OSPF ont en commun un en-tête de 24 octets. Cet en-tête de taille fixe contient les informations nécessaires pour déterminer la façon dont il convient de traiter le paquet. La Figure 10.16 montre cet en-tête OSPF commun.

Figure 10.16

En-tête OSPF fixé.



Le champ Numéro de version spécifie le numéro du protocole OSPF. Le numéro de version actuel pour OSPF est 2.

Le champ Type décrit le format des paquets. Le Tableau 10.2 dresse la liste des différentes valeurs possibles.

Tableau 10.2 : Valeurs pour le champ Type de l'en-tête OSPF

Type	Description
1	Hello
2	Description de la base de données
3	Requête état de liaison
4	Mise à jour état de liaison
5	Acquittement état de liaison

Le champ Longueur de paquet indique la longueur du paquet de protocole, en octets. Cette longueur prend en compte l'en-tête fixe OSPF.

Le champ ID router est une identification sur 32 bits de la source du paquet.

Le champ Area ID zone est un nombre de 32 bits identifiant la zone à laquelle appartient le paquet OSPF. Tous les paquets OSPF sont associés à une zone spécifique. En général, les paquets OSPF ne sont transmis que sur un pas. Les paquets

qui transitent par une liaison virtuelle sont étiquetés avec l'ID de zone dorsale 0.0.0.0.

Le champ Total de contrôle est le total de contrôle IP standard de l'ensemble du contenu du paquet OSPF, champ d'authentification de 64 bits exclus. Le total est calculé comme le complément à un de 16 bits du complément à un de la somme de tous les mots de 16 bits du paquet, en excluant le champ d'authentification. Si la longueur du paquet ne correspond pas à un nombre entier de mots de 16 bits, on complète le paquet avec un octet à 0 avant d'effectuer le calcul du total.

Le champ AuType indique le mécanisme d'authentification utilisé pour le paquet. Le champ Authentification est long de 64 bits et utilisable par le mécanisme d'authentification.

Les paquets OSPF sont transportés directement par IP, aucune couche de transport n'intervient. L'ID de protocole 89 est réservé pour OSPF. Le multiplexage des paquets OSPF au niveau de la couche IP plutôt qu'à celui de la couche de transport permet de rendre plus efficace l'implémentation.

Paquet Hello OSPF

Les paquets Hello OSPF sont des paquets OSPF de type 1 envoyés pour déterminer l'état des liaisons directement connectées. Ces paquets sont des multidiffusions/diffusions lorsque le support matériel sous-jacent le permet. La Figure 10.17 montre le format du message Hello.

Le champ Masque de réseau contient le masque de sous-réseau de l'interface par laquelle est envoyé le paquet Hello.

Intervalle Hello est le nombre de secondes qui s'écoulent entre les envois des paquets Hello de ce routeur.

Le champ Options indique les options prises en charge par le routeur.

Le champ Priorité routeur indique la priorité du routeur. Ce champ est utilisé pour choisir le Routeur désigné. Si ce champ vaut 0, le routeur ne peut devenir Routeur désigné.

Le champ RouterDeadInterval indique le nombre de secondes qui s'écoulent avant que l'on considère qu'un routeur silencieux est hors service.

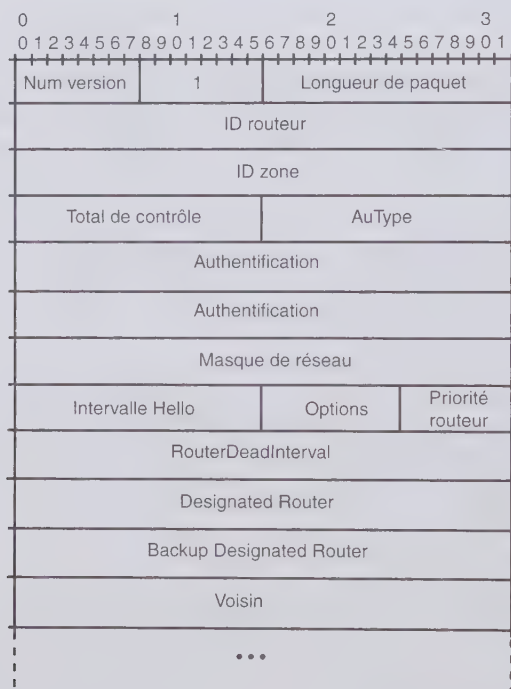
Le champ Designated Router ("routeur désigné") indique quel est le routeur désigné pour ce réseau. Le champ contient l'adresse IP de l'interface du routeur. Lorsque l'adresse vaut 0.0.0.0, il n'y a pas de routeur désigné.

• 10 •



Figure 10.17

Format du message Hello
OSPF.



Le champ Backup Designated Router ("routeur désigné de secours") indique l'adresse IP du routeur désigné de secours pour ce réseau. Il vaut 0.0.0.0 si ce routeur n'existe pas.

Le champ Voisin indique les ID de routeur de chaque routeur dont on a récemment vu passer des paquets Hello valides (depuis RouterDeadInterval) sur le réseau.

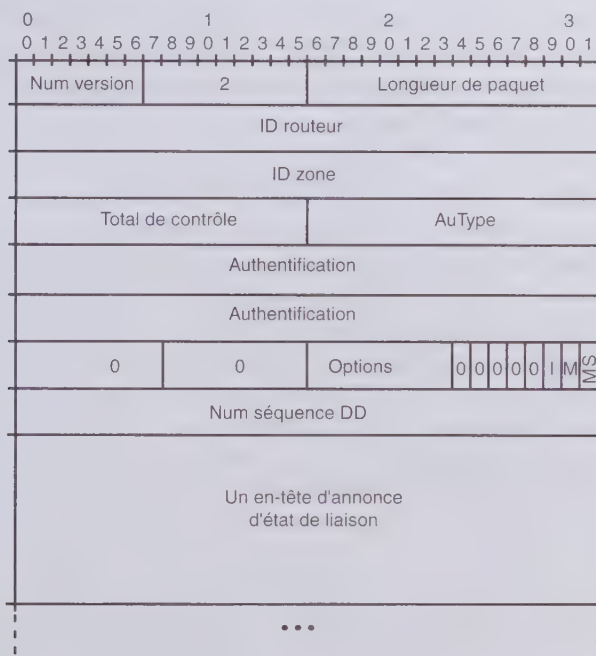
Paquet de description de base de données OSPF

Les paquets de description de base de données OSPF ont le type 2. Ces paquets sont échangés lorsqu'un routeur s'initialise à partir d'un serveur qui lui sert de maître pour déterminer le contenu de la base de données topologique. Du fait de la grande taille de cette base de données, de nombreux paquets peuvent devoir être envoyés pour la décrire. La Figure 10.18 montre le paquet de description OSPF.

La Figure 10.18 montre que la partie principale du message est une liste d'en-têtes.

Figure 10.18

Paquet de description de base de données OSPF.



Le champ Options décrit les fonctionnalités prises en charge par le routeur.

Le champ I-bit est le champ Init. Lorsque ce champ vaut 1, ce paquet est le premier de la séquence des paquets de Description de base de données. Le champ M-bit est le champ More ("plus"). Lorsque ce champ vaut 1, cela indique que d'autres paquets de description vont suivre. A eux deux, les champs I-bit et M-bit indiquent comment sont divisés les multiples messages formant une base de données topologique de grande taille.

Le champ MS-bit est le bit maître/esclave. Lorsque ce champ vaut 1, cela indique que le routeur est le maître lors du processus d'échange de base de données. Lorsqu'il vaut 0, le routeur est esclave.

Le numéro de séquence DD permet d'énumérer correctement la collection de paquets de description de manière que le routeur esclave puisse déterminer si un message est manquant. La valeur initiale de ce champ doit être non ambiguë. Ce numéro est incrémenté jusqu'à ce que tous les messages de description aient été envoyés.



Le reste du paquet est constitué d'une liste d'annonces d'états de liaison.

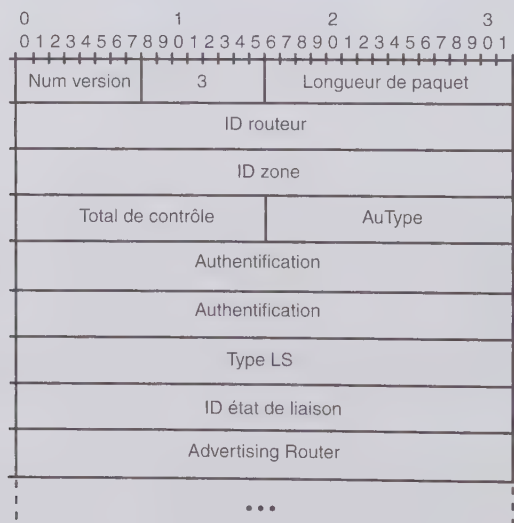
Chaque annonce d'état de liaison est décrite par son en-tête d'annonce d'état de liaison. L'annonce d'état de liaison est documentée dans la section "Format de l'annonce d'état de liaison".

Paquet de requête d'état de liaison OSPF

Les paquets de requête d'état de liaison ont le type OSPF 3. Après s'être initialisé à l'aide des messages de description de base de données OSPF reçus en provenance d'un routeur maître, un routeur peut s'apercevoir que certaines parties de sa base de données sont périmées. Le paquet de requête d'état de liaison permet de demander des fragments plus à jour dans la base de données d'un routeur voisin. Un routeur peut envoyer plusieurs paquets de requête d'état de liaison. L'état de liaison demandé est défini sans ambiguïté par le numéro de séquence LS (*Link State*), le total de contrôle LS et l'âge LS. La Figure 10.19 montre le format du paquet de requête d'état de liaison.

Figure 10.19

Paquet de requête d'état de liaison OSPF.



Chaque état de liaison demandé est spécifié par son type LS, l'ID d'état de liaison et le routeur qui fait l'annonce (*advertising router*). Ces valeurs permettent d'identifier sans ambiguïté l'annonce. Les paquets de requête sont des requêtes pour l'état le plus récent.

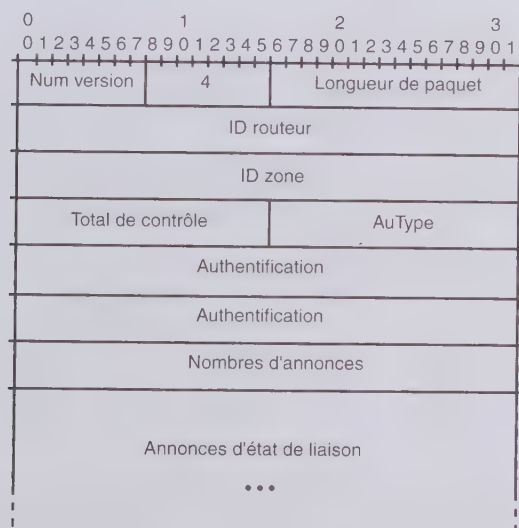
Paquet de mise à jour d'état de liaison OSPF

Les paquets de mise à jour d'état de liaison OSPF ont le type 4. Ces paquets permettent de diffuser dans le réseau des annonces d'état de liaison. Il est possible d'inclure plusieurs annonces d'état de liaison dans un seul paquet. Chaque paquet de mise à jour contient une collection d'annonces d'état, un pas plus loin que leur origine.

Ces paquets (voir Figure 10.20) sont multidiffusés/diffusés sur des réseaux prenant en charge ces modes d'envoi au niveau de la couche de liaison de données. La réception des paquets est confirmée à l'aide de paquets d'acquiescement d'état de liaison. Si les mises à jour ne sont pas reçues par un routeur donné, elles sont retransmises en utilisant des paquets de mise à jour envoyés en unicast.

Figure 10.20

Paquet de mise à jour d'état de liaison OSPF.



Le champ Nombre d'annonces indique le nombre d'annonces incluses dans la mise à jour.

Le reste du paquet est composé d'une liste d'annonces d'états de liaison. Chaque annonce est décrite par un en-tête. L'annonce d'état de liaison est documentée dans la section "Format de l'annonce d'état de liaison".

* 20 *



LE MACMILLAN

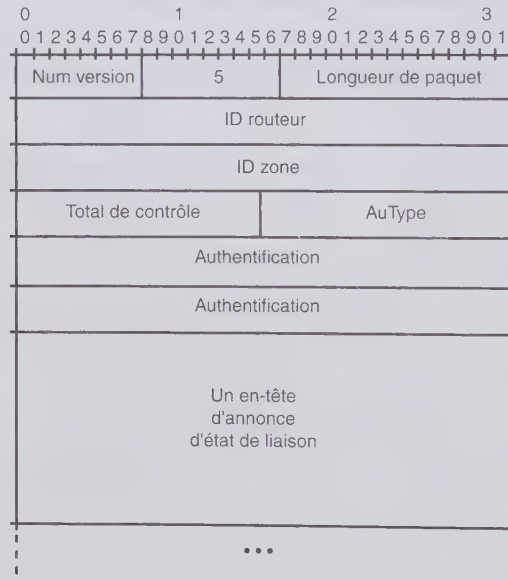
Paquet d'acquittement d'état de liaison OSPF

Les paquets d'acquittement d'état de liaison OSPF sont de type 5. Ces paquets permettent d'accuser réception d'un paquet de mise à jour d'état de liaison. Son but premier est de rendre fiable l'envoi massif d'annonces d'état de liaison. Des annonces multiples peuvent être acquittées à l'aide d'un seul paquet d'acquittement.

La Figure 10.21 montre le format du paquet d'acquittement d'état de liaison.

Figure 10.21

Paquet d'acquittement d'état de liaison OSPF.



Le corps du paquet d'acquittement est simplement une liste d'en-têtes d'état de liaison acquittés. Chaque annonce est décrite par un en-tête. Le format de l'annonce elle-même est développé dans la section suivante.

Format de l'annonce d'état de liaison

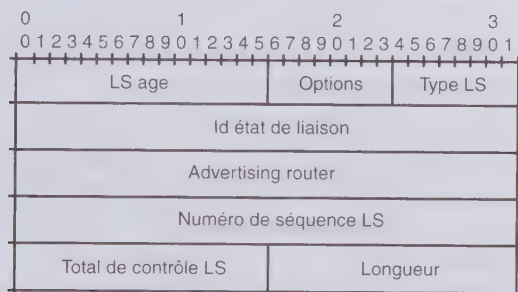
Il existe cinq types d'annonces d'état de liaison. Chacun commence par un en-tête d'état de liaison standard de 20 octets. Chaque annonce décrit une partie du domaine de routage OSPF. Chaque routeur envoie une annonce d'état de routeur. Lorsque le routeur est promu routeur désigné, il envoie une annonce d'état de réseau. Toutes les annonces sont alors envoyées en masse dans le domaine de routage OSPF. Cette collection d'annonces est appelée la *base de données topologique des états de liaison*. Cette base de données est utilisée par chaque routeur

pour construire l'arbre de plus court chemin, le routeur figurant à la racine de cet arbre. Les valeurs des sommets de cet arbre donne la table de routage elle-même.

La Figure 10.22 décrit l'en-tête de 20 octets commun.

Figure 10.22

En-tête d'annonce d'état de liaison.



Le champ LS age représente le temps écoulé en secondes depuis le dernier envoi d'annonce d'état de liaison.

Le champ Options indique des fonctionnalités facultatives prises en charge par la portion décrite du domaine de routage.

Le Type LS type indique le type de l'annonce d'état de liaison. Chaque type a un format d'annonce qui lui est propre (voir Tableau 10.3).

Tableau 10.3 : Types d'états de liaison OSPF

Type LS	Description
1	Liaisons de routeur
2	Liaisons de réseau
3	Liaison résumé (réseau IP)
4	Liaison résumé (ASBR)
5	Liaison SA externe

Le champ ID état de liaison identifie la partie de l'environnement de l'interréseau décrite par l'annonce.

Le champ Advertising router est l'ID du routeur qui est à l'origine de l'annonce.



Le champ Numéro de séquence LS détecte les annonces d'état de liaison trop anciennes ou en double. Les instances successives d'une annonce d'état de liaison reçoivent des numéros de séquence successifs.

Le champ Total de contrôle LS est un total de contrôle du contenu de l'annonce, prenant en compte l'en-tête d'annonce, mais pas le champ LS age.

Le champ Longueur indique la taille en octets de l'annonce, prenant en compte l'en-tête d'annonce faisant 20 octets.

L'en-tête d'annonce contient assez d'informations pour identifier de manière non ambiguë l'annonce (Type LS, ID état de liaison et Advertising Router). Si plusieurs instances d'une annonce existent, on retient la plus récente en regardant les champs LS age, Numéro de séquence LS et Total de contrôle LS qui figurent aussi dans l'en-tête.

La RFC 1583 décrit en détail le protocole OSPF.

• Modèle de table de routeur

Jusqu'ici, nous n'avons décrit que des parties de la table de routage. En général, les entrées des tables de routage contiennent des informations supplémentaires. Jusqu'à l'apparition des standards SNMP décrivant la base d'information de gestion (MIB) d'un routeur, il n'existait pas de représentation standard. Chaque fabricant de routeur concevait sa propre représentation de la table de routage. A partir de la publication de la RFC 1213, une définition formelle pour la table de routage a pris forme. Voici la définition du MIB de table de routage, utilisant la notation ASN.1 :

```
IpRouteEntry ::=
    SEQUENCE {
        ipRouteDest
            IPAddress,
        ipRouteIfIndex
            INTEGER,
        ipRouteMetric1
            INTEGER,
        ipRouteMetric2
            INTEGER,
        ipRouteMetric3
            INTEGER,
        ipRouteMetric4
            INTEGER,
        ipRouteNextHop
            IPAddress,
```

```

        ipRouteType
            INTEGER,
        ipRouteProto
            INTEGER,
        ipRouteAge
            INTEGER,
        ipRouteMask
            IPAddress,
        ipRouteMetric5
            INTEGER,
        ipRouteInfo
            OBJECT IDENTIFIER
    }

```

La Figure 10.23 montre les informations ci-avant traduites sous forme d'une table de routage. Au cours du fonctionnement du routeur, des entrées de route sont ajoutées dans cette table. Les colonnes décrivent le type d'informations indiquées pour chaque route.

Figure 10.23

Informations contenues dans une table de routage.

ipRoute Dest	ipRoute If Index	ipRoute Metric 1 • • • ipRoute Metric 5	ipRoute Next Hop	ipRoute Type	ipRoute Proto	ipRoute Age	ipRoute Mask	ipRoute Info



Voici un résumé de la signification de ces colonnes :

- **ipRouteDest.** L'adresse IP de destination de cette route. Une entrée avec la valeur 0.0.0.0 est considérée comme une route par défaut. Plusieurs routes pour une même destination peuvent apparaître dans la table, mais le mode d'accès à ces entrées multiples dépend du mécanisme d'accès défini par le protocole de gestion de réseau utilisé.
- **ipRouteIfIndex.** La valeur d'indice qui identifie de manière non ambiguë l'interface locale par le biais de laquelle on pourra franchir le prochain pas de la route.
- **ipRouteMetric1.** La métrique de routage principale pour cette route. La sémantique de cette métrique est déterminée par le protocole de routage spécifié par ipRouteProto. Si cette métrique n'est pas utilisée, la valeur doit être - 1.
- **ipRouteMetric2, ipRouteMetric3, ipRouteMetric4, ipRouteMetric5.** Une métrique de routage de secours pour la route. La sémantique de cette métrique est déterminée par le protocole de routage spécifié par ipRouteProto. Si cette métrique n'est pas utilisée, la valeur doit être -1.
- **ipRouteNextHop.** L'adresse IP du prochain pas de la route.
- **ipRouteType.** Le type de route. Peut être autre(1), invalide(2), directe(3) ou indirecte(4). Les valeurs directe/indirecte correspondent aux définitions de routage direct et indirect dans l'architecture IP. En affectant la valeur invalide(2), on rend invalide l'entrée correspondante dans l'objet ipRouteTable. Autrement dit, la destination est alors effectivement dissociée de la route.
- **IpRouteProto.** Le mécanisme de routage par le biais duquel la route a été connue. Les valeurs possibles sont : autre(1) pour des entrées manuelles ; local(2) pour des entrées via un réseau ; netmgmt(3) pour des entrées faites par un protocole de routage ; icmp(4) pour des entrées faites par un message ICMP tel que Redirect ; egp(5) ; ggp (6) : hello(7) ; rip(8) ; is-is(9) ; es-is(10) ; ciscogrp(11) ; bbnSpflgp(12) ; ospf(13) ; bgp(14).
- **ipRouteAge.** Le nombre de secondes écoulées depuis la dernière fois que la route a été mise à jour.
- **ipRouteMask.** Indique le masque avec lequel effectuer un ET logique sur l'adresse de destination avant de comparer le résultat au champ ipRouteDest. Si la valeur de ce dernier champ est 0.0.0.0 (route par défaut), la valeur du masque est elle aussi 0.0.0.0.
- **ipRouteInfo.** Une référence aux définitions MIB spécifiques au protocole de routage particulier responsable pour cette route.

Récapitulatif

Ce chapitre a traité en détail des protocoles RIP et OSPF qui sont utilisés avec les systèmes autonomes. Ce sont tous deux des protocoles IGP. RIP est l'exemple le plus représentatif d'un protocole à vecteur de distance, tandis qu'OSPF est un protocole à état de liaison. Le protocole RIP a subi de nombreuses modifications pour améliorer certains de ses défauts les plus manifestes. Le nouveau protocole RIP est appelé "RIP version 2" (RIPv2). Ce chapitre n'a toutefois traité que RIP version 1, qui reste encore très répandu.

Le protocole recommandé pour une utilisation entre différents systèmes autonomes est BGP4. RIPv2 et BGP4 seront traités dans un ouvrage à venir.

Testez vos connaissances

1. Quelle est la raison pour laquelle les organisations construisent leurs réseaux sous forme de systèmes autonomes ?
2. Indiquez les différences entre les protocoles à vecteur de distance et les protocoles à état de liaison.
3. Donnez des exemples de protocoles à vecteur de distance.
4. Donnez des exemples de protocoles à état de liaison.
5. Quelle est la métrique utilisée avec RIP ? Quelles sont ses valeurs maximale et minimale ?
6. Est-il possible de construire avec RIP des réseaux comprenant plus de 15 routeurs ?
7. Est-il possible de construire avec RIP des réseaux de plus de 15 routeurs entre deux nœuds d'un réseau ?
8. Un routeur RIP contient les entrées suivantes dans sa table de routage :

Destination	Distance	Routeur de prochain pas
134.33.0.0	1	(directement connecté)
145.108.0.0	1	(directement connecté)



La mise à jour RIP suivante est reçue en provenance du routeur 145.108.0.0.

Destination	Distance
144.19.0.0	2
200.1.1.0	3
200.2.5.0	2
0.0.0.0	

Quel est le nouveau contenu de la table de routage ? Quelle est la route par défaut dans cette table ?

9. Un routeur RIP contient les entrées suivantes dans sa table de routage :

Destination	Distance	Routeur de prochain pas
134.33.0.0	1	(directement connecté)
145.108.0.0	1	(directement connecté)
0.0.0.0	2	134.33.12.1
34.0.0.0	4	145.108.1.9
141.12.0.0	3	145.108.1.9

La mise à jour RIP suivante est reçue en provenance du routeur 145.108.0.0.

Destination	Distance
0.0.0.0	1
199.245.180.0	3
34.0.0.0	2
0.0.0.0	3
141.12.0.0	2

Quel est le nouveau contenu de la table de routage ? Quelle est la route par défaut dans cette table ?

10. On trouve une chaîne de 16 réseaux connectés par 16 routeurs. Les routeurs utilisent le protocole RIP. Si des intervalles de diffusion de routeurs adjacents sont décalés de 15 secondes, combien de temps faudra-t-il pour qu'une mise à jour de routage passe d'une extrémité du réseau à l'autre ?

11. Dressez la liste des problèmes posés par RIP. Dans quelles circonstances un administrateur de réseau a-t-il intérêt à choisir RIP ?
12. Décrivez les différences entre les routeurs RIP actifs et passifs. Donnez des exemples.
13. Quels sont les problèmes liés au compte de pas de RIP ?
14. Dans quelles circonstances RIP converge-t-il lentement ? Illustrez votre réponse par un exemple de réseau.
15. Qu'est-ce que le problème du "comptage à l'infini" dans RIP ?
16. Décrivez les techniques employées pour éliminer la convergence lente de RIP ?
17. Quel problème peut poser l'utilisation de la technique *poison reverse with triggered update* avec RIP ?
18. Quels sont les messages RIP courants dans un réseau ?
19. A quelle occasion un message RIP de requête est envoyé ?
20. Quel type de message RIP est utilisé pour diffuser les routes ?
21. Pourquoi RIP utilise-t-il un protocole de transport ? Quels sont les numéros de port utilisés par RIP pour ce protocole ?
22. Quel est le nombre maximal de routes pouvant être transmises dans un seul message RIP ? Justifiez votre réponse.
23. RIP peut-il être utilisé avec un réseau utilisant des VLSM ? Justifiez votre réponse.
24. Décodez le paquet suivant :


```

0: FF FF FF FF FF FF 00 00 C0 A2 0F 8E 08 00 45 00 | .....E.
10: 00 34 00 2C 00 00 40 11 0F 9E 90 13 4A C9 90 13 | .4.,..@.....J...
20: FF FF 02 08 02 08 00 20 FE 96 02 01 00 00 00 02 | .....
30: 00 00 90 13 00 00 00 00 00 00 00 00 00 00 00 00 | .....
40: 00 00 01                                     | ...
      
```
25. Dressez la liste de quelques-unes des caractéristiques de OSPF.
26. OSPF utilise-t-il un protocole de transport ? Tracez un diagramme montrant les couches de transport utilisées avec OSPF.
27. A quoi sert le paquet OSPF Hello ?
28. A quoi sert le paquet OSPF Description de la base de données ?
29. A quoi sert le paquet OSPF Requête d'état de liaison ?



30. A quoi sert le paquet OSPF Mise à jour d'état de liaison ?
31. Comment est assurée la remise fiable des paquets de mise à jour d'état de liaison ?
32. Quel type de paquet OSPF est utilisé pour envoyer en masse les messages d'état de liaison ?

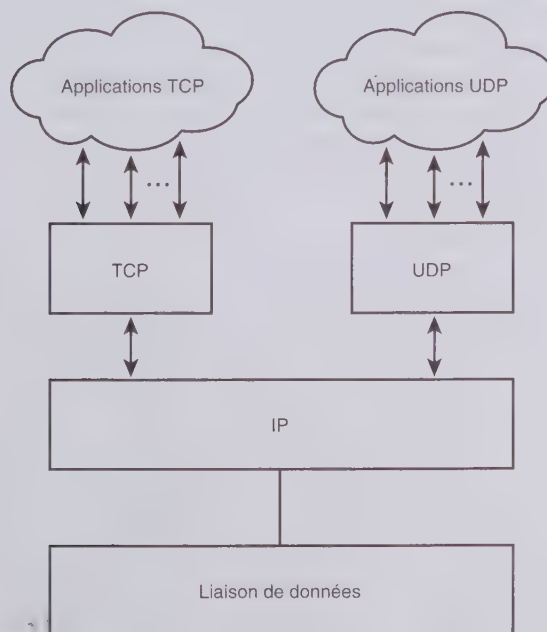
II

Protocoles de transport

Les protocoles de transport TCP/IP correspondent à la couche de Transport du modèle OSI. La pile de protocoles TCP/IP définit deux protocoles de transport standard : TCP et UDP. TCP (*Transmission Control Protocol* ou "Protocole de contrôle de transmission") implémente un protocole de flux de données fiable tandis que UDP (*User Datagram Protocol* ou "Protocole de datagrammes utilisateur") implémente un protocole de flux de données non fiable.

TCP, comme UDP, s'exécute au-dessus d'IP et se fonde sur les services fournis par ce dernier (voir Figure 11.1). IP fournit un service de datagrammes sans connexion entre deux machines. En utilisant TCP et UDP, vous pouvez remettre des données non pas seulement à un ordinateur distant, mais aussi à un processus d'application s'exécutant sur un ordinateur distant. Ces processus d'application sont identifiés par des numéros de port. TCP peut garantir que les données sont remises de façon fiable à destination, en fournissant un service orienté connexion.

Figure 11.1
TCP et UDP.



En revanche, UDP est sans connexion et ne peut garantir la remise des données. UDP s'avère toutefois utile pour de nombreuses applications, en particulier celles pour lesquelles des données doivent être envoyées à une application donnée s'exécutant sur une machine, ou celles dont les données doivent être diffusées ou multidiffusées.

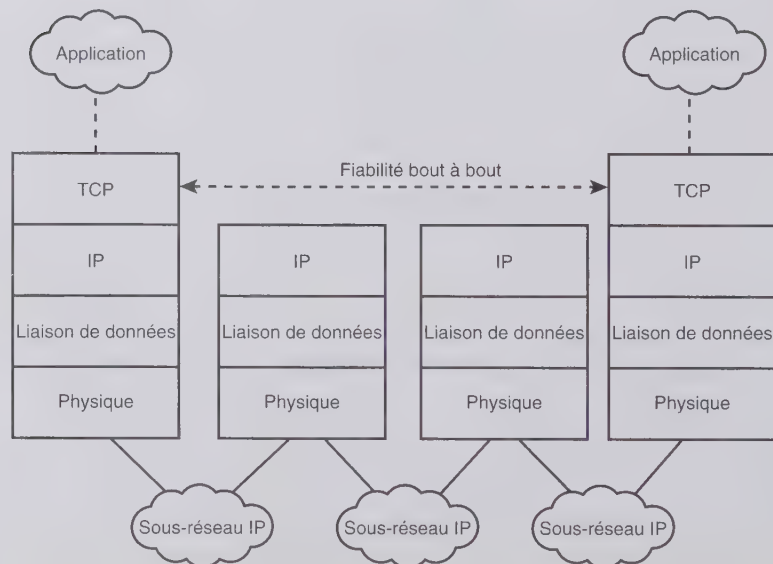
Transmission Control Protocol

Le protocole TCP (*Transmission Control Protocol*) fournit un service primordial dans la pile TCP/IP, consistant en une méthode généraliste de remise fiable des données. Plutôt que d'inventer leur propre protocole de transport, les applications utilisent généralement TCP pour remettre des données de manière fiable. En effet, le protocole TCP est en pleine maturité et a pu bénéficier de nombreuses améliorations garantissant performances et fiabilité.

TCP garantit la fiabilité point à point entre un processus d'application s'exécutant sur un système et un autre processus d'application s'exécutant sur une machine distante (voir Figure 11.2). Cette fiabilité est assurée grâce à l'ajout de différents services au-dessus d'IP. IP est sans connexion et ne garantit en rien la fiabilité de la remise des paquets. TCP part du principe qu'IP est non fiable par essence ; il ajoute ses propres services pour garantir la fiabilité. Comme TCP part de l'hypothèse la plus pessimiste sur les services fournis par le réseau, ce protocole peut

Figure 11.2

Fiabilité point à point de TCP.



s'exécuter sur une vaste gamme de matériels. Tout ce dont TCP a besoin c'est d'un service de datagramme simple, non fiable, dans la couche inférieure. Certains réseaux, tels que X.25, fournissent des services orientés connexion sophistiqués au niveau de la couche de Réseau. Dans ce cas, les datagrammes IP sont transmis sur un circuit virtuel X.25 et les paquets de protocole TCP sont encapsulés dans les datagrammes IP.

TCP est le protocole de transport le plus utilisé pour les connexions fiables, full-duplex sur circuits virtuels. Il s'exécute la plupart du temps sur IPv4 ou IPv6, mais certaines expérimentations ont été menées pour l'exécuter au-dessus d'autres protocoles de couche réseau.

Bien qu'IP soit implémenté sur les hôtes et les routeurs, TCP ne l'est généralement que sur les hôtes pour fournir une remise point à point fiable. De nos jours, la plupart des routeurs ne fonctionnent pas de manière dédiée. Ils fournissent d'autres services. Cela les rend plus aisés à configurer et à administrer. Ainsi, de nombreux routeurs commerciaux peuvent aussi implémenter TCP ou UDP pour offrir la connexion à distance et des fonctions de gestion de réseau. S'il arrive donc que TCP et UDP soient implémentés sur les routeurs, ils ne servent pas pour les services de routage proprement dits (ni pour les messages de ces derniers).

Le protocole TCP est décrit dans la RFC 792. Le protocole UDP est décrit dans la RFC 768. La RFC 1122 (STD3) contient des ajouts très importants. Les protocoles TCP et UDP ont le statut recommandé. Mais, dans les faits, la totalité des périphériques TCP/IP – sauf les routeurs purs – implémentera ces deux protocoles. Dans la plupart des implémentations, la partie UDP est incorporée au module TCP. Elle n'est donc pas distincte de l'implémentation TCP.

• Fonctionnalités TCP

Les fonctionnalités de TCP sont les suivantes :

- transfert de données de base ;
- fiabilité ;
- contrôle du flux ;
- multiplexage ;
- connexions ;
- préséance et sécurité.



Transfert de données de base

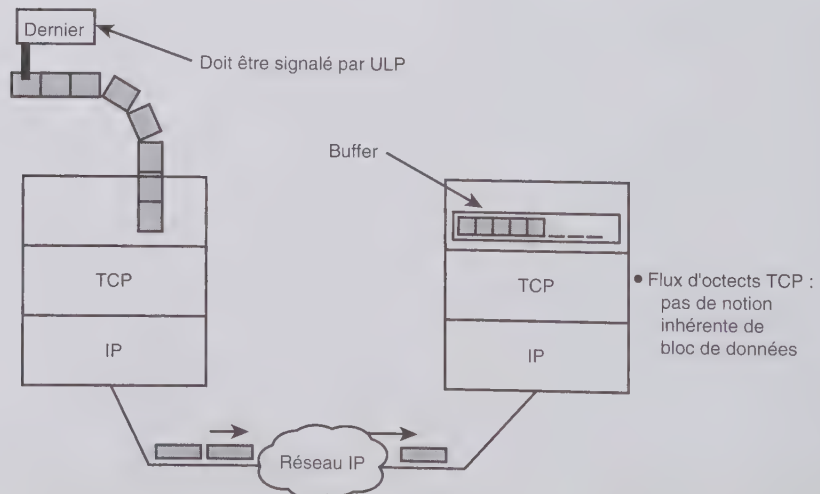
Le *transfert de données de base* est la capacité de TCP à transférer un flux continu d'octets dans chaque direction. Les octets sont envoyés entre des processus d'application s'exécutant sur des systèmes distants utilisant TCP. Les processus d'application regroupent ensuite dans un *segment de message* un ensemble d'octets devant être envoyés/reçus. Les segments de message peuvent être de taille arbitraire. En fin de compte, les messages doivent être envoyés sous forme de datagrammes IP, dont la taille est limitée par la taille de MTU d'une interface de réseau. Néanmoins, au niveau de TCP, il n'existe pas de réelle restriction sur la taille des messages, car les détails de la conversion des segments de messages en datagrammes IP ne concernent que la couche IP. Afin d'être plus efficace dans la gestion des messages, les connexions TCP négocient généralement une taille maximum de segment.

Les messages envoyés par TCP sont considérés comme un flux d'octets (voir Figure 11.3). TCP assure le suivi de chaque octet reçu/envoyé. La notion de bloc de données est inconnue de TCP, à la différence d'autres protocoles de transport qui assurent généralement le suivi du numéro de TPDU (*Transport Protocol Data Unit* ou "Unité de données du protocole de transport") plutôt que de celui de l'octet. TCP peut être utilisé pour fournir plusieurs connexions de circuits virtuels entre deux hôtes TCP.

Les processus d'application utilisant TCP envoient des données dans la taille la plus appropriée. Ainsi, par exemple, une application peut envoyer des données ne faisant qu'un octet ou plusieurs Ko. TCP numérote chaque octet qu'il envoie. Les octets sont remis aux processus d'application à l'autre extrémité dans leur ordre d'envoi. On appelle cela le *séquençement* des octets.

Figure 11.3

Flux d'octets TCP
(source : Learning Tree).



Une application peut envoyer des données à TCP par groupes de quelques octets. TCP met ces données dans un tampon, puis envoie ces octets sous la forme d'un message unique ou de plusieurs segments de message plus petits. Tout ce que TCP garantit est que les données arriveront au destinataire dans leur ordre d'envoi. Ainsi, si une application envoie 1 024 octets de données sur une période de 10 secondes, ces données peuvent être envoyées sur le réseau sous la forme d'un unique paquet TCP de 1024 octets ou de quatre paquets TCP de 256, ou de n'importe quelle autre combinaison.

Comme TCP envoie les données sous forme de flux d'octets, on ne trouve pas de véritable marqueur de fin de message dans le flux. Pour s'assurer que toutes les données soumises au module TCP ont bien été transmises, une fonction de *push* est nécessaire. Le *push* fait que TCP envoie rapidement toute donnée reçue jusqu'ici en provenance d'une application.

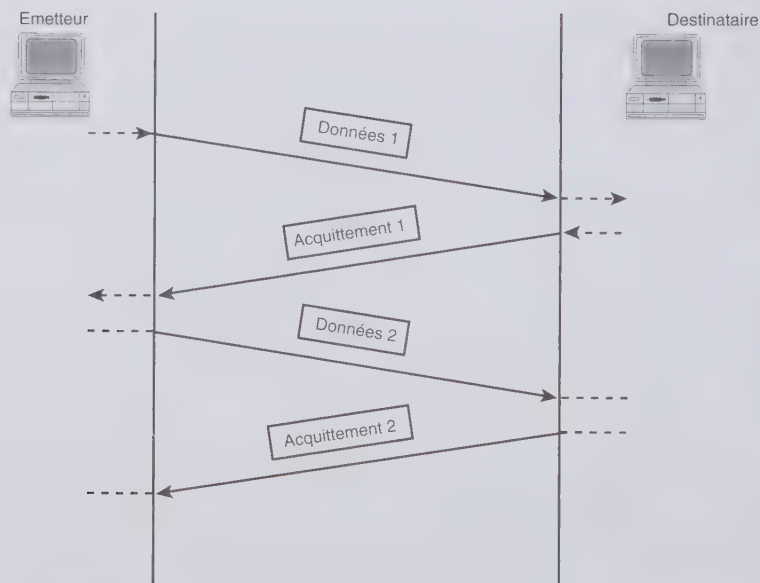
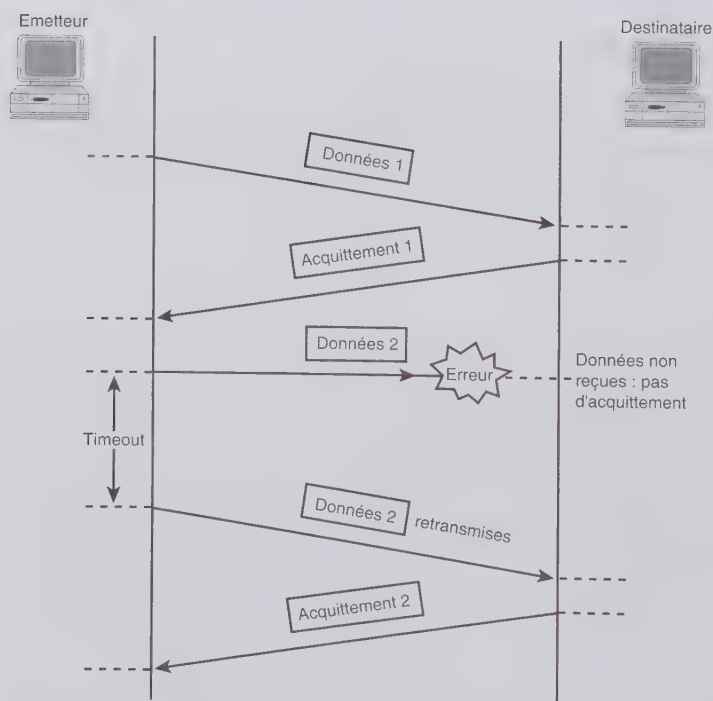
Les données envoyées par TCP sont traitées comme un flux d'octets non structuré. TCP ne dispose pas de mécanisme permettant de superposer une structure propre à l'application sur ces données. Ainsi, vous ne pouvez pas demander à TCP de traiter des données comme un jeu d'enregistrements dans une base de données afin d'envoyer les enregistrements un à un. Cette mise en structure doit être gérée par les processus d'application communiquant à l'aide de TCP.

Fiabilité

Une des fonctionnalités les plus importantes de TCP est la remise fiable de point à point. Pour cela, TCP doit pouvoir restaurer les données endommagées, perdues, dupliquées ou remises en mauvais état. Il utilise le mécanisme PAR (*Positive Acknowledgment Retransmission* ou "Retransmission d'acquiescement positif") présenté sur la Figure 11.4. Sur cette figure, la réception des données et acquiescements se déroule sans problème. La Figure 11.5 montre le mécanisme après une perte de données.

TCP implémente ce mécanisme en affectant un numéro de séquence à chaque octet transmis et en requérant un acquiescement positif (ACK) du module TCP récepteur. Si cet acquiescement ne parvient pas avant expiration d'une certaine temporisation (timeout), TCP effectuera une retransmission. Au niveau du module TCP récepteur les numéros de séquence servent à ordonner les segments et à éliminer les doublons. Pour détecter les données erronées, on se sert du champ checksum (total de contrôle) de l'en-tête TCP. Les segments de données dont le total de contrôle ne correspond pas seront écartés. A moins d'une interruption physique de la liaison, TCP peut se remettre de la plupart des erreurs des systèmes de communication que l'on rencontre sur l'Internet.



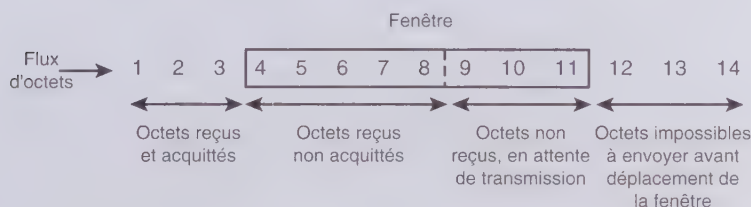
Figure 11.4*Le mécanisme PAR.***Figure 11.5***Le mécanisme PAR après une perte de données.*

Contrôle du flux

Les machines qui émettent et reçoivent les segments de données TCP ne le font pas toutes au même rythme en raison de la diversité des unités centrales et de la bande passante. Il peut donc arriver que l'émetteur envoie ses données beaucoup plus rapidement que le récepteur ne peut les gérer. C'est pourquoi TCP implémente un mécanisme de contrôle du flux de données. A cet effet, il se sert d'une fenêtre glissante (voir Figure 11.6). Le flux de données est numéroté octet par octet, chaque numéro s'appelle un *numéro de séquence*.

Figure 11.6

La fenêtre glissante de contrôle de flux de TCP.



Le module TCP du récepteur renvoie à l'émetteur un acquittement qui fixe un intervalle pour les numéros de séquence à partir du dernier segment correctement reçu. Cet intervalle est appelé *fenêtre* : il indique à l'émetteur le nombre d'octets qu'il peut transmettre sans avoir besoin d'autre autorisation.

La Figure 11.6 présente une fenêtre correspondant à des numéros de séquence allant de 4 à 11. Les octets 1 à 3 ont déjà été envoyés et acquittés. Les octets 4 à 11 sont déjà envoyés ou sont sur le point de l'être, mais ils n'ont pas encore été acquittés. L'émetteur TCP n'a pas obligation d'envoyer autant d'octets que le permettrait la fenêtre, laquelle ne fait qu'indiquer un maximum. Sur la Figure 11.6, les octets 4 à 8 ont déjà été envoyés, mais pas les octets 9 à 11. Mais ils le seront bientôt. En effet, tant que le segment entre dans les limites imposées par la fenêtre l'émetteur peut l'envoyer sans délai. Voici le résumé des caractéristiques du mécanisme de contrôle du flux que l'on trouve dans TCP :

- Les octets qui précèdent la limite inférieure de la fenêtre ont été envoyés et acquittés.
- Les octets se trouvant dans l'intervalle défini par la fenêtre peuvent être envoyés sans délai. S'ils ont déjà été envoyés, leur acquittement n'est pas encore parvenu.
- Les octets qui dépassent la limite supérieure de la fenêtre n'ont pas encore été envoyés et ne pourront l'être que lorsqu'il passeront dans l'intervalle de la fenêtre.



La limite inférieure de la fenêtre est l'octet non acquitté dont le numéro est le plus faible. La fenêtre va progresser, on dit de gauche à droite, dès qu'elle recevra des acquittements pour les données expédiées. Le paquet TCP qui contient l'acquittement contient aussi des informations concernant la taille de la fenêtre que l'émetteur peut se permettre.

La taille de la fenêtre dépend de la quantité d'espace libre encore disponible dans le tampon de données du récepteur. Lorsque le récepteur est trop sollicité il renvoie dans les acquittements une taille de fenêtre réduite. Dans les cas extrêmes cette taille peut se réduire à 1 : c'est ce que l'on appelle le syndrome de la fenêtre stupide ou SWS (voir plus bas, "Le syndrome SWS").

Lorsque le module TCP du récepteur renvoie une taille de fenêtre égale à zéro cela indique à l'expéditeur que ses tampons sont pleins et qu'il ne faut plus lui envoyer aucune donnée. TCP prévoit des mécanismes pour réduire ou augmenter la taille de la fenêtre lorsque cela est nécessaire (état de congestion).

Grâce à la fenêtre glissante le trafic est moins inconstant et l'attente des acquittements réduite.

Multiplexage

TCP peut servir simultanément à plusieurs processus de la même machine : c'est le *multiplexage*. Ces processus communiquent par la même interface réseau et partagent donc la même adresse IP que l'interface réseau. L'adresse IP ne permet donc pas d'identifier un processus.

TCP associe en fait un numéro de port à chaque application qui utilise ses services. Une telle association permet de distinguer les connexions TCP entre les différents processus d'applications se trouvant sur des machines distantes. A chaque connexion correspondra une paire de numéros de port. La Figure 11.7 illustre le multiplexage des connexions TCP.

Chaque machine attache les numéros de port à ses applications indépendamment des autres machines. Sur de nombreux systèmes, il existe un processus appelé *logger* ou *super daemon* chargé de la surveillance des numéros de port qui sont identifiés ou usuels pour les autres systèmes.

Connexions

Avant de pouvoir envoyer des données à l'aide de TCP, les processus doivent d'abord établir une connexion. Une connexion s'établit entre le numéro de port de l'émetteur et celui du récepteur. On appelle cela les extrémités (*end points*) de la connexion. Une extrémité est définie par une paire de valeurs : l'adresse IP et le numéro de port (voir Figure 11.8) :

(adresse IP, numéro de port)

Figure 11.7
Multiplexage.

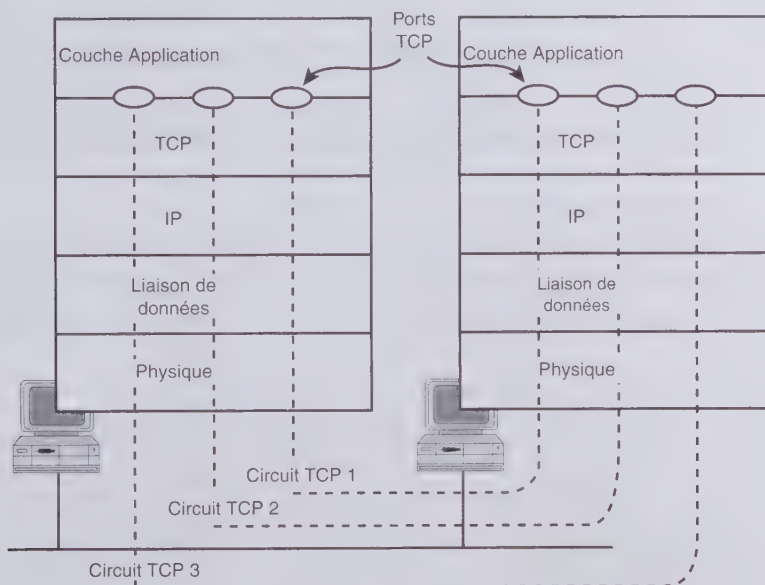
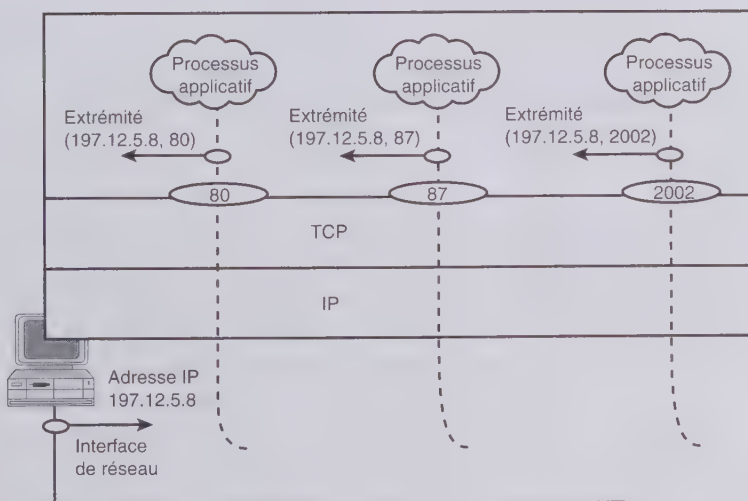


Figure 11.8
Une extrémité TCP.

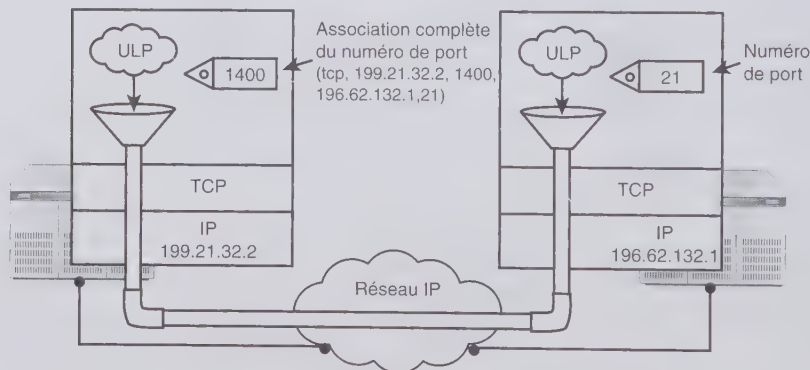


L'adresse IP correspond à l'adresse de l'interface réseau par laquelle l'application TCP/IP communique. Le numéro de port est un numéro que TCP associe à l'application. Pour identifier une extrémité, il faut préciser non seulement l'adresse IP, mais encore le numéro de port. Le numéro de port seul ne suffirait pas, car chaque module TCP choisit indépendamment les numéros de ports et l'on pourrait avoir un numéro identique pour deux applications différentes situées sur des machines distinctes.

On établit une connexion entre deux extrémités (voir Figure 11.9). Pour identifier une connexion, il faut les données correspondant aux deux extrémités :

(adresse IP 1, numéro de port 1, adresse IP 2, numéro de port 2)

Figure 11.9
Connexion TCP
entre extrémités.



TCP peut prendre en charge simultanément plusieurs connexions multiplexées sur la même interface réseau.

Une connexion est définie par deux extrémités. De plus, TCP peut transmettre les données sur cette connexion dans un sens ou dans l'autre. C'est une connexion en *full duplex*.

Une autre notion est utile pour comprendre les protocoles de transport, c'est le concept d'association. Une demi-association est la donnée d'une extrémité et d'un nom de protocole :

(Protocole de transport, adresse IP, numéro de port.)

Par exemple, voici les demi-associations qui correspondent à la Figure 11.9 :

(tcp, 199.21.32.2, 1400)

(tcp, 196.62.132.1, 21)

Une association complète est la réunion de deux demi-associations. On l'exprime par un couple (une paire dans laquelle l'ordre des termes importe) :

(TransportProtocol, IP address1, port number1, IP address2, port number2.)

Le protocole de transport apparaît une seule fois puisqu'il s'agit du même pour les deux demi-associations. Voici l'association complète qui correspond à la Figure 11.9 :

(tcp, 199.21.32.2, 1400, 196.62.132.1, 21)

• L'environnement de l'hôte TCP

L'implémentation de TCP consiste en un module de protocole qui interagit avec le système d'exploitation de l'ordinateur. Dans de nombreux systèmes on accède au module TCP comme on accède au système de fichiers du système d'exploitation. L'interface Sockets BSD implémentée sur les systèmes Unix ou l'interface Winsock implémentée par les systèmes propriétaires de Microsoft reposent sur ce principe.

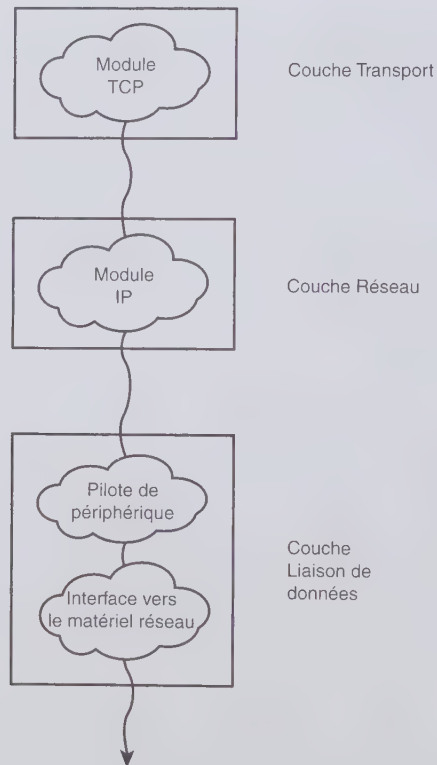
Le module laisse gérer ses structures de données et ses services par les fonctions du système d'exploitation. L'interface réseau elle-même est généralement contrôlée par un module de pilote de périphérique. TCP n'interagit pas directement avec ce module, mais appelle le module IP, qui appelle à son tour le module de pilote de périphérique (voir Figure 11.10).

Il est possible de déléguer le traitement des modules de pile du protocole TCP/IP à un processeur frontal (*front-end*). Ce processeur agit comme un périphérique dédié aux protocoles de communication et exécute les modules du protocole (voir Figure 11.11). Dans cette perspective, un mécanisme est nécessaire pour que l'hôte communique avec le processeur frontal. Il s'agit généralement d'une interface de système de fichiers.

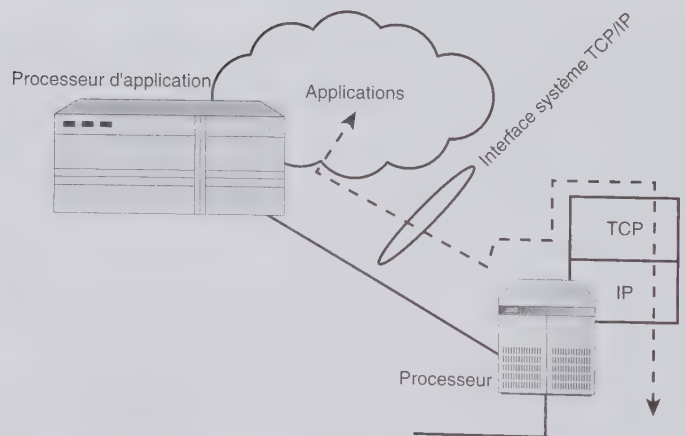


Figure 11.10

Interactions entre le module TCP et le réseau.

**Figure 11.11**

Déléguer le traitement de TCP.



De façon générale l'interface entre applications et module TCP utilise les appels systèmes suivants :

- OPEN qui sert à ouvrir une connexion.
- CLOSE qui sert à fermer une connexion.
- SEND qui sert à envoyer les données lorsque la connexion est ouverte.
- RECEIVE qui permet de recevoir les données lorsque la connexion est ouverte.
- STATUS qui fournit des renseignements sur la connexion.

On implémente ces appels système comme des appels normaux provenant de programmes utilisateur. Ainsi l'appel d'ouverture ou de fermeture de connexion sont l'équivalent d'appels d'ouverture ou de fermeture de fichier. Les appels SEND et RECEIVE équivalent à des appels d'écriture ou de lecture sur un fichier.

Les appels système TCP peuvent interagir avec d'autres modules TCP n'importe où sur le système d'interréseau. Pour s'exécuter, les appels système TCP ont besoin qu'on leur passe certains paramètres. Parmi ces paramètres, citons les adresses IP, le type de service, la préséance, la sécurité et le numéro de port de l'application.

• Ouverture et fermeture d'une connexion TCP

Lors de l'appel OPEN, on spécifie une connexion sur le port local. Les paramètres correspondants sont les extrémités de la connexion. Un appel OPEN passif indique que le processus entend accepter les requêtes de connexion qui lui arrivent plutôt que d'être à l'origine d'une connexion. L'appel OPEN passif peut éventuellement spécifier une extrémité particulière afin de n'accepter que cette connexion.

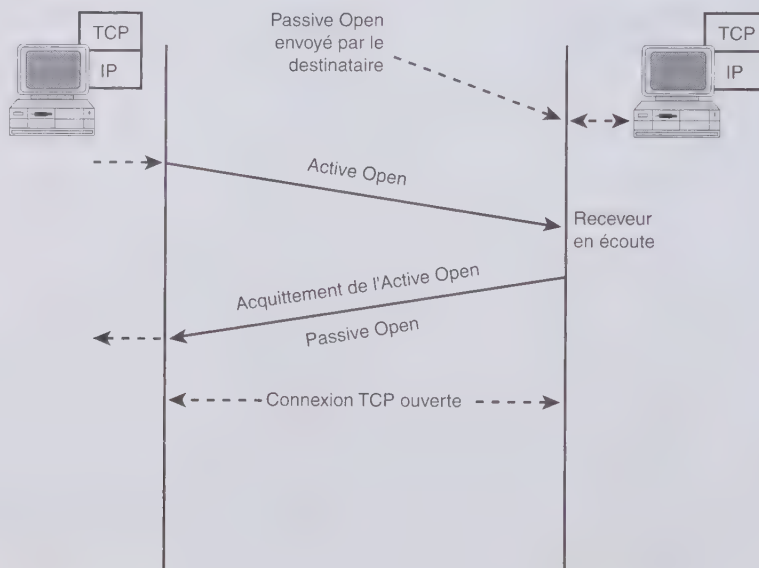
TCP informe les processus ayant émis un appel OPEN passif lorsque d'autres processus émettent des appels actifs et qu'une connexion s'est établie (voir Figure 11.12).

On peut aussi utiliser deux appels d'ouverture active simultanés pour connecter des processus (voir Figure 11.13). C'est particulièrement intéressant pour un système distribué, où chaque composant peut agir de manière asynchrone.

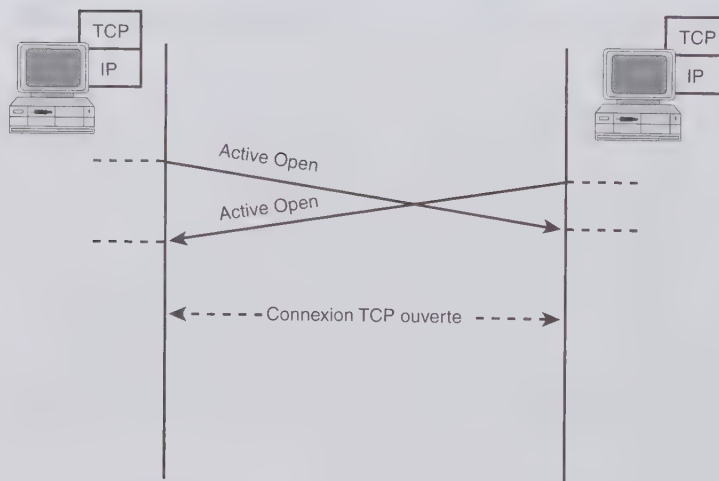


Figure 11.12

Etablir une connexion TCP avec un appel OPEN actif et un appel OPEN passif.

**Figure 11.13**

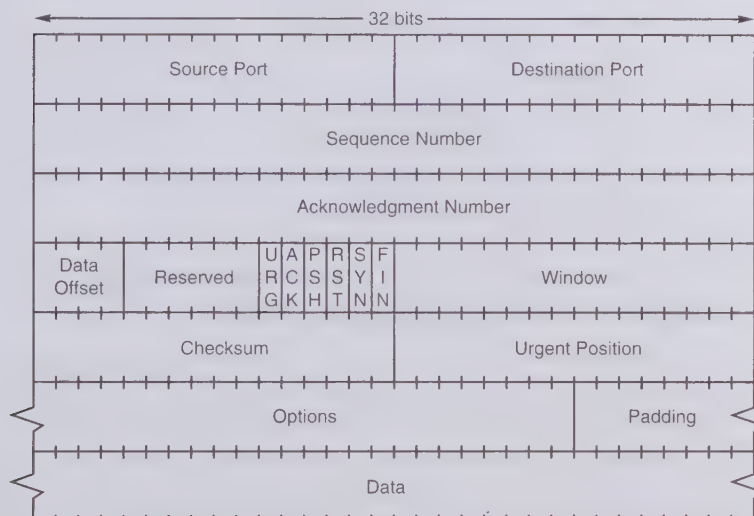
Etablir une connexion TCP avec deux appels OPEN actifs.



• Le format des messages TCP

La Figure 11.14 présente la structure des paquets TCP. Comme la plupart des protocoles TCP/IP l'en-tête TCP utilise un format en mots de 32 bits.

Figure 11.14
La structure des paquets TCP.



Dans les sections qui suivent, nous étudierons les différents champs de l'en-tête TCP.

Numéros du port d'origine et du port de destination (champs **Source Port Number** et **Destination Port Number**)

Ces champs servent à identifier les extrémités du circuit virtuel TCP. Ils contiennent respectivement le numéro de port du processus d'origine et le numéro de port du processus de destination. Certains numéros sont usuels (*well-known*), d'autres sont enregistrés et d'autres sont affectés dynamiquement.

Périodiquement, une RFC décrit les numéros de port usuels. Le Tableau 2.5 (Chapitre 2) présentait l'organisation des numéros de port TCP. Les numéros compris entre 0 et 1023 sont réservés à l'IANA : ce sont les numéros de port *usuels*. Les numéros supérieurs à 1023 peuvent être enregistrés, mais ils ne sont pas sous contrôle de l'IANA : ce sont les numéros de port enregistrés. Parmi les



numéros supérieurs à 1023, tous ne sont cependant pas des numéros enregistrés : les autres sont affectés librement en fonction des besoins, ce sont les numéros de port transitoires. Les numéros de port transitoires sont alloués à l'intérieur d'une implémentation TCP. L'algorithme que l'on utilise pour allouer ces numéros varie selon les implémentations.

Les modules TCP peuvent associer librement un numéro de port et un processus, sauf que les numéros usuels sont réservés à certaines applications. Dans de nombreux systèmes d'exploitation, les applications sont "propriétaires" des ports qui leur sont attribués aléatoirement (et dont les numéros sont supérieurs à 1023). Dans certaines implémentations, une fonction de "hashing" prend en compte le nom du processus dans l'attribution des numéros de port : les bits de poids fort du numéro sont associés au nom du processus.

Numéros de séquence et d'acquittement (champs Sequence Number et Acknowledgment Number)

La fiabilité de la transmission par TCP repose sur les numéros de séquence et d'acquittement. Le principe consiste à attribuer un numéro à chaque octet de données. Le numéro du premier octet de données du segment de message figure dans l'en-tête TCP du segment : c'est le *numéro de séquence de segment*.

Lorsque le récepteur envoie un segment de message, il envoie en même temps un numéro d'acquittement qui doit correspondre au numéro de séquence de l'octet de données suivant attendu. Les transmissions TCP sont full duplex : un même module TCP peut être tantôt émetteur et tantôt récepteur.

Lorsque TCP transmet un segment il en fait une copie dans la file d'attente de transmission et déclenche un temporisateur. La copie sera supprimée si l'acquittement correspondant à ce segment parvient à l'émetteur avant expiration du temporisateur TCP. Sinon, une retransmission a lieu à partir de cette copie.

Le champ de 32 bits Sequence Number (voir Figure 11.14) contient le numéro de séquence du premier octet de données du message courant. Si le flag SYN vaut 1, ce champ définit le numéro de séquence initial ou ISN correspondant à la session en cours. Le premier décalage de données vaudra donc $ISN + 1$. On utilise une valeur de 32 bits pour éviter d'employer des numéros de séquence déjà attribués à des données qui sont en transit sur le réseau.

Le champ Acknowledgment Number (numéro d'acquittement) contient le numéro de séquence du prochain octet auquel s'attend le récepteur. Les acquittements TCP sont cumulés : un même acquittement peut servir à acquitter plusieurs segments de message TCP. Après établissement d'une connexion il y a toujours envoi d'un numéro d'acquittement.

L'acquittement signifie que le module TCP a reçu les données, mais il ne garantit pas que les données sont parvenues à l'application. Néanmoins, le module TCP a

la charge de faire parvenir les données et l'application finira par les obtenir, à moins qu'une erreur dans le module TCP ou dans l'interface de l'application ne l'empêche.

Décalage de données (champ Data Offset)

Ce champ indique le nombre de mots de 32 bits qui se trouvent dans l'en-tête TCP (voir Figure 11.14). Cette information est nécessaire parce que le champ Options a une longueur variable. Lorsque le champ Options est vide, la valeur de Data Offset est cinq (soit 20 octets). La longueur de l'en-tête est toujours un multiple de 32 bits.

Sur la Figure 11.14 une seule option est définie : la taille de segment maximum ou MSS. Cette option joue un rôle au début de la connexion TCP. La présence de cette option porte la valeur du champ Data Offset à six (24 octets).

Les champs Reserved qui suivent Data Offset doivent recevoir la valeur 0.

Les flags (champ Flags)

Voici la signification des flags que l'on trouve dans l'en-tête TCP (voir Figure 11.14) :

- L'activation du flag ACK indique que le champ Acknowledgment Number est valide.
- L'activation du flag SYN indique l'ouverture d'une connexion de circuit virtuel.
- L'activation du flag FIN indique la fermeture de la connexion.
- L'activation du flag RST indique la réinitialisation du circuit virtuel pour cause d'erreurs irrécupérables. A la réception du flag RST le récepteur doit immédiatement terminer la connexion. Les deux extrémités ayant mis fin à leur connexion, les ressources associées à cette connexion sont libérées. Le transfert des données est interrompu, ce qui risque d'amener la perte des données en transit. Le flag RST n'est pas la voie de fermeture normale d'une liaison TCP. Normalement on ferme une liaison TCP à l'aide du flag FIN. Le flag RST s'avère souvent nécessaire à la suite d'une défaillance d'hôte ou d'un retard dans la duplication des paquets SYN.
- L'activation du flag PSH indique la remise immédiate des données au processus de la couche supérieure.
- L'activation du flag URG indique l'envoi des données hors bande sans attendre que le récepteur ait traité les octets déjà envoyés dans le flux.



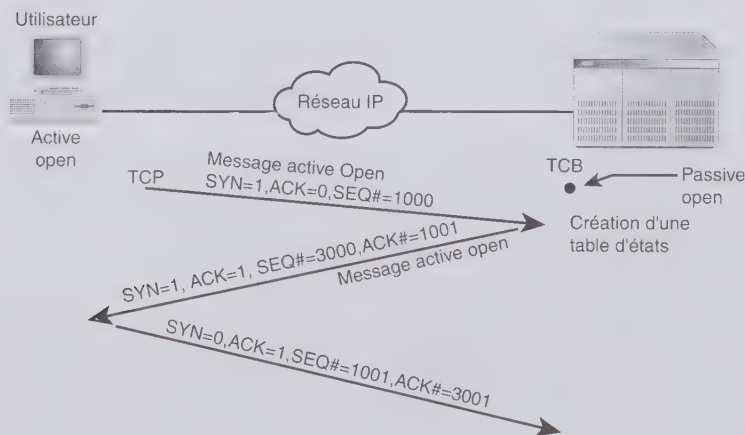
On ouvre une connexion TCP à l'aide d'une procédure de négociation en trois temps. Voici les flags SYN et ACK correspondant :

SYN = 1 et ACK = 0	Paquet d'ouverture de connexion
SYN = 1 et ACK = 1	Acquittement de l'ouverture de connexion
SYN = 0 et ACK = 1	Paquet de données ou paquet ACK

La procédure d'établissement de la connexion utilise le flag de synchronisation SYN et implique l'échange de trois messages, d'où le terme négociation en trois temps (voir Figure 11.15).

Figure 11.15

Utilisation du flag SYN
(source : Learning Tree).



TCB= Transmission Control Block

L'initialisation d'une connexion suppose la conjonction de l'arrivée d'un segment contenant le flag SYN avec l'attente d'une entrée TCB. Le segment est généré par l'appel OPEN actif alors que l'entrée TCB est créée par l'appel OPEN passif. L'activation du flag SYN permet la synchronisation des numéros de séquence dans les deux directions. La Figure 11.15 présente un segment TCP généré par ouverture active et possédant les flags suivants :

SYN = 1, ACK = 0

Le récepteur TCP, qui a envoyé un appel OPEN passif, reçoit l'appel OPEN actif, il l'acquitte avec les flags suivants :

SYN = 1, ACK = 1

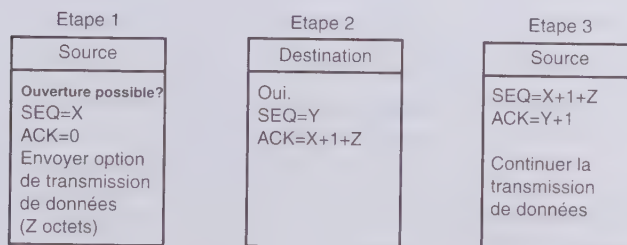
On appelle parfois paquet OPEN passif l'acquittement du paquet OPEN actif. Le flag ACK = 1 indique que le champ Acknowledgment Number contient des données valides.

La Figure 11.16 illustre le mécanisme de négociation en trois temps.

Figure 11.16

*La négociation en trois temps
(source : Learning Tree).*

•Three-way Handshake



Dans l'étape 1 de la Figure 11.16 le numéro de séquence pour l'ouverture active est X. On peut éventuellement envoyer aussi Z octets de données dans le paquet d'ouverture. Il pourrait s'agir de données d'identification pour le service requis (par exemple le nom d'utilisateur et le mot de passe). La plupart des applications TCP/IP n'envoient les données d'identification que lorsque l'établissement de la connexion est achevé. Néanmoins, en particulier lorsque TCP /IP fonctionne sur un réseau X.25, les données d'identification font parfois partie de la requête d'ouverture de connexion. Voici les numéros de séquence et d'acquittement de cette étape :

Sequence Number = X

Acknowledgment Number = 0

Dans l'étape 2 le récepteur répond par un numéro d'acquittement égal à $X + 1 + Z$, c'est-à-dire le numéro du prochain octet attendu. Il envoie aussi Y, le numéro de séquence initial utilisé par la destination. Voici les numéros de séquence et d'acquittement de cette étape :

Sequence Number = Y

Acknowledgment Number = $X + 1 + Z$

Dans l'étape 3 la source acquitte le numéro de séquence que lui a envoyé le récepteur.



Voici les numéros de séquence et d'acquittement de cette étape :

$$\text{Sequence Number} = X + 1 + Z$$

$$\text{Acknowledgment Number} = Y + 1$$

La nécessité de la dernière étape de la négociation en trois temps n'est pas toujours bien comprise. Elle informe le récepteur que son numéro de séquence initial a bien été reçu par l'émetteur qui ouvre la connexion TCP. Sans cet acquittement on ne pourrait pas être sûr que l'émetteur connaît le numéro de séquence initial du récepteur. Voici un résumé des trois étapes de la négociation en trois temps :

- **Étape 1.** L'émetteur envoie son numéro de séquence initial ou ISS (SEND).
- **Étape 2.** Le récepteur envoie un numéro d'acquittement obtenu en ajoutant 1 à l'ISS ou à la quantité de données que lui a envoyées l'émetteur. Le récepteur envoie son numéro de séquence initial ou IRS (RECEIVER).
- **Étape 3.** L'émetteur envoie un numéro d'acquittement pour l'ISN du récepteur.

Pour comprendre pourquoi TCP n'envoie pas un numéro de séquence déjà attribué à un segment présent le réseau, on doit comprendre ce qu'est la durée de vie maximum d'un segment ou MSL (*Maximum Segment Lifetime*). Il s'agit de l'intervalle de temps que doit respecter TCP avant de réattribuer des numéros de séquence lorsqu'il vient de démarrer ou lorsque la connexion vient de se rétablir après une défaillance. La spécification TCP fixe MSL à deux minutes. Si l'implémentation TCP est capable de retenir en mémoire les numéros de séquence en cours d'utilisation, MSL n'a plus lieu d'être et les numéros de séquence n'auront qu'à être supérieurs aux derniers qui ont été utilisés.

TCP numérote chaque octet, les numéros défilent rapidement. Si le débit est de 2 Mbit/s, on utilise 2^{32} octets en 4 heures et demie. La valeur de MSL n'excède jamais quelques dizaines de secondes. Pour 100 Mbit/s la valeur de MSL est 5,4 minutes (une valeur supérieure à la spécification de 2 minutes que l'on trouve dans la RFC 793).

Pour sélectionner l'ISN à 32 bits la RFC 793 suggère d'utiliser un générateur de numéro de séquence initiale. Ce générateur peut par exemple être relié à l'horloge à 32 bits dont le bit de poids faible est incrémenté toutes les quatre microsecondes. Les numéros ISN parcourent alors un cycle environ toutes les 4,55 heures. Il est vraiment très improbable qu'un segment reste dans le réseau aussi longtemps et les ISN auront toujours des valeurs distinctes.

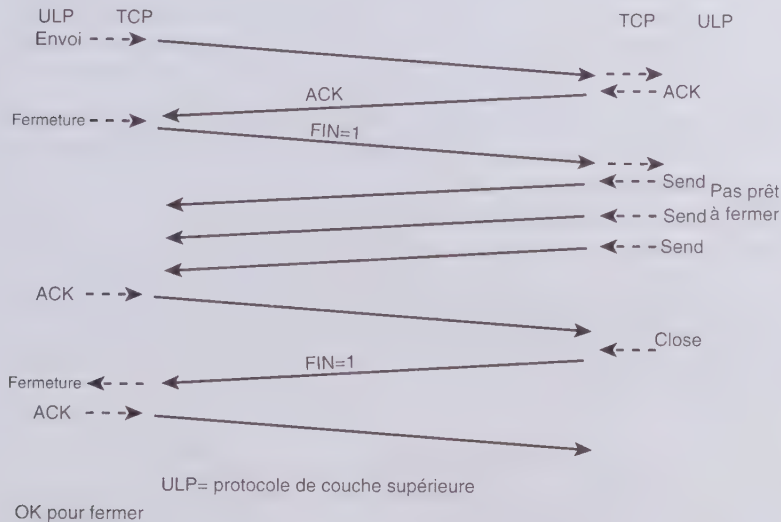
Après l'établissement de la connexion TCP le flag ACK contient la valeur 1 pour indiquer que le champ Acknowledgment Number (numéro d'acquittement) est valide. La transmission des données pourra se faire dans les deux sens, jusqu'à la fermeture de la connexion.

Au moment de la fermeture, une extrémité envoie le flag FIN. Mais la fermeture de la connexion TCP n'est réalisée que lorsque l'autre extrémité renvoie le flag

FIN et accepte la fermeture. Ce genre de fermeture est appelé *fermeture gracieuse*. La fermeture TCP permet d'éviter les pertes de données que pourrait occasionner une fermeture unilatérale (voir Figure 11.17).

Figure 11.17

Utilisation du flag FIN lors de la fermeture TCP (source : Learning Tree).



Le flag PSH servait à l'origine à ce que le module TCP traite tout de suite les données qu'il venait de recevoir. Lorsque l'application envoie une série d'appels SEND sans activer PSH, le module TCP a tendance à accumuler les données. De même, le module récepteur peut très bien recevoir une série de segments et les placer dans une file d'attente interne sans les passer à l'application. Le flag PSH n'est pas un marqueur d'enregistrement et il est indépendant des frontières de segment, même si certaines implémentations considèrent à tort PSH comme un marqueur d'enregistrement. Le transmetteur doit en effet écraser les bits PSH successifs lorsqu'il empaquette les données en un segment de la plus grande taille possible.

TCP implémente généralement des flags PSH dans les appels SEND. Mais s'il ne le fait pas, voici ce qu'il faut faire :

1. Eviter d'accumuler systématiquement les données dans le tampon ;
2. Activer le bit PSH du dernier segment (par exemple lorsqu'il n'y a plus de données dans la file d'attente).



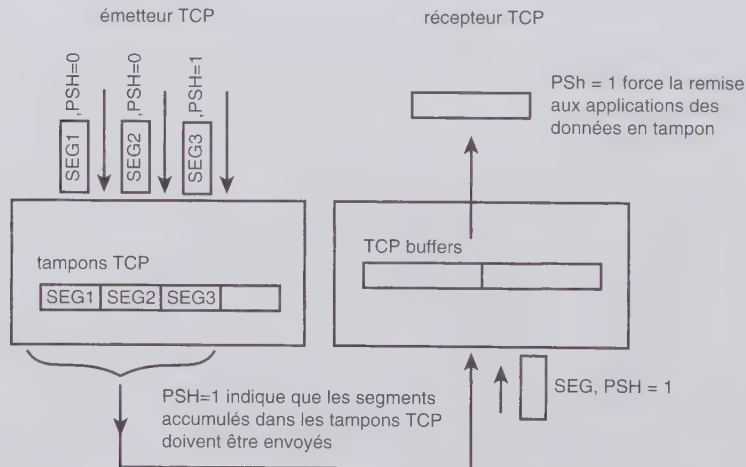
La RFC 793 soutient à tort qu'il est obligatoire de passer le flag PSH reçu à la couche application, alors que c'est désormais facultatif.

Le programme d'application est logiquement obligé d'activer le flag PSH dans l'appel SEND lorsqu'il entend forcer la remise des données pour éviter un verrouillage des communications. Mais TCP essaiera toujours d'envoyer des segments de taille maximale pour des raisons d'efficacité. Donc, côté émetteur, l'activation de PSH ne garantit pas la transmission immédiate des données.

Lorsque le flag PSH n'est pas implémenté dans les appels SEND de TCP (ou lorsque l'interface application/TCP repose sur un modèle de streaming pur), c'est à la couche application que revient en partie la charge de rassembler les fragments trop petits pour former des segments de taille acceptable. En général, un protocole d'application interactif doit activer le flag PSH, au moins dans le dernier appel SEND des commandes ou des réponses. Dans un protocole de transfert en masse tel que FTP, on doit activer le flag PSH dans le dernier segment du fichier ou lorsque c'est nécessaire éviter un verrouillage du tampon. La Figure 11.18 illustre l'utilisation du flag PSH.

Figure 11.18

Utilisation du flag PSH
(source : Learning Tree).



- Un émetteur TCP doit écraser les bits PSH successifs lorsqu'il transforme les données en paquets (RFC 1122)
- Les applications interactives doivent définir le bit PSH à la fin de chaque séquence commande/réponse
- Les protocoles de transfert en masse tels que FTP doivent définir le flag PSH du dernier segment d'un fichier.

Côté récepteur, le bit PSH provoque la remise des données à l'application, même si le tampon n'est pas tout à fait plein. Il peut être intéressant de ne pas activer systématiquement PSH pour ne pas appeler trop souvent et inutilement l'application, en particulier sur un hôte qui fonctionne en temps partagé avec de nombreuses applications.

TCP prévoit un mécanisme pour envoyer les données hors bande. On utilise le flag URG et le champ Urgent Pointer (pointeur urgent) pour passer devant la file d'attente lorsqu'un message est important pour le récepteur. Ce mécanisme permet par exemple à l'émetteur d'envoyer des signaux d'interruption au récepteur.

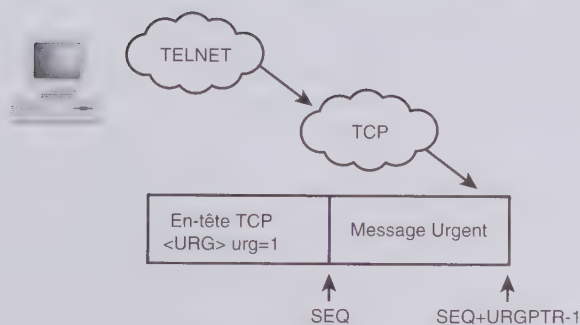
Sans ce mécanisme on ne pourrait pas par exemple interrompre le traitement des données utilisateur que l'on envoie parce que le signal d'interruption serait bloqué dans la file d'attente du récepteur et placé après les données en cours de traitement.

Si le flag URG est activé, le champ Urgent Pointer est valide (voir Figure 11.19). La RFC 1122 indique que le pointeur pointe vers le numéro de séquence du dernier octet des données urgentes (LAST et pas LAST + 1 comme le soutenait à tort la RFC 793).

Figure 11.19

Le flag URG de TCP
(source : Learning Tree).

- Utilisé pour envoyer des données hors bande :
- Sans attendre que le récepteur ne traite les octets déjà dans le flux
- N'a de sens que si URG=1



- Le pointeur Urgent indique le dernier octet du message urgent.



Une implémentation TCP doit prendre en charge les séquences de données urgentes quelle que soit leur taille. La couche TCP doit informer la couche application de façon asynchrone dès qu'elle reçoit un pointeur urgent sans qu'existent les données urgentes attendantes ou lorsque le pointeur urgent avance dans le flux de données. L'application doit au moins pouvoir savoir s'il reste des données urgentes à lire. En principe l'envoi de messages urgents est possible avec n'importe quelle application, mais on s'en sert normalement pour envoyer des commandes d'interruption aux programmes Telnet. La notification asynchrone ou hors bande permet à l'application de passer en mode urgent à la simple lecture de données provenant de la connexion TCP. De cette façon, on dispose éventuellement d'un moyen pour envoyer des commandes de contrôle à une application dont les tampons d'entrée normaux sont remplis de données non traitées.

La fenêtre (champ Window)

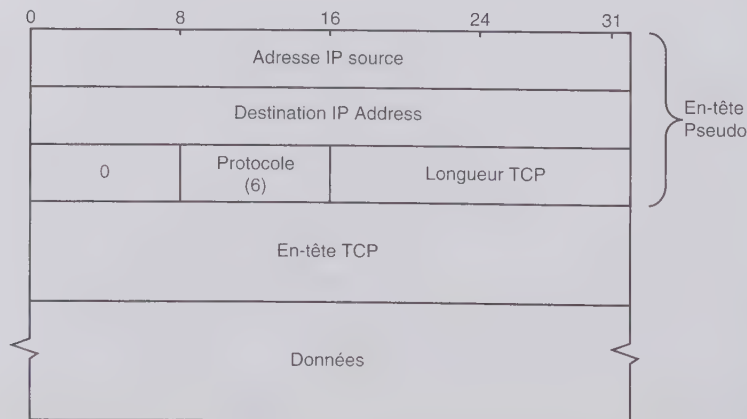
Ce champ implémente un contrôle de flux. C'est le récepteur qui fixe la taille de la "fenêtre" pour l'émetteur. Une fenêtre spécifie le nombre d'octets que le récepteur est en mesure d'accepter (à partir du numéro d'acquittement).

Le total de contrôle (champ checksum)

Ce champ contient le complément à 1 de la somme des compléments à 1 de tous les mots de 16 bits se trouvant dans le paquet TCP. Un pseudo en-tête de 96 bits (Figure 11.20) est réservé en amont de l'en-tête TCP pour le calcul du total de contrôle. Ce pseudo en-tête sert à déterminer si le paquet est parvenu à destination et évite les erreurs de routage des segments TCP. Il contient l'ID du protocole (6 pour TCP) et les adresses IP d'origine et de destination. Les numéros de port d'origine et de destination figurent dans l'en-tête TCP, ce qui achève de définir la connexion TCP (association complète).

Figure 11.20

Le rôle du pseudo en-tête dans le calcul du total de contrôle TCP.



Le champ Length (longueur) du pseudo en-tête contient la longueur de l'en-tête TCP plus la longueur des données en octets. Il s'agit d'une valeur calculée (elle n'est pas expressément transmise). Elle ne tient pas compte des 12 octets du pseudo en-tête.

Les options (champ Options)

Ce champ se trouve placé à la fin de l'en-tête TCP et occupe un nombre de bits multiple de huit. Ce champ participe au total de contrôle. Une option peut commencer à n'importe quel octet. Il existe deux formats d'option :

- Cas 1 : option à un seul octet indiquant le type de l'option ;
- Cas 2 : un octet pour le type, un octet pour la longueur et un ou plusieurs octets pour des données.

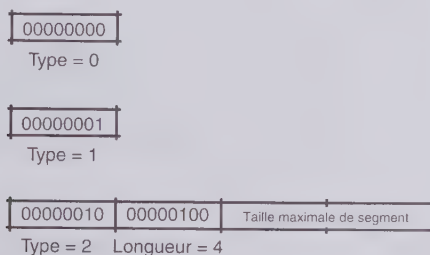
Pour le cas 2, la longueur qui figure dans le deuxième octet prend en compte non seulement les données, mais l'octet de type et l'octet de longueur. Il arrive que le nombre d'options ne permette pas d'atteindre la taille indiquée dans le champ Data Offset. Dans ce cas là, on remplit le reste de l'en-tête après l'octet de fin des options avec des zéros.

Voici quelles sont les valeurs d'octet correspondant à différents types d'options :

Type	Longueur	Signification
0	-	Fin des options (End-of-Options).
1	-	Pas d'opération (No-Operation).
2	4	Taille maximale de segment (MSS).

Le format des deux premiers types correspond au cas numéro 1. La Figure 11.21 présente les options TCP.

Figure 11.21
Les options TCP.



La fin des options ne coïncide pas toujours avec la fin de l'en-tête TCP telle que la définit le champ Data Offset. L'option End-of-Option doit apparaître à la fin de la liste des options, pas après chaque option de la liste. De plus, cette option n'est pas nécessaire lorsque la fin des options coïncide naturellement avec la fin de l'en-tête TCP.

Le type No-operation sert seulement à aligner le commencement de l'option suivante sur le début d'un mot. Mais les émetteurs TCP ne prennent pas toujours cette précaution et il faudra que les récepteurs puissent traiter les options, même lorsque leur disposition ne respecte pas la limite des mots.

En fait, l'option MSS est la seule qui transporte vraiment une information. La valeur de MSS se négocie à l'ouverture de la connexion TCP, lorsque l'on synchronise les numéros de séquence (flag SYN activé). En général, chaque extrémité informe l'autre de sa valeur de MSS. Lorsque l'option MSS n'est pas utilisée il n'y en principe a pas de limite à la taille des segments.

• Les ACK cumulés

La signification des numéros de séquence TCP est assez différente de ce qu'elle peut être dans d'autres protocoles. Pour la plupart des protocoles, le numéro de séquence reflète le numéro du paquet. Pour TCP, un numéro de séquence correspond à un numéro d'octet et les segments peuvent contenir plus ou moins d'octets. Si l'acquittement d'un segment ne parvient pas à temps à l'émetteur, celui-ci retransmet les données dans un segment. Mais le nouveau segment peut très bien contenir des données supplémentaires par rapport au premier segment envoyé (des données générées depuis). C'est pourquoi le numéro d'acquittement ne peut pas être un numéro de segment.

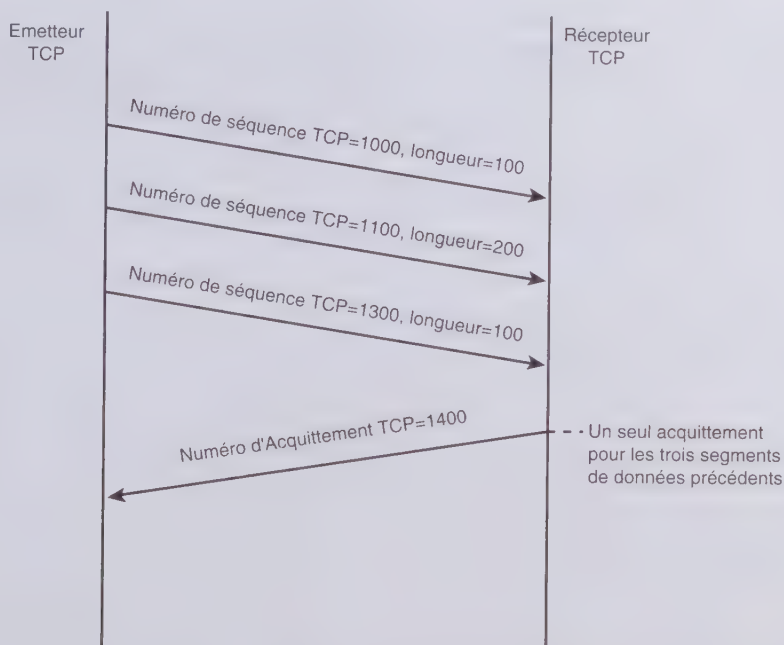
Un numéro d'acquittement TCP indique dans le flux de données la position des dernières données reçues par le module TCP distant. En fait, plus concrètement, il indique le numéro du prochain octet à envoyer. Le numéro d'acquittement est aussi la limite inférieure de la fenêtre TCP.

Dire que les acquittements TCP sont cumulés signifie que le numéro d'acquittement renvoyé prend en compte les données accumulées jusque-là. Un numéro d'acquittement peut ainsi correspondre à la réception de plusieurs segments de données. Sur la Figure 11.22, l'envoi de trois segments correspond à un seul numéro d'acquittement.

De nombreuses implémentations TCP font en sorte de diminuer le nombre de segments d'acquittement envoyés. Par exemple, une implémentation TCP peut n'envoyer qu'un acquittement pour deux segments de données.

Figure 11.22

Un seul acquittement pour plusieurs segments.



Comme les segments TCP sont encapsulés dans des datagrammes IP, rien ne garantit qu'ils parviendront à destination dans le bon ordre. Le récepteur utilisera les numéros de séquence pour reconstruire un flux continu de données. Le récepteur n'acquitte pas forcément le numéro d'octet le plus haut, mais le numéro le plus haut qui soit précédé par une suite continue d'octets. Si une lacune existe (données non transmises), le récepteur ne peut acquitter que les numéros qui précèdent.

L'avantage des acquittements cumulés est que la perte d'un acquittement n'oblige pas forcément à une retransmission. Mais l'inconvénient est que le système des acquittements cumulés ne fonctionne plus lorsque des données sont perdues et qu'une lacune apparaît. Les données transmises après cette lacune ne seront jamais acquittées. Sur la Figure 11.23 les segments SEG2 et SEG3 ont bien été reçus, mais le segment SEG1 s'est perdu.



L'émetteur TCP n'est pas toujours averti lorsque la transmission réussit (source : Learning Tree).

- [illegible]

1. L'attente de l'émetteur excède la temporisation pour les trois segments. La retransmission de SEG2 et SEG3 surcharge alors inutilement le trafic.
2. L'attente de l'émetteur excède la temporisation pour SEG1 seulement. Le segment est retransmis et le récepteur peut acquitter les trois segments. Ici, le trafic n'est pas surchargé inutilement, mais l'émetteur doit attendre longtemps l'acquiescement et ne peut pas tirer parti de toute la largeur de sa fenêtre en attendant.

Lorsque différentes couches IP envoient le même message sur le même chemin, mais à des moments différents, le temps de transmission sera probablement différent, et ce en raison des facteurs suivants :

- Une liaison WAN envoie typiquement ses messages en série, les uns à la suite des autres. Si la liaison est occupée, le message reste bloqué dans une file d'attente.

- Il arrive que les routeurs qui connectent les liaisons WAN soient encombrés par un brusque accroissement du trafic. Le datagramme devra alors rester dans une file d'attente jusqu'à ce que le routeur lui indique une route. Après ce premier retard il arrive que le datagramme attende que l'interface réseau correspondant à la nouvelle route se libère. Dans le pire des cas, les datagrammes parviennent trop rapidement au routeur et sa mémoire s'engorge, ce qui l'oblige bientôt à rejeter des datagrammes parce qu'il ne peut plus les stocker en mémoire. Sachez qu'un tel scénario est tout à fait possible parce que dans un réseau IP il n'existe aucune allocation de mémoire préalable pour les différentes connexions TCP.

La couche IP est conçue de manière que la route des différents datagrammes IP soit déterminée séparément. Dans une connexion, les segments TCP successifs peuvent ne pas être envoyés sur le même chemin, aussi se transmettent-ils parfois à des vitesses très différentes, traversant divers ensembles de routeurs et de liaisons.

Lorsque le protocole TCP envoie un message il attend le retour des acquittements avant de déplacer sa fenêtre. Cette attente peut durer aussi longtemps que le prévoit la valeur de l'*intervalle de temporisation*. Etant donné l'importante variation des délais de transmission, TCP adapte la valeur du temporisateur. L'algorithme qui permet cette adaptation s'appelle *algorithme de retransmission adaptative*. En voici les étapes :

1. On mesure le temps qui s'écoule entre le moment où on envoie un octet de données identifié par son numéro de séquence et le moment où on reçoit l'accusé de réception correspondant à ce numéro. La valeur est notée RTT (*round-trip time* c'est-à-dire durée de l'aller-retour). A partir de cette valeur, on calcule une valeur corrigée notée SRTT (*smoothed round-trip time*) :

$$\text{SRTT} = (a * \text{SRTT}) + ((1 - a) * \text{RTT})$$

Dans cette formule, a est un facteur de pondération valant entre 0 et 1 ($0 \leq a < 1$).

La valeur SRTT est une estimation du temporisateur. Elle prend en compte l'ancienne valeur de SRTT et fait la moyenne pondérée avec la valeur RTT.

Lorsque $a = 0,5$ SRTT est simplement une moyenne arithmétique :

$$\text{SRTT} = (\text{SRTT} + \text{RTT})/2$$

Lorsque $a < 0,5$ SRTT attribue plus de poids à RTT. Si $a > 0,5$ SRTT attribue plus de poids à son ancienne valeur.

Lorsque $a = 0$ on a SRTT = RTT. Lorsque a s'approche de 1 la valeur de SRTT est pratiquement une constante. En pratique on n'utilise pas des valeurs aussi extrêmes pour a .

2. A partir de la valeur SRTT, on calcule le temporisateur de retransmission noté RTO (*retransmission time-out*) :

$$\text{RTO} = \min[\text{UBOUND}, \max[\text{LBOUND}, (b * \text{SRTT})]]$$



Le symbole UBOUND désigne la limite supérieure du temporisateur (par exemple 1 minute), et LBOUND désigne la limite inférieure (par exemple, 1 seconde). La valeur finale de RTO se situera donc entre ces deux valeurs. Le facteur b agit sur le délai. Si $b = 1$ RTO aura la même valeur que SRTT (à moins que celle-ci soit en dehors de l'intervalle encadré par LBOUND et UBOUND). Dans ce cas TCP sera rapidement averti de la perte des paquets et la retransmission ne tardera pas. En contrepartie, TCP risque d'effectuer des retransmissions inutiles. On peut donc abaisser la valeur de b . Le document RFC 793 suggère d'employer des valeurs situées entre 1,3 et 2.

En 1987, Karn et Partridge ont publié un article dans les actes du ACM SIG-COMM'87 (*Improving Round-Trip Estimates in Reliable Transport Protocols*). Cet article n'est pas devenu un document RFC, mais il a influencé de nombreuses implémentations TCP, en particulier celle de Unix BSD. L'algorithme correspondant est couramment appelé depuis *algorithme de Karn*. La documentation RFC 1122 recommande pour les réseaux sujets à la congestion de combiner l'algorithme de Karn avec l'algorithme prudent de Van Jacobson. L'algorithme de Van Jacobson vous sera présenté dans ce chapitre à la section "Réduire l'impact d'une congestion avec TCP". Voici en attendant l'explication de l'algorithme de Karn.

Dans l'algorithme de Karn on n'ajuste pas la valeur de SRTT, mais on corrige directement la valeur du temporisateur de repli. Karn considère que si une retransmission a eu lieu c'est qu'un événement grave s'est produit sur le réseau. L'algorithme implémente la stratégie de repli de temporisateur à l'aide d'une équation simple telle que :

$$\text{Backoff TIMEOUT} = G * \text{TIMEOUT}$$

Cette équation va servir aussi longtemps que le temporisateur se trouve au-dessous d'une certaine valeur définie au préalable (limite supérieure). Typiquement G vaut 2 et l'algorithme se comporte comme un algorithme de backoff exponentiel binaire. Cet algorithme n'affecte aucunement la valeur de SRTT. Lorsque TCP reçoit un acquittement pour un segment et qu'une retransmission n'est pas nécessaire, il mesure de nouveau la valeur RTT et recalcule SRTT et RTO.

• Réduire l'impact d'une congestion avec TCP

En cas de congestion, le module TCP réagit en ajustant la valeur du temporisateur, mais finit par effectuer des retransmissions lorsque l'attente de l'acquittement est trop longue. Or les retransmissions ne font qu'aggraver la congestion du réseau. En particulier, les retransmissions se succèdent parfois trop rapidement pour que le réseau puisse les stocker : on risque alors un *effondrement du réseau*. Il faut donc se doter d'une méthode pour arrêter ou au moins ralentir les retransmissions en cas de congestion.

En 1988 Van Jacobson a publié un article dans les actes d'ACM SIGCOMM'88 (*Congestion Avoidance and Control*). Cet article propose une technique permettant de répondre à une congestion en réduisant la quantité de données émises par l'expéditeur. Dans le Chapitre 8 nous avons étudié le message ICMP Type 4 Source Quench qui pouvait résoudre une congestion. Les routeurs envoient ce message à l'émetteur lorsqu'ils détectent une congestion. L'émetteur transmet alors ce message au module TCP. Dans le Chapitre 8, nous disions qu'il était impossible d'annuler le message ICMP Source Quench. La technique de Jacobson opère en ajustant dynamiquement avec un algorithme la taille de la fenêtre TCP pour résoudre la congestion.

L'algorithme de Jacobson est présent dans toutes les implémentations TCP récentes. En fait, selon RFC 1122 son implémentation est obligatoire, tout comme celle de l'algorithme de Karn.

L'algorithme de Jacobson tient compte de la valeur de la *taille de la fenêtre de congestion*. La taille réelle de la fenêtre est ajustée en choisissant la valeur minimum entre la taille de la fenêtre de congestion et la taille de la fenêtre TCP du récepteur :

Taille réelle de la fenêtre = $\min \{ \text{taille de la fenêtre de congestion, taille de la fenêtre TCP} \}$

Lorsque le réseau fonctionne normalement, la taille de la fenêtre de congestion est la même que la taille de la fenêtre TCP, donc la taille réelle de la fenêtre est celle de la fenêtre TCP. En cas de congestion, l'algorithme de Jacobson divise par deux la taille de la fenêtre de congestion chaque fois qu'un segment est perdu. La congestion se traduit donc par une décroissance exponentielle de la taille de la fenêtre. A la fin, la fenêtre TCP se réduira à un seul segment, c'est-à-dire que l'expéditeur n'enverra plus qu'un seul datagramme à la fois. En utilisant en parallèle l'algorithme de Karn qui multiplie chaque fois par deux la valeur du temporisateur de retransmission, on obtient un *repli exponentiel* ou *décroissance multiplicative*.

L'algorithme de Jacobson est dit algorithme "prudent" (*slow start*) parce qu'il ne rétablit la taille de la fenêtre que très progressivement, segment par segment. Ainsi, à la réception du premier acquittement qui fait suite à une congestion, la taille de la fenêtre passe de 1 à 2. Le module TCP peut désormais expédier deux segments à la fois. Si l'expéditeur reçoit trois acquittements supplémentaires, la taille de la fenêtre de congestion passe à 4, et quand les acquittements correspondant aux 4 segments expédiés reviennent, la taille de fenêtre de congestion passe à 8. L'accroissement de taille se poursuit ainsi jusqu'à ce que la taille de la fenêtre de congestion soit égale à la taille de la fenêtre du récepteur. Cette croissance est exponentielle et pour pouvoir envoyer N segments il suffit de $\log_2(N)$ transmissions.

* 20. *



LE MACMILLAN

Si la fenêtre de congestion grandit trop vite cela encombre brutalement le trafic alors que le réseau est à peine sorti de la congestion. Une nouvelle congestion réduit à nouveau la taille de la fenêtre de congestion. Pour éviter un accroissement trop rapide de la taille de la fenêtre, Jacobson propose une technique qu'il appelle *évitement de congestion* (*congestion avoidance*). Dès que la taille de la fenêtre de congestion vaut la moitié de la taille d'origine de la fenêtre TCP, elle n'augmente de un segment que lorsque tous les segments expédiés sont acquittés.

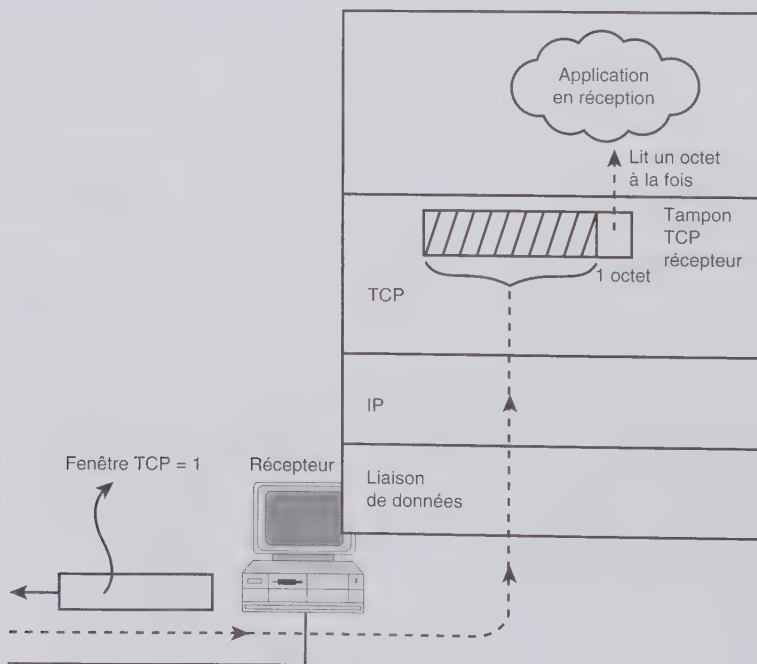
• Le syndrome SWS

Les premières implémentations de TCP présentaient une sorte de pathologie que l'on a baptisé *syndrome SWS* (pour *silly window syndrome* ou syndrome de la fenêtre stupide). En effet, dans certaines conditions, la transmission TCP ne s'effectuait plus qu'octet par octet. Un segment occupait donc un octet et il fallait acquitter séparément chaque octet de données. On observait le syndrome SWS dans les cas suivants :

- L'application se trouvant sur le récepteur lit des données octet par octet parce que son tampon est plein et TCP renvoie dans l'acquittement une taille de fenêtre égale à 1 (Figure 11.24).

Figure 11.24

Syndrome SWS provoqué par le récepteur.



- L'application se trouvant sur l'expéditeur génère les données octet par octet et TCP les envoie immédiatement (Figure 11.25).

On voit sur la Figure 11.24 que, le récepteur envoyant une taille de fenêtre égale à 1 octet, l'émetteur envoie un seul octet de données. Le récepteur reçoit cet octet de données et son module TCP spécifie de nouveau une taille de fenêtre égale à 1.

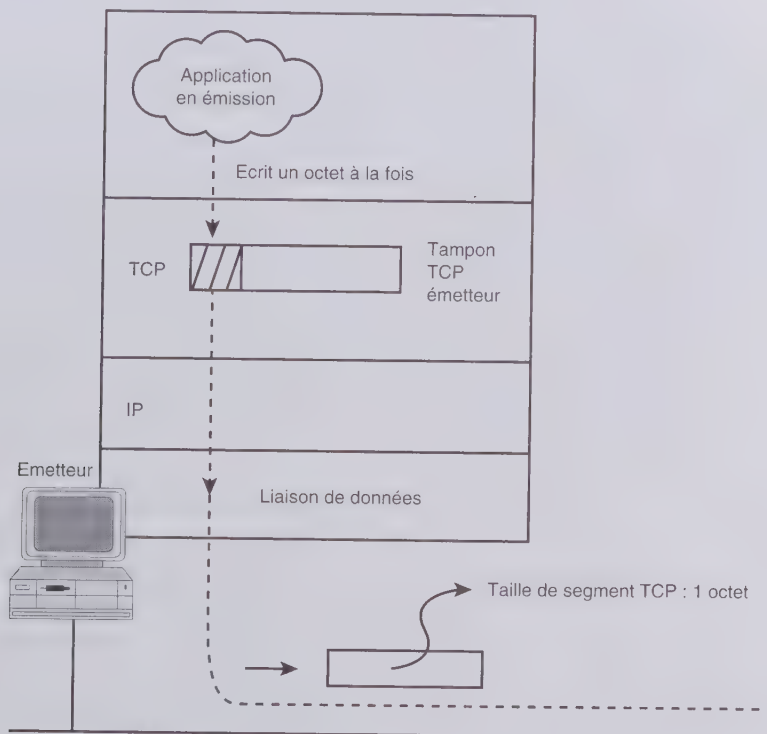
Sur la Figure 11.25 l'expéditeur envoie les données octet par octet. Il n'attend pas que les données s'accumulent et envoie les données à mesure que l'application les produit.

Quelle qu'en soit la cause, le syndrome SWS a des conséquences très fâcheuses :

- On génère un nombre de paquets trop important.
- Aussi bien au niveau du récepteur qu'au niveau de l'expéditeur, il faut que les couches liaison de données, IP et TCP traitent séparément chaque paquet. Au niveau des routeurs deux couches seulement interviennent pour traiter les paquets : liaison de données et IP.

Figure 11.25

Syndrome SWS provoqué par l'expéditeur.



- Calculer la route, le total de contrôle et gérer l'encapsulation et la décapsulation de chaque paquet entraîne une surcharge de travail dans toutes les couches concernées. Par ailleurs, il faut attribuer un numéro de séquence et d'acquiescement distinct chaque fois et les gérer avec des tampons émission/réception.
- Le rapport entre la place occupée par l'en-tête et la quantité de données transportées est beaucoup trop important : le protocole devient inefficace. Le débit des données est bas.
- Etant donné qu'il faut acquiescer chaque octet, les délais s'accumulent et abaissent encore le débit des données. Soient 1 000 paquets de 1 octet sur une liaison dont le délai est 5 ms. Le délai cumulé sera de 5 secondes ($1\,000 * 5\text{ ms}$). Si le temps de transmission sur la liaison est faible comparé au délai de la liaison, on a un débit global égal à $1\,000 * 8\text{ bits/octet} \div 5\text{ secondes} = 1\,600\text{ bit/s}$.

Pour prévenir le syndrome SWS les implémentations suivantes de TCP ont fait appel à diverses heuristiques :

- **Retarder la notification de la taille de la fenêtre.** Si la taille de fenêtre reçue est inférieure à une certaine valeur le récepteur retarde la notification de la taille de la fenêtre jusqu'à ce qu'elle atteigne :
min ($0.5 * \text{taille du tampon du récepteur}$, taille de segment maximale).
- **Retarder l'acquiescement.** Puisque les acquiescements contiennent la taille de fenêtre, il faut aussi les retarder. En conséquence, le trafic va diminuer sur le réseau. Les acquiescements TCP étant cumulés on peut se permettre d'attendre que les données s'accumulent sans risquer d'encombrer brusquement le réseau lorsque l'on décidera de les acquiescer. Le seul problème est que l'expéditeur peut s'impatier et retransmettre les segments avant d'en recevoir acquiescement. C'est pourquoi on ne retarde jamais les acquiescements plus de 500 millisecondes, voire 200 millisecondes dans de nombreuses implémentations. Par ailleurs il faut que le récepteur puisse estimer la valeur de RTT pour utiliser l'algorithme de correction du temporisateur. Il faut donc acquiescer tous les autres segments de données.

Plutôt que retarder l'acquiescement on peut adopter une démarche différente. On envoie sans retard l'acquiescement des paquets reçus, mais sans notifier un accroissement de la fenêtre, et ce tant que l'on n'atteint pas une certaine taille de fenêtre. Cependant, cette approche n'est pas recommandée par les standards TCP.

Côté expéditeur on prévient le syndrome SWS grâce à l'algorithme d'évitement de Nagle. En voici les principes :

- L'expéditeur envoie le premier segment de données, mais pas forcément les segments suivants s'ils sont de petite taille (même lorsque l'application active le flag PSH).

- Les données ne sont envoyées que lorsque l'une des conditions suivantes est remplie :

1. La quantité de données peut remplir un segment ayant la taille maximale.
2. L'expéditeur reçoit un acquittement.
3. Un temporisateur arrive à expiration.

Si l'application génère les données assez rapidement, le tampon TCP se remplira assez vite pour atteindre la taille maximale de segment. Mais si l'expéditeur reçoit un acquittement, il envoie les données accumulées jusqu'alors même si elles forment un segment dont la taille est inférieure.

On peut désactiver l'algorithme d'évitement de Nagle pour les applications en temps réel qui ont besoin d'un délai minimum, et envoyer en même temps des quantités importantes de données.

• Gérer une connexion TCP rompue

Une connexion TCP peut se rompre lorsque la machine distante connaît une défaillance ou que les liaisons du réseau sont défectueuses. Plusieurs moyens permettent d'avertir TCP de cette rupture. D'abord, si un message ICMP lui indique qu'il est impossible d'atteindre l'adresse de destination, après quelques retransmissions infructueuses, TCP avertira l'application. Mais les messages ICMP peuvent se perdre lors de la rupture de la connexion. TCP tentera un certain nombre de retransmissions (premier palier). En cas d'échec, TCP va demander à la couche IP de vérifier si le problème ne vient pas de la défaillance d'un routeur ou de la route. IP recalcule une route et pendant ce temps, TCP effectue une deuxième vague de retransmissions (second palier). En cas d'échec, TCP déclare que la connexion est rompue et doit libérer toutes les ressources associées à cette connexion, en particulier les tampons de contrôle de transmission (TCB).

Lorsqu'aucune donnée n'a été échangée entre les deux extrémités d'un lien, celle-ci passe à l'état inactif. La connexion occupe alors encore les ressources, même si la liaison ou l'une de ses extrémités est en fait défaillante. Dans certaines implémentations, TCP envoie un message de conservation (*keep-alive message*) qui vérifie périodiquement que la connexion est toujours opérationnelle. Si ce message n'est pas acquitté, TCP considère que la connexion est rompue. Il est suggéré d'émettre un message de conservation toutes les deux heures, mais dans certaines implémentations les vérifications se répètent toutes les 10 ou 15 minutes. Le principal est de rester cohérent avec les temporisateurs des applications



qui s'exécutent généralement sur l'hôte TCP. Un temporisateur trop bref encombrerait le trafic, mais un temporisateur trop important empêcherait TCP d'être averti rapidement en cas de rupture de connexion.

• TCP comme automate à états finis

La Figure 11.26 présente TCP comme un automate à états finis, c'est-à-dire un modèle logique qui représente le comportement du système et le changement de son état interne en réponse à divers événements. Les boîtes contiennent les états de l'automate et les flèches représentent le passage d'un état à l'autre. Les libellés qui figurent sur les flèches se rapportent à l'événement qui provoque le changement d'état et à la réponse de TCP :

événement

réponse

Un x pour la réponse indique que TCP ne prévoit pas de réponse particulière.

Les extrémités de la connexion TCP commencent à l'état fermé. De là, une connexion peut s'établir soit par une ouverture active (active OPEN), soit par une ouverture passive (passive OPEN).

Une ouverture active (active OPEN) provoque la création d'un tampon TCB et l'envoi d'un segment de flag SYN par TCP qui passe à l'état SYN SENT. Lorsque l'autre extrémité renvoie le flag SYN et le flag ACK TCP envoie un acquittement qui met fin à la négociation en trois temps puis passe à l'état CONNECTION ESTABLISHED.

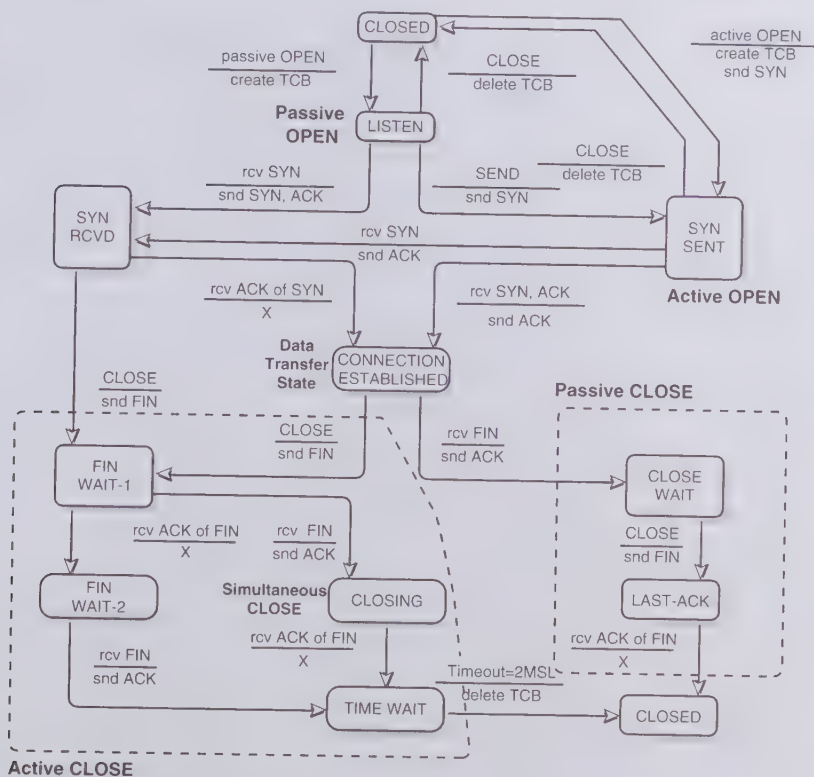
Une ouverture passive (passive OPEN) provoque la création d'un tampon TCB par TCP qui passe à l'état LISTEN. Lorsque l'autre extrémité lui envoie le flag SYN de la requête d'ouverture active, TCP renvoie dans un segment le numéro de séquence avec les flags SYN et ACK, puis il passe à l'état SYN RCVD. A la réception d'un flag ACK ou SYN provenant de l'autre extrémité, la connexion TCP passe à l'état CONNECTION ESTABLISHED.

TCP passe à l'état TIME WAIT lorsque l'on ferme la connexion. Cela permet de respecter un délai égal à deux fois MSL (*maximum segment lifetime* ou durée de vie maximale d'un segment) et d'éviter les doublons dans les numéros de segment (voir la section "Le champ Flag").

Les états FIN WAIT-1 et FIN WAIT-2 interviennent tous les deux dans le mécanisme de clôture de la connexion TCP parce qu'il nécessite l'accord des deux extrémités.

Figure 11.26

TCP comme machine
à états finis.



MSL= Maximum Segment Lifetime (2 minutes per RFC 793)

TCB = Transmission Control Buffer (contains state information on the TCP end point)

• Les traces TCP

Le protocole Telnet utilise TCP et pourra nous servir à étudier sa trace. La Figure 11.27 présente une trace Telnet complète correspondant à un utilisateur effectuant les opérations suivantes :

1. Logage au serveur Telnet.
2. Exécution d'un certain nombre de commandes sur le serveur Telnet.
3. Délogage du serveur Telnet.



CH. 11 : PROTOCOLES DE TRANSFERT

Figure 11.27

Une trace Telnet.

No.	Source	Destination	Layer	Size	Summary
1	0000C024282D	FFFFFFFFFFFF	arp	0064	Req by 144.19.74.44 for 144.19.7
2	0000C024282D	FFFFFFFFFFFF	arp	0064	Req by 144.19.74.44 for 144.19.7
3	0000C0A20F8E	0000C024282D	arp	0064	Reply 144.19.74.201=0000C0A20F8E
4	0000C024282D	0000C0A20F8E	tcp	0064	Port:6295 ---> TELNET SYN
5	0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK SYN
6	0000C024282D	0000C0A20F8E	telnt	0069	Cmd=Do; Code=Echo; Cmd=Do; Code=
7	0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
8	0000C0A20F8E	0000C024282D	telnt	0064	Cmd=Do; Code=;
9	0000C024282D	0000C0A20F8E	telnt	0064	Cmd=Won't; Code=;
10	0000C0A20F8E	0000C024282D	telnt	0067	Cmd=Will; Code=Echo; Cmd=Will; C
11	0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
12	0000C024282D	0000C0A20F8E	telnt	0072	Cmd=Do; Code=Suppress Go Ahead;
13	0000C0A20F8E	0000C024282D	telnt	0070	Cmd=Do; Code=Terminal Type; Cmd=
14	0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
15	0000C024282D	0000C0A20F8E	telnt	0072	Cmd=Will; Code=Terminal Type; Cm
16	0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
17	0000C0A20F8E	0000C024282D	telnt	0064	Cmd=Subnegotiation Begin; Code=T
18	0000C024282D	0000C0A20F8E	telnt	0071	Cmd=Subnegotiation Begin; Code=T
19	0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
20	0000C0A20F8E	0000C024282D	telnt	0067	Cmd=Do; Code=Echo; Cmd=Will; Cod
21	0000C024282D	0000C0A20F8E	telnt	0069	Cmd=Won't; Code=Echo; Cmd=Don't;
22	0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
23	0000C0A20F8E	0000C024282D	telnt	0104	Data=..Linux 1.3.50 (ltreec1.ltr
24	0000C024282D	0000C0A20F8E	tcp	0064	Port:6295 ---> TELNET ACK
25	0000C0A20F8E	0000C024282D	telnt	0064	Data=..
26	0000C024282D	0000C0A20F8E	tcp	0064	Port:6295 ---> TELNET ACK
27	0000C0A20F8E	0000C024282D	telnt	0073	Data=ltreec1 login:
28	0000C024282D	0000C0A20F8E	tcp	0064	Port:6295 ---> TELNET ACK
29	0000C024282D	0000C0A20F8E	telnt	0064	Data=u
30	0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
31	0000C0A20F8E	0000C024282D	telnt	0064	Data=u
32	0000C024282D	0000C0A20F8E	tcp	0064	Port:6295 ---> TELNET ACK
33	0000C024282D	0000C0A20F8E	telnt	0064	Data=s
34	0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
35	0000C0A20F8E	0000C024282D	telnt	0064	Data=s
36	0000C024282D	0000C0A20F8E	tcp	0064	Port:6295 ---> TELNET ACK
37	0000C024282D	0000C0A20F8E	telnt	0064	Data=e
38	0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
39	0000C0A20F8E	0000C024282D	telnt	0064	Data=e
40	0000C024282D	0000C0A20F8E	tcp	0064	Port:6295 ---> TELNET ACK
41	0000C024282D	0000C0A20F8E	telnt	0064	Data=r
42	0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
43	0000C0A20F8E	0000C024282D	telnt	0064	Data=r
44	0000C024282D	0000C0A20F8E	tcp	0064	Port:6295 ---> TELNET ACK
45	0000C024282D	0000C0A20F8E	telnt	0064	Data=1
46	0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
47	0000C0A20F8E	0000C024282D	telnt	0064	Data=1
48	0000C024282D	0000C0A20F8E	tcp	0064	Port:6295 ---> TELNET ACK
49	0000C024282D	0000C0A20F8E	telnt	0064	Data=..
50	0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
51	0000C0A20F8E	0000C024282D	telnt	0064	Data=..
52	0000C024282D	0000C0A20F8E	tcp	0064	Port:6295 ---> TELNET ACK
53	0000C0A20F8E	0000C024282D	telnt	0068	Data=Password:
54	0000C024282D	0000C0A20F8E	tcp	0064	Port:6295 ---> TELNET ACK
55	0000C024282D	0000C0A20F8E	telnt	0064	Data=u
56	0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
57	0000C024282D	0000C0A20F8E	telnt	0064	Data=s
58	0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
59	0000C024282D	0000C0A20F8E	telnt	0064	Data=e
60	0000C0A20F8E	0000C024282D	tcp	0064	Port:TELNET ---> 6295 ACK
61	0000C024282D	0000C0A20F8E	telnt	0064	Data=r

Figure 11.27

Une trace Telnet.
(suite)

```

62 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK
63 0000C024282D 0000C0A20F8E telnt 0064 Data=1
64 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK
65 0000C024282D 0000C0A20F8E telnt 0064 Data=p
66 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK
67 0000C024282D 0000C0A20F8E telnt 0064 Data=w
68 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK
69 0000C024282D 0000C0A20F8E telnt 0064 Data=.
70 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK
71 0000C0A20F8E 0000C024282D telnt 0064 Data=...
72 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
73 0000C0A20F8E 0000C024282D telnt 0113 Data=Last login: Wed May 28 18
74 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
75 0000C0A20F8E 0000C024282D telnt 0073 Data=Linux 1.3.50...
76 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
77 0000C0A20F8E 0000C024282D telnt 0064 Data=.
78 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
79 0000C0A20F8E 0000C024282D telnt 0069 Data=Power, n...
80 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
81 0000C0A20F8E 0000C024282D telnt 0119 Data=.The only narcotic regula
82 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
83 0000C0A20F8E 0000C024282D telnt 0064 Data=.
84 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
85 0000C0A20F8E 0000C024282D telnt 0069 Data=ltrec1:-$
86 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
87 0000C0A20F8E 0000C024282D telnt 0064 Data=l
88 0000C024282D 0000C0A20F8E tcp 0064 Port:TELNET ---> 6295 ACK
89 0000C0A20F8E 0000C024282D telnt 0064 Data=l
90 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
91 0000C0A20F8E 0000C024282D telnt 0064 Data=s
92 0000C024282D 0000C0A20F8E tcp 0064 Port:TELNET ---> 6295 ACK
93 0000C0A20F8E 0000C024282D telnt 0064 Data=s
94 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
95 0000C0A20F8E 0000C024282D telnt 0064 Data=
96 0000C024282D 0000C0A20F8E tcp 0064 Port:TELNET ---> 6295 ACK
97 0000C0A20F8E 0000C024282D telnt 0064 Data=
98 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
99 0000C0A20F8E 0000C024282D telnt 0064 Data=
100 0000C024282D 0000C0A20F8E tcp 0064 Port:TELNET ---> 6295 ACK
101 0000C0A20F8E 0000C024282D telnt 0064 Data=
102 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
103 0000C0A20F8E 0000C024282D telnt 0064 Data=a
104 0000C024282D 0000C0A20F8E tcp 0064 Port:TELNET ---> 6295 ACK
105 0000C0A20F8E 0000C024282D telnt 0064 Data=a
106 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
107 0000C0A20F8E 0000C024282D telnt 0064 Data=l
108 0000C024282D 0000C0A20F8E tcp 0064 Port:TELNET ---> 6295 ACK
109 0000C0A20F8E 0000C024282D telnt 0064 Data=l
110 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
111 0000C0A20F8E 0000C024282D telnt 0064 Data=.
112 0000C024282D 0000C0A20F8E tcp 0064 Port:TELNET ---> 6295 ACK
113 0000C0A20F8E 0000C024282D telnt 0064 Data=.
114 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
115 0000C0A20F8E 0000C024282D telnt 0067 Data=total 7..
116 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
117 0000C0A20F8E 0000C024282D telnt 0130 Data=drwxr-xr-x 3 user1 u
118 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
119 0000C0A20F8E 0000C024282D telnt 0131 Data=drwxr-xr-x 19 root r
120 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
121 0000C0A20F8E 0000C024282D telnt 0138 Data=-rw-r--r-- 1 user1 u
122 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
123 0000C0A20F8E 0000C024282D telnt 0132 Data=-rw-r--r-- 1 user1 u

```



Figure 11.27

Une trace Telnet.
(suite)

```

124 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
125 0000C0A20F8E 0000C024282D telnt 0130 Data=-rw-r--r-- 1 user1
126 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
127 0000C0A20F8E 0000C024282D telnt 0132 Data=-rw-r--r-- 1 user1
128 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
129 0000C0A20F8E 0000C024282D telnt 0134 Data=drwxr-xr-x 2 user1
130 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
131 0000C0A20F8E 0000C024282D telnt 0064 Data=l
132 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
133 0000C0A20F8E 0000C024282D telnt 0068 Data=treec1:~$
134 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
135 0000C024282D 0000C0A20F8E telnt 0064 Data=p
136 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK
137 0000C0A20F8E 0000C024282D telnt 0064 Data=p
138 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
139 0000C024282D 0000C0A20F8E telnt 0064 Data=s
140 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK
141 0000C0A20F8E 0000C024282D telnt 0064 Data=s
142 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
143 0000C024282D 0000C0A20F8E telnt 0064 Data=
144 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK
145 0000C0A20F8E 0000C024282D telnt 0064 Data=
146 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
147 0000C024282D 0000C0A20F8E telnt 0064 Data=-
148 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK
149 0000C0A20F8E 0000C024282D telnt 0064 Data=-
150 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
151 0000C024282D 0000C0A20F8E telnt 0064 Data=a
152 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK
153 0000C0A20F8E 0000C024282D telnt 0064 Data=a
154 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
155 0000C024282D 0000C0A20F8E telnt 0064 Data=l
156 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK
157 0000C0A20F8E 0000C024282D telnt 0064 Data=l
158 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
159 0000C024282D 0000C0A20F8E telnt 0064 Data=x
160 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK
161 0000C0A20F8E 0000C024282D telnt 0064 Data=x
162 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
163 0000C024282D 0000C0A20F8E telnt 0064 Data=..
164 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK
165 0000C0A20F8E 0000C024282D telnt 0064 Data=..
166 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
167 0000C0A20F8E 0000C024282D telnt 0133 Data= F UID PID PPID PRI
168 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
169 0000C0A20F8E 0000C024282D telnt 0140 Data= 0 0 1 0 36
170 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
171 0000C0A20F8E 0000C024282D telnt 0143 Data= 0 0 2 1 36
172 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
173 0000C0A20F8E 0000C024282D telnt 0140 Data= 0 0 8 1 36
174 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
175 0000C0A20F8E 0000C024282D telnt 0140 Data= 0 0 24 1 36
176 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
177 0000C0A20F8E 0000C024282D telnt 0140 Data= 0 0 38 1 36
178 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
179 0000C0A20F8E 0000C024282D telnt 0140 Data= 0 0 40 1 36
180 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
181 0000C0A20F8E 0000C024282D telnt 0140 Data= 0 1 42 1 36
182 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
183 0000C0A20F8E 0000C024282D telnt 0140 Data= 0 0 44 1 36
184 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
185 0000C0A20F8E 0000C024282D telnt 0139 Data= 0 0 46 1 36

```

Sur la Figure 11.27 les paquets 1 à 3 concernant une résolution ARP (voir Chapitre 5). La trace Telnet proprement dite ne commence qu'à partir du quatrième paquet.

Figure 11.27

Une trace Telnet.
(suite)

```

186 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
187 0000C0A20F8E 0000C024282D telnt 0140 Data= 0 0 49 1 30
188 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
189 0000C0A20F8E 0000C024282D telnt 0140 Data= 0 0 51 1 30
190 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
191 0000C0A20F8E 0000C024282D telnt 0140 Data= 0 0 53 1 30
192 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
193 0000C0A20F8E 0000C024282D telnt 0140 Data= 0 0 57 1 30
194 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
195 0000C0A20F8E 0000C024282D telnt 0140 Data= 0 0 61 1 30
196 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
197 0000C0A20F8E 0000C024282D telnt 0140 Data= 0 0 62 1 30
198 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
199 0000C0A20F8E 0000C024282D telnt 0140 Data= 0 0 63 1 30
200 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
201 0000C0A20F8E 0000C024282D telnt 0140 Data= 0 0 64 1 30
202 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
203 0000C0A20F8E 0000C024282D telnt 0140 Data= 0 0 65 1 30
204 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
205 0000C0A20F8E 0000C024282D telnt 0131 Data= 0 0 145 1 30
206 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
207 0000C024282D 0000C0A20F8E telnt 0064 Data=,
208 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK
209 0000C0A20F8E 0000C024282D telnt 0136 Data= 0 0 258 44 29
210 0000C0A20F8E 0000C024282D telnt 0064 Cmd=;
211 0000C024282D 0000C0A20F8E telnt 0064 Data=,
212 0000C0A20F8E 0000C024282D telnt 0064 Data=,
213 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK
214 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
215 0000C0A20F8E 0000C024282D telnt 0071 Data=..litrec1:-$
216 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
217 0000C0A20F8E 0000C024282D telnt 0064 Data=1
218 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK
219 0000C0A20F8E 0000C024282D telnt 0064 Data=1
220 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
221 0000C024282D 0000C0A20F8E telnt 0064 Data=o
222 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK
223 0000C0A20F8E 0000C024282D telnt 0064 Data=o
224 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
225 0000C024282D 0000C0A20F8E telnt 0064 Data=g
226 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK
227 0000C0A20F8E 0000C024282D telnt 0064 Data=g
228 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
229 0000C024282D 0000C0A20F8E telnt 0064 Data=o
230 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK
231 0000C0A20F8E 0000C024282D telnt 0064 Data=o
232 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
233 0000C024282D 0000C0A20F8E telnt 0064 Data=u
234 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK
235 0000C0A20F8E 0000C024282D telnt 0064 Data=u
236 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
237 0000C024282D 0000C0A20F8E telnt 0064 Data=t
238 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK
239 0000C0A20F8E 0000C024282D telnt 0064 Data=t
240 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK
241 0000C024282D 0000C0A20F8E telnt 0064 Data=..
242 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK
243 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK FIN
244 0000C024282D 0000C0A20F8E tcp 0064 Port:6295 ---> TELNET ACK FIN
245 0000C0A20F8E 0000C024282D tcp 0064 Port:TELNET ---> 6295 ACK

```



Les sections qui suivent s'attarderont sur les aspects de la trace Telnet qui concernent TCP. Nous présentons la Figure 11.27 pour que vous puissiez vous référer facilement aux numéros de paquet correspondant aux traces que nous étudierons.

La trace TCP à l'ouverture de la connexion (négociation en trois temps)

Les Figures 11.28 à 11.30 vous montrent les trois paquets TCP différents qui interviennent lors de la négociation en trois temps, nécessaire à l'ouverture d'une connexion TCP.

Le paquet de la requête d'ouverture de connexion TCP constitue la première étape de la négociation. Il contient un numéro de séquence et les flags suivants :

Sequence Numbers = 25784320 (ISS)

Acknowledgment Number = 0 (non valide car ACK = 0)

Autre élément notable, le champ Data Offset contient la valeur 6 qui indique une option TCP. Cette option se trouve placée dans l'option MSS (option-kind=2) et contient les codes hexadécimaux suivants :

02 04 04 00

Ces codes se traduisent par les options TCP que voici :

Option:	MAXIMUM SEGMENT SIZE
Option Length:	4
Maximum Segment Size :	1024

Les valeurs de port d'origine, de port de destination et de taille de fenêtre pour le premier paquet sont les suivants :

Source Port:	6295
Destination Port:	23 (port standard du serveur Telnet)
Window:	4096

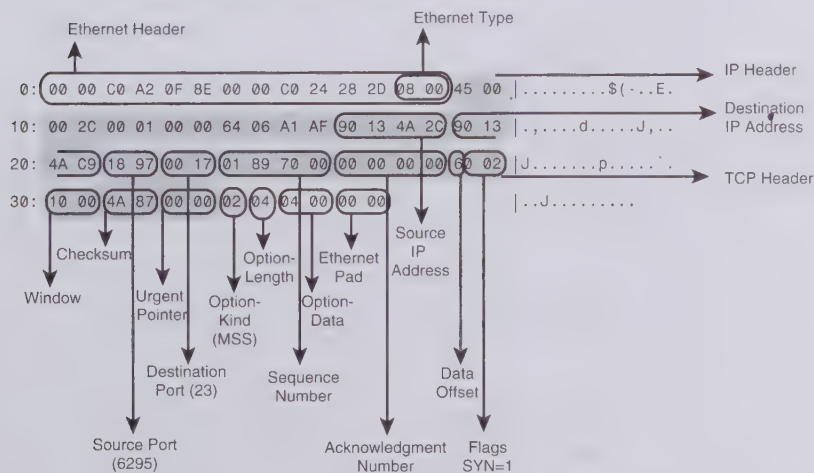
Le port d'origine correspond au port sur lequel le client Telnet recevra les données en provenance du serveur Telnet. Le port de destination est le port sur lequel on contacte le serveur Telnet à l'ouverture de la connexion. La taille de fenêtre est le nombre d'octets que le client Telnet peut recevoir, c'est-à-dire la taille du tampon de réception TCP du client Telnet.

Figure 11.28

*La requête OPEN TCP
envoie le numéro
de séquence de l'expéditeur
(ISS).*

```

Packet Number : 4          6:38:38 PM
Length : 64 bytes
ether: ===== Ethernet Datalink Layer =====
      Station: 00-00-C0-24-28-2D ----> 00-00-C0-A2-0F-8E
      Type: 0x0800 (IP)
ip: ===== Internet Protocol =====
      Station: 144.19.74.44 ----> 144.19.74.201
      Protocol: TCP
      Version: 4
      Header Length (32 bit words): 5
      Precedence: Routine
                Normal Delay, Normal Throughput, Normal Reliability
      Total length: 44
      Identification: 1
      Fragmentation allowed, Last fragment
      Fragment Offset: 0
      Time to Live: 100 seconds
      Checksum: 0xA1AF(Valid)
tcp: ===== Transmission Control Protocol =====
      Source Port: 6295
      Destination Port: TELNET
      Sequence Number: 25784320
      Acknowledgement Number: 0
      Data Offset (32-bit words): 6
      Window: 4096
      Control Bits: Synchronize Sequence Numbers (SYN)
      Checksum: 0x4A87(Valid)
      Urgent Pointer: 0
      Option: MAXIMUM SEGMENT SIZE
                Option Length: 4
                Maximum Segment Size : 1024
  
```



La Figure 11.29 présente l'acquittement correspondant à la requête d'ouverture de connexion. C'est la deuxième étape de la négociation :

SYN = 1

ACK = 1

Tous autres flags TCP à zéro

Sequence Number = 1508167651 (IRS)

Acknowledgment Number = 25784321 (ISS + 1)

Le serveur Telnet envoie le paquet de la Figure 11.29 pour synchroniser les numéros de séquence. Ce paquet contient l'ISS (numéro de séquence du serveur). Le champ Data Offset contient la valeur 6, indiquant une option TCP. Cette option se trouve dans l'option MSS (option-kind=2) et contient les codes hexadécimaux suivants :

02 04 04 00

Ces codes se traduisent par les options TCP que voici :

Option:	MAXIMUM SEGMENT SIZE
Option Length:	4
Maximum Segment Size :	1024

Dans cet exemple le serveur et le client Telnet reconnaissent la même valeur pour la taille maximum des segments (MSS). Ce n'est pas le cas général. Les valeurs de port d'origine, de port de destination et de taille de fenêtre pour le paquet d'acquittement sont les suivants :

Source Port:	6295
Destination Port:	23 (port standard du serveur Telnet)
Window:	30719

Le port d'origine correspond au port sur lequel le serveur Telnet recevra les données en provenance du client Telnet. Le port de destination est le port du client Telnet d'où provient la requête d'ouverture de la connexion. La taille de fenêtre est le nombre d'octets que le serveur Telnet peut recevoir, c'est-à-dire la taille du tampon de réception TCP du serveur Telnet. Remarquez que cette valeur est supérieure à la taille de fenêtre du client.

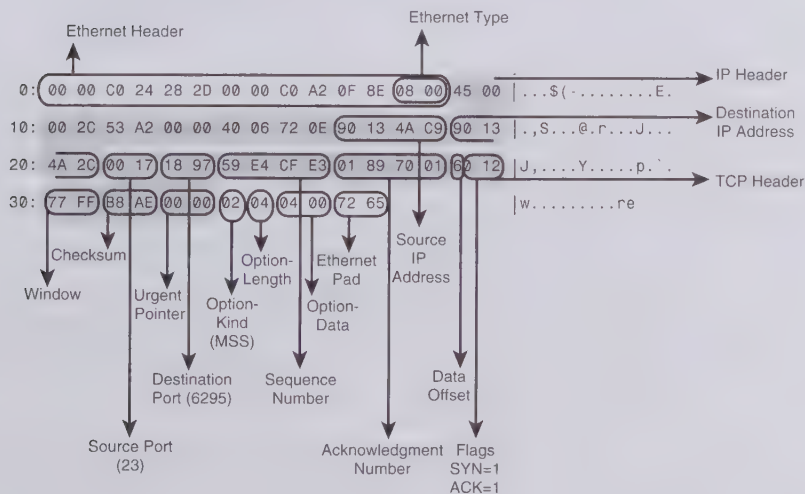
Figure 11.29

*Acquittement de la requête
OPEN : envoi du numéro de
séquence du récepteur (IRS).*

```

Packet Number : 5           6:38:38 PM
Length : 64 bytes
ether: ===== Ethernet Datalink Layer =====
      Station: 00-00-C0-A2-0F-8E -.-.-> 00-00-C0-24-28-2D
      Type: 0x0800 (IP)
ip: ===== Internet Protocol =====
      Station: 144.19.74.201 -.-.-> 144.19.74.44
      Protocol: TCP
      Version: 4
      Header Length (32 bit words): 5
      Precedence: Routine
                Normal Delay, Normal Throughput, Normal Reliability
      Total length: 44
      Identification: 21410
      Fragmentation allowed, Last fragment
      Fragment Offset: 0
      Time to Live: 64 seconds
      Checksum: 0x720E(Valid)
tcp: ===== Transmission Control Protocol =====
      Source Port: TELNET
      Destination Port: 6295
      Sequence Number: 1508167651
      Acknowledgement Number: 25784321
      Data Offset (32-bit words): 6
      Window: 30719
      Control Bits: Acknowledgement Field is Valid (ACK)
                    Synchronize Sequence Numbers (SYN)
      Checksum: 0xB8AE(Valid)
      Urgent Pointer: 0
      Option: MAXIMUM SEGMENT SIZE
              Option Length: 4
              Maximum Segment Size : 1024

```



La Figure 11.30 présente la dernière étape de la négociation. Le client Telnet acquitte le numéro de séquence envoyé par le serveur. Voici les flags et le numéro de séquence du paquet correspondant :

ACK = 1

PSH = 1

Tous autres flags TCP à zéro

Sequence Number = 25784321 (ISS + 1)

Acknowledgment Number = 1508167652 (IRS + 1)

Le flag PSH est activé pour suggérer à TCP d'envoyer les données. C'est ici que la technique d'évitement de Nagle intervient pour éviter le syndrome SWS (voir "Le syndrome SWS").

Le champ Data Offset contient la valeur 5 qui n'indique aucune option TCP.

Les valeurs de port d'origine, de port de destination et de taille de fenêtre pour ce paquet sont les suivants :

Source Port:	6295
Destination Port:	23 (standard Telnet server port)
Window:	4096

Le port d'origine correspond à l'extrémité de la connexion TCP associée au client Telnet. Le port de destination correspond à l'extrémité de la connexion TCP associée au serveur Telnet. La taille de fenêtre est le nombre d'octets que le client Telnet peut recevoir, c'est-à-dire la taille du tampon de réception TCP du client Telnet.

Trace TCP concernant les données

Dans le paquet 6 (voir Figure 11.30) qui clôt la négociation en trois temps et acquitte l'IRS envoyé par le récepteur, on trouve neuf octets de données Telnet :

FF FD 01 FF FD 03 FF FB 1F

Voici comment nous sommes arrivés à cette conclusion. Dans l'en-tête IP on trouve la taille du datagramme (voir Figure 11.30) : 49 octets. Sur ces 49 octets, 20 sont occupés par l'en-tête IP (Header Length = 5 mots) et 20 autres par l'en-tête TCP (Data Offset = 5 words). Or, $49 - 20 - 20 = 9$. Donc il reste 9 octets de données Telnet.

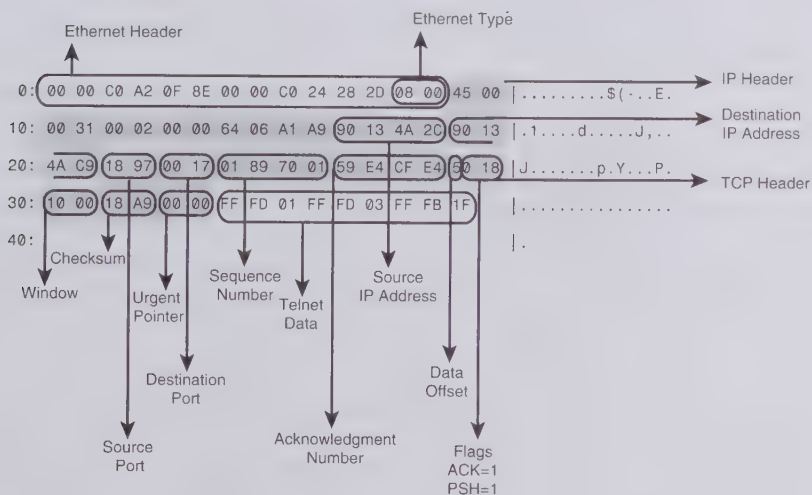
Figure 11.30

*Acquittement du numéro de
séquence du récepteur (IRS).*

```

Packet Number : 6           6:38:38 PM
Length : 69 bytes
ether: ===== Ethernet Datalink Layer =====
      Station: 00-00-C0-24-28-2D ---> 00-00-C0-A2-0F-8E
      Type: 0x0800 (IP)
  ip: ===== Internet Protocol =====
      Station: 144.19.74.44 ---> 144.19.74.201
      Protocol: TCP
      Version: 4
      Header Length (32 bit words): 5
      Precedence: Routine
            Normal Delay, Normal Throughput, Normal Reliability
      Total length: 49
      Identification: 2
      Fragmentation allowed, Last fragment
      Fragment Offset: 0
      Time to Live: 100 seconds
      Checksum: 0xA1A9(Valid)
  tcp: ===== Transmission Control Protocol =====
      Source Port: 6295
      Destination Port: TELNET
      Sequence Number: 25784321
      Acknowledgement Number: 1508167652
      Data Offset (32-bit words): 5
      Window: 4096
      Control Bits: Acknowledgement Field is Valid (ACK)
                    Push Function Requested (PSH)
      Checksum: 0x18A9(Valid)
      Urgent Pointer: 0
  telnt: ===== Telnet Protocol =====
      Command: Do
      Option Code: Echo
      Command: Do
      Option Code: Suppress Go Ahead
      Command: Will
      Option Code:

```



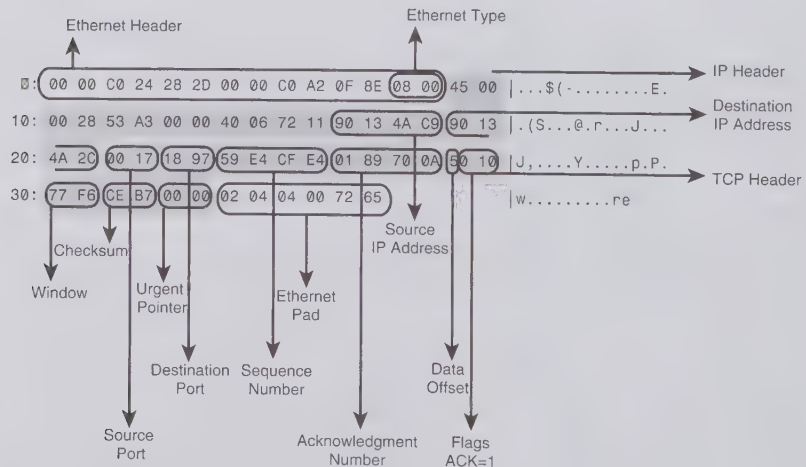
Le paquet 6 correspond au début de la transmission des données dans la trace TCP. Le paquet 7 (voir Figure 11.31) contient un acquittement envoyé par le serveur Telnet. Comme le serveur ne dispose d'aucune donnée à envoyer, il se contente d'envoyer un acquittement.

Figure 11.31

Le paquet d'acquiescement.

```

Packet Number : 7          6:38:38 PM
Length : 64 bytes
ether: ===== Ethernet Datalink Layer =====
      Station: 00-00-C0-A2-0F-8E ----> 00-00-C0-24-28-2D
      Type: 0x0800 (IP)
ip: ===== Internet Protocol =====
      Station: 144.19.74.201 ----> 144.19.74.44
      Protocol: TCP
      Version: 4
      Header Length (32 bit words): 5
      Precedence: Routine
          Normal Delay, Normal Throughput, Normal Reliability
      Total length: 40
      Identification: 21411
      Fragmentation allowed, Last fragment
      Fragment Offset: 0
      Time to Live: 64 seconds
      Checksum: 0x7211(Valid)
tcp: ===== Transmission Control Protocol =====
      Source Port: TELNET
      Destination Port: 6295
      Sequence Number: 1508167652
      Acknowledgement Number: 25784330
      Data Offset (32-bit words): 5
      Window: 30710
      Control Bits: Acknowledgement Field is Valid (ACK)
      Checksum: 0xCEB7(Valid)
      Urgent Pointer: 0
  
```



Voici les flags et les numéros de séquence de ce paquet :

ACK = 1

Tous autres flags TCP à zéro

Sequence Number: 1508167652

Acknowledgment Number: 25784330

Remarquez que le numéro d'acquittement renvoyé contient l'acquittement des neuf octets de données Telnet.

ISN from Telnet client = 25784321

Telnet data = 9

Acknowledgment Number data	= ISN from Telnet client + Telnet
	= 25784321 + 9
	= 25784330

Une seconde après l'envoi du paquet d'acquittement le serveur Telnet envoie le paquet 8 (Figure 11.32). Si le serveur avait pu disposer des données un peu plus tôt, il les aurait envoyées dans le paquet 7. TCP retarde autant que possible l'envoi des acquittements pour éviter de multiplier les paquets, mais l'attente ne peut pas excéder 500 millisecondes (voir "Le syndrome SWS"). Pour déterminer le temps écoulé entre la capture des paquets 7 et 8 il suffit de comparer les marques de temps des codes produits par l'analyseur de protocole (premières lignes).

Trace TCP de "fermeture gracieuse"

Les paquets 217 à 242 (voir Figure 11.27) indiquent comment le client Telnet transmet les unes après les autres les commandes de logout au serveur. Les paquets 243 à 245 montrent le mécanisme de fermeture gracieuse. Dans le paquet 243 (Figure 11.33) le serveur Telnet qui reçoit la commande de logout requiert la fermeture de la connexion TCP en affectant la valeur 1 au flag FIN. L'activation de ce flag indique que le serveur Telnet n'a plus aucune donnée à envoyer.



Figure 11.32

Transmission de données TCP.

```

Packet Number : 8           6:39:12 PM
Length : 64 bytes
ether: ===== Ethernet Datalink Layer =====
Station: 00-00-C0-A2-0F-8E ---> 00-00-C0-24-28-2D
Type: 0x0800 (IP)
ip: ===== Internet Protocol =====
Station: 144.19.74.201 ---> 144.19.74.44
Protocol: TCP
Version: 4
Header Length (32 bit words): 5
Precedence: Routine
        Normal Delay, Normal Throughput, Normal Reliability
Total length: 40
Identification: 21533
Fragmentation allowed, Last fragment
Fragment Offset: 0
Time to Live: 64 seconds
Checksum: 0x7197(Valid)
tcp: ===== Transmission Control Protocol =====
Source Port: TELNET
Destination Port: 6295
Sequence Number: 1508170215
Acknowledgement Number: 25784420
Data Offset (32-bit words): 5
Window: 30719
Control Bits: Acknowledgement Field is Valid (ACK)
               No More Data from Sender (FIN)
Checksum: 0xC450(Valid)
Urgent Pointer: 0

```

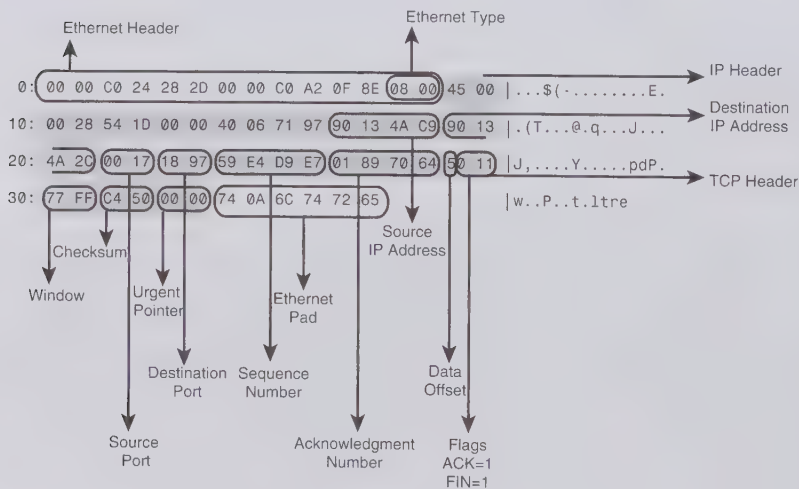


Figure 11.33

*La requête de fermeture est
envoyée par le serveur Telnet.*

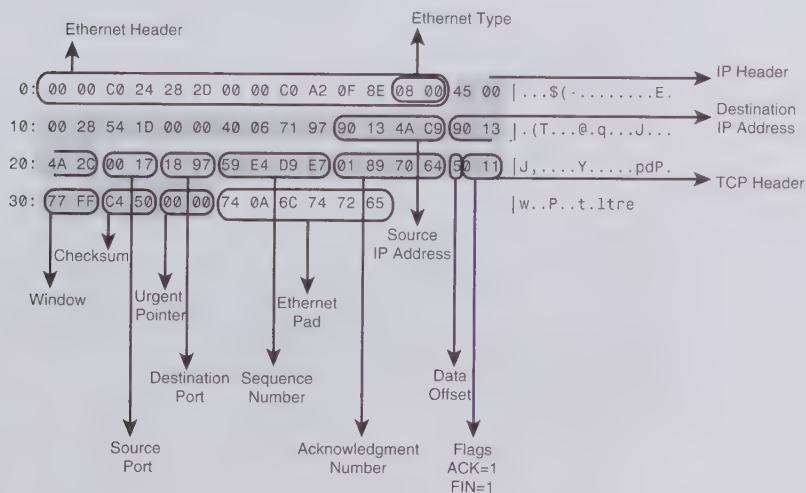
```

Packet Number : 243          6:39:12 PM
Length : 64 bytes
ether: ===== Ethernet Datalink Layer =====
Station: 00-00-C0-A2-0F-8E ----> 00-00-C0-24-28-2D
Type: 0x0800 (IP)

ip: ===== Internet Protocol =====
Station: 144.19.74.201 ----> 144.19.74.44
Protocol: TCP
Version: 4
Header Length (32 bit words): 5
Precedence: Routine
        Normal Delay, Normal Throughput, Normal Reliability
Total length: 40
Identification: 21533
Fragmentation allowed, Last fragment
Fragment Offset: 0
Time to Live: 64 seconds
Checksum: 0x7197(Valid)

tcp: ===== Transmission Control Protocol =====
Source Port: TELNET
Destination Port: 6295
Sequence Number: 1508170215
Acknowledgement Number: 25784420
Data Offset (32-bit words): 5
Window: 30719
Control Bits: Acknowledgement Field is Valid (ACK)
               No More Data from Sender (FIN)
Checksum: 0xC450(Valid)
Urgent Pointer: 0

```



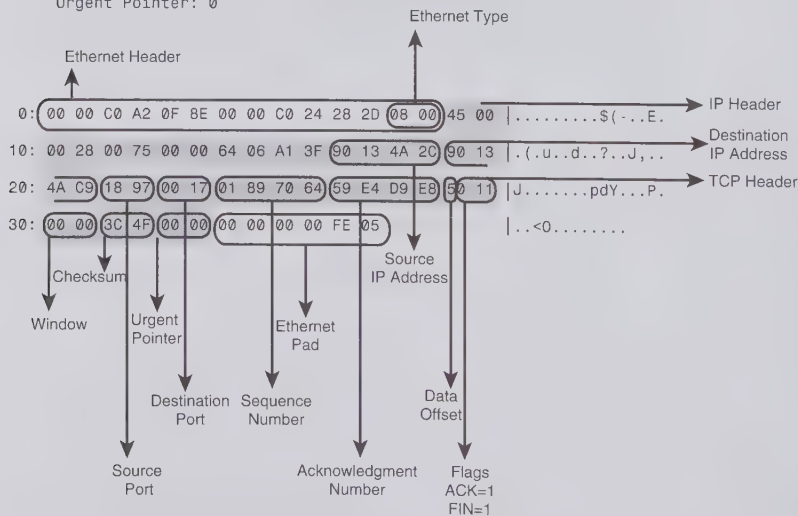
Dans le paquet 244 (Figure 11.34) le client Telnet affecte la valeur 1 au flag FIN, ce qui indique qu'il ne dispose plus d'aucune donnée et qu'il est prêt à refermer la connexion. Une remarque : dans cette implémentation la taille de la fenêtre vaut zéro pour indiquer à TCP que ni le client ni le serveur n'accepteront plus aucune donnée.

Figure 11.34

La requête de fermeture est acceptée par le client Telnet.

```

Packet Number : 244          6:39:12 PM
Length : 64 bytes
ether: ===== Ethernet Datalink Layer =====
      Station: 00-00-C0-24-28-2D ----> 00-00-C0-A2-0F-8E
      Type: 0x0800 (IP)
ip: ===== Internet Protocol =====
      Station: 144.19.74.44 ----> 144.19.74.201
      Protocol: TCP
      Version: 4
      Header Length (32 bit words): 5
      Precedence: Routine
          Normal Delay, Normal Throughput, Normal Reliability
      Total length: 40
      Identification: 117
      Fragmentation allowed, Last fragment
      Fragment Offset: 0
      Time to Live: 100 seconds
      Checksum: 0xA13F(Valid)
tcp: ===== Transmission Control Protocol =====
      Source Port: 6295
      Destination Port: TELNET
      Sequence Number: 25784420
      Acknowledgement Number: 1508170216
      Data Offset (32-bit words): 5
      Window: 0
      Control Bits: Acknowledgement Field is Valid (ACK)
                   No More Data from Sender (FIN)
      Checksum: 0x3C4F(Valid)
      Urgent Pointer: 0
  
```



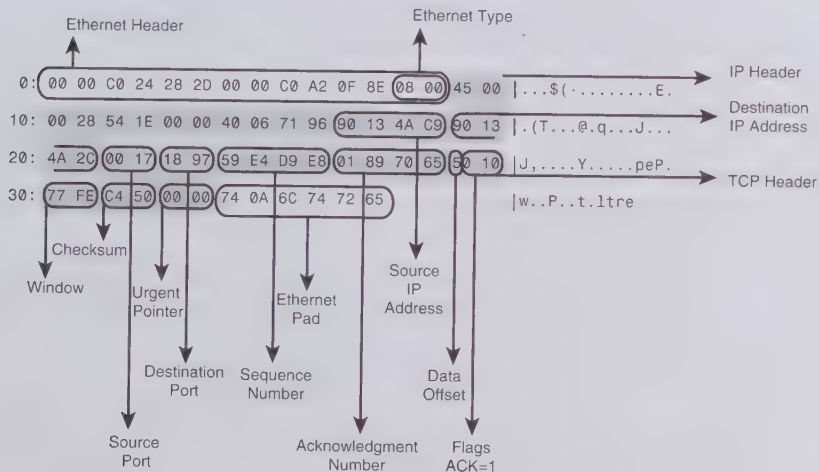
Dans la Figure 11.35 le serveur Telnet acquitte la réception du flag FIN provenant du client Telnet. A ce moment là, les modules TCP qui se trouvent sur le client et sur le serveur libèrent les ressources associées à la connexion TCP.

Figure 11.35

Acquittement de la requête de fermeture en provenance du client Telnet.

```

Packet Number : 245          6:39:12 PM
Length : 64 bytes
ether: ===== Ethernet Datalink Layer =====
      Station: 00-00-C0-A2-0F-8E -> 00-00-C0-24-28-2D
      Type: 0x0800 (IP)
ip: ===== Internet Protocol =====
     Station: 144.19.74.201 -> 144.19.74.44
     Protocol: TCP
     Version: 4
     Header Length (32 bit words): 5
     Precedence: Routine
           Normal Delay, Normal Throughput, Normal Reliability
     Total length: 40
     Identification: 21534
     Fragmentation allowed, Last fragment
     Fragment Offset: 0
     Time to Live: 64 seconds
     Checksum: 0x7196(Valid)
tcp: ===== Transmission Control Protocol =====
     Source Port: TELNET
     Destination Port: 6295
     Sequence Number: 1508170216
     Acknowledgement Number: 25784421
     Data Offset (32-bit words): 5
     Window: 30718
     Control Bits: Acknowledgement Field is Valid (ACK)
     Checksum: 0xC450(Valid)
     Urgent Pointer: 0
  
```



User Datagram Protocol

Il existe un certain nombre d'applications qui n'ont pas besoin de la robustesse du protocole TCP. Le protocole UDP (*User Datagram Protocol* ou "Protocole de datagramme utilisateur") permet d'identifier les applications sur les machines et fournit une vérification rudimentaire pour les erreurs.

Alors que TCP est orienté connexion, le protocole UDP opère au niveau des datagrammes. UDP ne peut pas établir de connexion. Pour envoyer des données par UDP, on les encapsule dans un datagramme IP, avec un en-tête UDP. La couche IP envoie le paquet UDP en un seul datagramme, sauf si on requiert une fragmentation. UDP ne séquence pas les données et elles peuvent parfois arriver en ordre dispersé. Les applications qui auraient besoin de données séquencées devront soit fournir elles-mêmes un mécanisme de séquençement, soit utiliser le protocole TCP. Cependant, dans un environnement de réseau local (LAN) la probabilité que les données n'arrivent pas dans l'ordre est faible (les délais sont courts et prévisibles, la topologie du réseau très simple).

UDP est utile pour des applications orientées commande/réponse dont les commandes et les réponses peuvent tenir dans un seul datagramme. Ce protocole ne présente pas l'inconvénient de devoir ouvrir puis refermer une connexion, ce qui s'avère particulièrement coûteux lorsqu'il s'agit de petites quantités de données. UDP est aussi utile pour des applications qui requièrent une diffusion ou une multidiffusion. Avec TCP, si on envoie un paquet en diffusion à 1 000 stations l'émetteur devra ouvrir 1 000 connexions, leur envoyer individuellement les données et refermer 1 000 connexions. Un tel processus est très lourd et utilise des ressources (pour le maintien des connexions). En comparaison, UDP permet à l'émetteur de se contenter d'envoyer les données au module IP en requérant une distribution en diffusion/multidiffusion. Ce sont les capacités sous-jacentes du réseau qui se chargeront d'envoyer les données en diffusion/multidiffusion.

UDP fournit en outre un contrôle d'erreurs qui assure aux données une intégrité rudimentaire. Si le réseau sous-jacent est fiable on pourra désactiver le total de contrôle d'UDP sans préjudice et accélérer d'autant le traitement UDP.

Voici le résumé des caractéristiques du protocole UDP :

- UDP possède un mécanisme permettant d'identifier les processus d'application à l'aide de numéros de port UDP.
- UDP est orienté datagrammes, ce qui évite les problèmes liés à l'ouverture, au maintien et à la fermeture des connexions.

- UDP est efficace pour les applications en diffusion/multidiffusion.
- UDP ne séquence pas les données. La remise conforme des données n'est pas garantie.
- UDP peut éventuellement vérifier l'intégrité des données (et des données seules) avec un total de contrôle.
- UDP est plus rapide, plus simple et plus efficace que TCP, mais il est moins robuste.

Le protocole UDP permet une transmission sans connexion, mais aussi sans sécurité. On peut considérer que c'est un protocole de couche de transport. Mais c'est un protocole de couche de transport qui n'assure pas l'intégrité des données transmises. Malgré cet inconvénient, de nombreuses applications reposent sur UDP :

- TFTP (*Trivial File Transfer Protocol*) ;
- DNS (*Domain Name System*) ;
- NFS (*Network File System*) version 2 ;
- SNMP (*Simple Network Management Protocol*) ;
- RIP (*Routing Information Protocol*) ;
- de nombreux services qui envoient des données en diffusion, par exemple WHOD (*Who Daemon pour serveurs Unix*).

Les sections qui suivent détaillent le protocole UDP.

• Le format de l'en-tête UDP

L'en-tête UDP a une taille fixe de huit octets. La Figure 11.36 présente le format de cet en-tête.

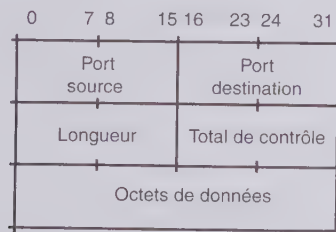


Figure 11.36

Le format de l'en-tête UDP.



Le champ Source Port (port d'origine) occupe 16 bits. Il indique en principe :

- le numéro de port du processus émetteur ;
- le numéro de port où on peut adresser les réponses lorsque l'on ne dispose d'aucun autre renseignement ;
- si sa valeur est 0, le champ indique qu'aucun numéro de port n'est attribué.

Le champ Destination Port (port de destination) identifie le processus correspondant à l'adresse IP de destination et auquel on envoie les données UDP. Comme TCP, UDP effectue le démultiplexage des données à l'aide de numéros de port. Lorsque le protocole UDP reçoit un datagramme sans numéro de port (c'est-à-dire lorsqu'aucune application UDP n'est associée au port indiqué), il génère un message d'erreur ICMP indiquant qu'il est impossible de contacter le port et il rejette le datagramme.

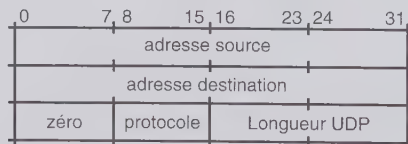
Le champ Length contient la longueur du paquet UDP en octets. Cette valeur tient compte de l'en-tête UDP et des données. La valeur minimale est 8 et correspond à un paquet où le champ de données est vide.

Le champ de total de contrôle occupe 16 bits. C'est le complément à 1 de la somme des compléments à 1 du pseudo-en-tête extrait de l'en-tête IP, de l'en-tête UDP et des données, avec un remplissage de zéros de manière à obtenir une longueur qui soit un multiple de deux octets. UDP utilise la même procédure de total de contrôle que TCP.

Le pseudo en-tête en préfixe de l'en-tête UDP contient l'adresse d'origine, l'adresse de destination, le protocole (à UDP correspond le numéro 17) et la longueur UDP (Figure 11.37). Ces informations sont destinées à prévenir les erreurs de routage. La taille du pseudo en-tête est de 12 octets et contient toutes les spécifications associées à la destination (même si aucune connexion n'est établie).

Figure 11.37

Le pseudo en-tête UDP utilisé dans le calcul du total de contrôle.



Si le total de contrôle vaut zéro on le transmet sous une forme ne contenant que des 1 (arithmétique en complément à 1). Par conséquent, avec ce mode de calcul on n'aura jamais un total de contrôle dont tous les bits valent zéro. Un total de contrôle dont tous les bits valent zéro indiquera en fait que l'on n'utilise pas le total de contrôle. On peut considérer les choses sous un autre angle : 0 possède deux représentations en arithmétiques en complément à 1, c'est-à-dire soit avec

tous les bits à 0, soit avec tous les bits à 1. Les 1 servent pour le total de contrôle calculé, les zéro indiquent que l'on désactive le total de contrôle. On pourra désactiver le total de contrôle pour une application s'exécutant sur un réseau IP fiable (par exemple un réseau local) et s'épargner ainsi la vérification des datagrammes UDP.

Le protocole UDP s'exécute au-dessus d'IP. Le total de contrôle IP ne concerne que l'en-tête alors que le total de contrôle UDP vérifie aussi les données. Donc, pour assurer l'intégrité des données, il faudra activer le total de contrôle UDP.

• Position et encapsulation des couches UDP

La Figure 11.38 montre qu'UDP s'exécute au-dessus d'IP. On associe un numéro de port aux applications qui s'exécutent sur UDP. Ce numéro de port possède un espace d'adressage distinct de celui du numéro de port TCP. On pourra donc avoir un numéro de port TCP égal à 1017 qui renvoie à une application et un numéro de port UDP égal à 1017 renvoyant à une autre application. Le Tableau 2.5 (voir Chapitre 2) présentait les numéros de port TCP et UDP.

Figure 11.38

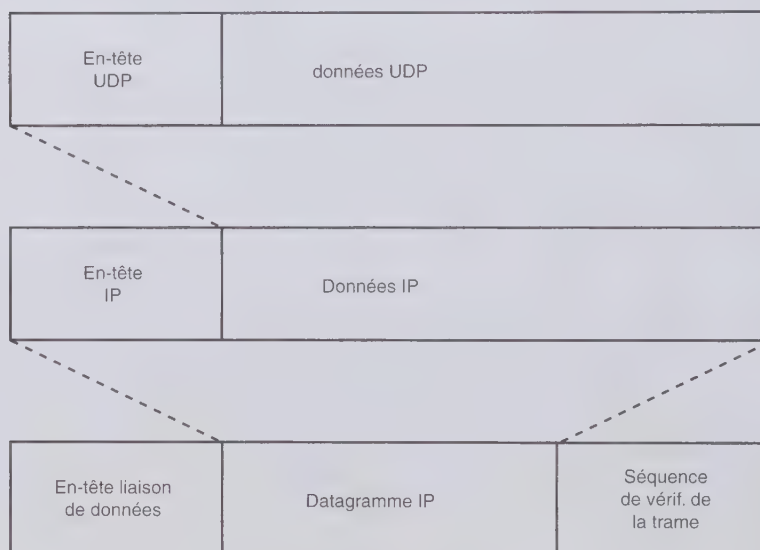
La position des couches UDP.



La Figure 11.39 présente l'encapsulation des données UDP dans un datagramme IP, lequel est à son tour encapsulé dans une trame de la couche de liaison de données. Le numéro de protocole correspondant à UDP est 17.

Figure 11.39

L'encapsulation des couches UDP.



La couche UDP ayant besoin de connaître l'adresse IP dans le pseudo en-tête, les deux protocoles sont très liés. Le pseudo en-tête participe au total de contrôle UDP. Le module UDP doit pouvoir déterminer les adresses IP d'origine et de destination ainsi que le champ Protocol de l'en-tête IP. Une interface UDP/IP doit pouvoir retourner le datagramme complet (y compris l'en-tête IP) en réponse à une opération RECEIVE émise par l'application UDP. L'interface autorisera UDP à passer un datagramme IP complet (avec l'en-tête) à la couche IP qui l'enverra. La couche IP pourra vérifier certains champs et calculer le total de contrôle.

• La trace UDP

Les Figures 11.40 et 11.41 présentent chacun un paquet UDP : le premier est généré par WHOD, l'autre par le protocole RIP.

Les numéros de port d'origine et de destination du paquet UDP engendré par WHOD valent 513. Les numéros de port d'origine et de destination du paquet UDP engendré par RIP valent 520. Dans les deux cas, le total de contrôle présente une valeur non nulle (il est donc activé).

Figure 11.40

Un paquet UDP
engendré
par WHOD.

```

Packet Number : 2                               10:57:22 PM
Length : 130 bytes
ether: ===== Ethernet Datalink Layer =====
      Station: 00-00-C0-BB-E8-73 ---> FF-FF-FF-FF-FF-FF
      Type: 0x0800 (IP)
ip: ===== Internet Protocol =====
      Station: 199.245.180.15 ---> 199.245.180.255
      Protocol: UDP
      Version: 4
      Header Length (32 bit words): 5
      Precedence: Routine
            Normal Delay, Normal Throughput, Normal Reliability
      Total length: 112
      Identification: 296
      Fragmentation allowed, Last fragment
      Fragment Offset: 0
      Time to Live: 64 seconds
      Checksum: 0x805B(Valid)
udp: ===== User Datagram Protocol =====
      Source Port: WHO
      Destination Port: WHO
      Length = 92
      Checksum: 0x086A(Valid)

Data:
0: 01 01 00 00 31 EE A4 F5 00 00 00 00 67 61 74 65 |.....1.....gate
10: 6B 65 65 70 65 72 00 00 00 00 00 00 00 00 00 00 |keeper.....
20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
30: 00 00 00 00 00 00 00 00 31 EE 7F 85 63 6F 6E 73 |.....1.....cons
40: 6F 6C 65 00 72 6F 6F 74 00 00 00 00 31 EE 80 10 |ole.root...1...
50: 00 00 24 36                                     |$.6

```

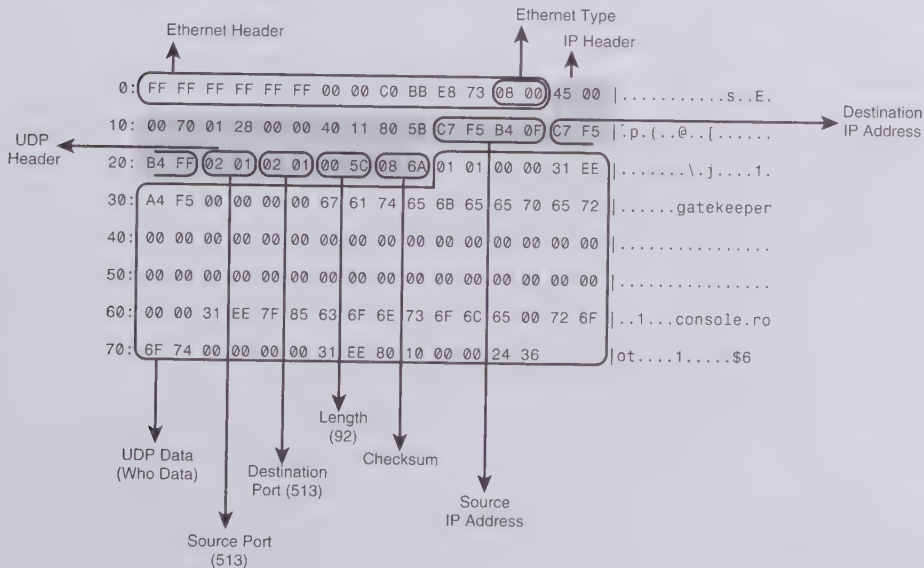


Figure 11.41

Un paquet UDP engendré
par RIP.

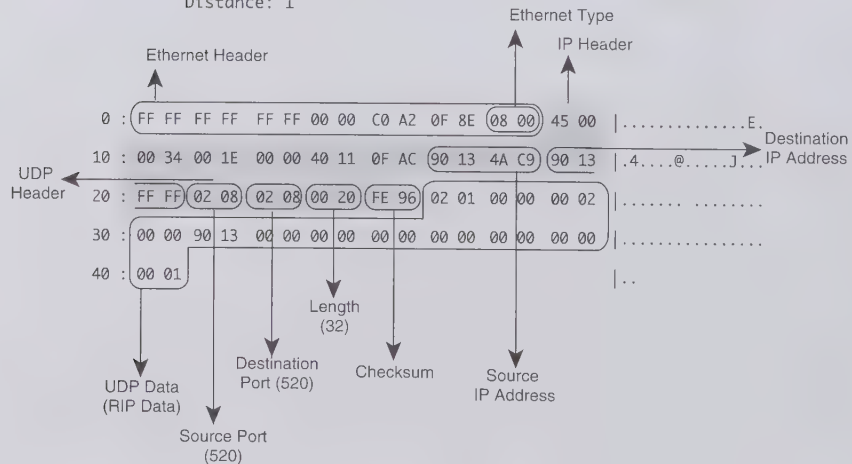
```

Packet Number : 4          10:57:57 PM
Length : 70 bytes
ether: ===== Ethernet Datalink Layer =====
      Station: 00-00-C0-A2-0F-8E ----> FF-FF-FF-FF-FF-FF
      Type: 0x0800 (IP)

ip: ===== Internet Protocol =====
      Station: 144.19.74.201 ----> 144.19.255.255
      Protocol: UDP
      Version: 4
      Header Length (32 bit words): 5
      Precedence: Routine
              Normal Delay, Normal Throughput, Normal Reliability
      Total length: 52
      Identification: 30
      Fragmentation allowed, Last fragment
      Fragment Offset: 0
      Time to Live: 64 seconds
      Checksum: 0x0FAC(Valid)

udp: ===== User Datagram Protocol =====
      Source Port: ROUTER
      Destination Port: ROUTER
      Length = 32
      Checksum: 0xFE96(Valid)

rip: ===== Routing Information Protocol =====
      Command: Response
      Version: 1
      Family ID: IP
          IP Address: 144.19.0.0
          Distance: 1
  
```



Remarquez que le champ Protocol a pour valeur 17, indiquant l'encapsulation d'UDP.

Récapitulatif

TCP et UDP sont des protocoles de couche de transport standard implémentés pour les réseaux IP. Tous les périphériques qui ne sont pas des routeurs et implémentent la suite de protocole TCP/IP servent aussi à implémenter TCP. Les routeurs proprement dits n'implémentent ni TCP ni UDP. Mais la plupart des routeurs du commerce offrent une gestion de réseau et une capacité de logage à distance, ce qui implique l'utilisation des protocoles de transport TCP et UDP. Dans de nombreuses implémentations TCP, UDP fait partie du module TCP (alors que c'est un concept tout à fait différent).

A l'aide de TCP et UDP vous pouvez remettre les données à une machine distante, mais aussi directement à un processus en cours d'exécution sur cette machine. On identifie les processus par des numéros de port. La remise des données TCP est fiable, car elle est orientée connexion. Cependant UDP est utile pour de nombreuses applications, par exemple lorsque l'on doit envoyer les données à une application précise sur la même machine ou si l'on veut envoyer des données en diffusion ou en multidiffusion.

Testez vos connaissances

1. Quelles sont les caractéristiques de TCP et d'UDP qui identifient les processus d'application ?
2. Dans quel intervalle les numéros de port TCP et UDP sont-ils pris ? Les numéros de port TCP sont-ils indépendants des numéros de port UDP ? Pourquoi ?
3. Faites la liste des caractéristiques TCP et expliquez-les brièvement.
4. Enumérez les conséquences de l'attribution de numéros de séquence individuels pour chaque octet du flux de données TCP.
5. Expliquez le mécanisme PAR intervenant dans la fiabilité de TCP.
6. Comment obtient-on un contrôle de flux dans TCP ?
7. Soit le flux de données TCP suivant :
 Left edge of window = 1111
 Window size = 6,000 octets
 TCP segment sent but not acknowledged = 800 octets



Répondez aux questions suivantes :

- a) Les données des octets 900 à 1110 ont-elles été acquittées ?
 - b) Peut-on envoyer immédiatement les données qui commencent à l'octet 7111 ?
 - c) Recensez les numéros de séquence des données pouvant être envoyées immédiatement à partir du numéro 1911.
8. Supposons, toujours pour le flux de données de la question 7, que l'on reçoive les paramètres d'acquittement suivants :
- ACK flag = 1
- Acknowledgment Number = 2111
- Window size = 3,000 octets
- Quelle sera la taille de la fenêtre et l'extrémité gauche de la fenêtre au niveau de l'émetteur ? Faites un schéma de cette fenêtre.
9. Comment réalise-t-on le multiplexage/démultiplexage des données dans TCP ?
10. Qu'est-ce qu'une extrémité de connexion TCP ? Soit la connexion TCP définie par les paramètres suivants pour l'émetteur et le récepteur :
- Sender IP = 200.13.44.2
- Sender TCP port number = 2012
- Destination IP = 132.44.45.66
- Destination TCP port number = 23
- Enumérez les extrémités de connexion. La connexion devra-t-elle traverser une frontière de routeur ?
11. Qu'est-ce qu'une demi-association TCP ? Soit la connexion TCP définie par les paramètres suivants :
- Sender IP = 145.34.46.62
- Sender TCP port number = 3000
- Destination IP = 145.34.45.61
- Destination TCP port number = 21
- Enumérez les demi-associations de cette connexion. Devra-t-elle traverser une frontière de routeur ? Dans quels cas devra-t-elle le faire ?
12. Qu'est-ce qu'une association complète TCP ? Soit la connexion TCP définie par les paramètres suivants :
- Enumérez les associations complètes de cette connexion. La connexion devra-t-elle traverser une frontière de routeur ? Dans quels cas devra-t-elle le faire ?

13. Qu'est-ce qu'un tampon TCB ?
14. Décrivez les deux types d'appels TCP OPEN.
15. Quelle est la différence entre un numéro de port usuel et un numéro de port réservé ?
16. Comment associe-t-on un numéro de port au processus d'une application ?
17. Quel est le rôle des champs Sequence Number ("numéro de séquence") et Acknowledgment Number ("numéro d'acquittement") de TCP ? Comment peut-on savoir si ces champs contiennent des informations valides ?
18. Quand le champ Acknowledgment Number ("numéro d'acquittement") est-il invalide ?
19. Est-ce que le fait de recevoir un acquittement signifie que le processus distant a reçu les données ? Développez.
20. Quel est le rôle du champ Data Offset de l'en-tête TCP ? Quelle valeur type peut-il contenir ?
21. Soit un en-tête TCP dont le champ Data Offset vaut 6. Quelle conclusion peut-on en tirer ? Pouvez-vous déterminer d'autres renseignements concernant TCP à partir de l'en-tête TCP ?
22. Décrivez le rôle du flag TCP PSH.
23. Décrivez le rôle du flag TCP RST.
24. Décrivez le rôle du flag TCP URG. Quel autre champ lui est apparenté ?
25. Décrivez comment les flags SYN et ACK peuvent indiquer les divers types de message TCP.
26. Qu'est-ce que la négociation en trois temps de TCP ? Expliquez-en les trois phases.
27. Soit l'ouverture d'une connexion TCP entre une couche dont l'ISS vaut 345435 et une autre dont l'IRS vaut 675444. Donnez la valeur des flags SYN et ACK, le numéro de séquence et le numéro d'acquittement des messages TCP lors de la négociation en trois temps, sachant qu'aucune donnée n'est transférée.
28. Soit l'ouverture d'une connexion TCP entre une couche dont l'ISS vaut 67200 et une autre dont l'IRS vaut 893001. Sachant que le message OPEN d'ouverture active contient 17 octets de données, donnez la valeur des flags SYN et ACK, le numéro de séquence et le numéro d'acquittement des messages échangés lors de la négociation à trois temps.



29. Soit l'ouverture d'une connexion TCP entre une couche dont l'ISN vaut 345001 et une autre dont l'ISN vaut 234010. Sachant que le message OPEN d'ouverture active contient 17 octets de données, le message OPEN d'ouverture passive 20 et l'acquittement de l'IRS 12, donnez la valeur des flags SYN et ACK, le numéro de séquence et le numéro d'acquittement des messages échangés lors de la négociation à trois temps.
30. A quoi correspond la valeur MSL de TCP ?
31. Une implémentation TCP pouvant garder en mémoire les numéros de séquence en cours d'utilisation, faut-il attendre toute la durée du MSL avant de pouvoir commencer à utiliser des numéros de séquence plus importants ? Développez.
32. Si une couche TCP génère des messages à 1,44 Mbit/s combien de temps faut-il avant de réutiliser le même numéro de séquence ?
33. Comment sélectionne-t-on l'ISN pour TCP ?
34. Que se passerait-il au démarrage si TCP commençait toujours avec un ISN de 0 ? Avec un ISN de 1 ?
35. Peut-on résoudre le problème en se servant de la date et de l'heure courante pour déterminer l'ISN ?
(Un indice : considérez ce qui se passe en cas de défaillance.)
36. Quel est le rôle du flag FIN ? Décrivez le mécanisme de fermeture de TCP.
37. Comment calcule-t-on le total de contrôle ? Est-ce que ce total s'applique aussi aux données ? Quel est le rôle joué par le pseudo en-tête dans le calcul du total de contrôle ?
38. Quel est le rôle de l'option MSS de TCP ? Quand peut-on l'envoyer ?
39. Quels sont les avantages et désavantages du caractère cumulatif des acquittements TCP ?
40. Pourquoi est-il nécessaire d'utiliser des temporisateurs adaptatifs avec TCP ?
41. Donnez les formules de calcul du temporisateur de retransmission. Si $\alpha = 0,8$, $\beta = 2$, $LBOUND = 1\text{ ms}$ et $UBOUND = 1\text{ s}$, calculez les valeurs du temporisateur de retransmission correspondant aux mesures suivantes du RTT :
- RTT = 0.21
- RTT = 0.18
- RTT = 0.34
- RTT = 0.52
- RTT = 0.81

42. Comment l'algorithme de Karn opère-t-il pour accroître l'efficacité des temporisateurs de retransmission ?
43. Décrivez l'algorithme de contrôle de congestion de Van Jacobson.
44. Qu'est-ce que le syndrome SWS ?
45. Comment évite-t-on le SWS côté émetteur ?
46. Comment évite-t-on le SWS côté récepteur ?
47. Expliquez pourquoi la technique d'évitement de Nagle convient aux émetteurs qu'ils envoient les données par petites quantités ou par grandes quantités.
48. Comment TCP gère-t-il les ruptures de connexion ?
49. Enumérez les caractéristiques principales d'UDP.
50. Quand faut-il utiliser UDP plutôt que TCP ?
51. Quand faut-il utiliser TCP plutôt que UDP ?
52. Que signifie la valeur 0 comme numéro de port de l'émetteur UDP ?
53. Analysez le protocole de transport à partir du dump hexadécimal ci-après.
La couche de liaison de données est une couche Ethernet.

```

0: 00 00 C0 24 28 2D 00 00 C0 A2 0F 8E 08 00 45 00 |...$(-.....E.
10: 00 29 54 0C 40 00 40 06 31 A7 90 13 4A C9 90 13 |..)T.@.1...J...
20: 4A 2C 00 17 18 97 59 E4 D9 D2 01 89 70 5B 50 38 |J,...Y....p[PB
30: 77 FF C5 44 00 01 FF 20 30 20 20 20 20 20 20 20 |w..D... 0

```

54. Analysez le protocole de transport à partir du dump hexadécimal ci-après.
La couche de liaison de données est une couche Ethernet.

```

0: FF FF FF FF FF FF 00 00 C0 BB E8 73 08 00 45 00 |.....s..E.
10: 00 70 01 2A 00 00 40 11 80 59 C7 F5 B4 0F C7 F5 |..)p*..@..Y.....
20: B4 FF 02 01 02 01 00 5C 07 02 01 01 00 00 31 EE |.....\.....1.
30: A5 A9 00 00 00 00 67 61 74 65 6B 65 65 70 65 72 |.....gatekeeper
40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
60: 00 00 31 EE 7F 85 63 6F 6E 73 6F 6C 65 00 72 6F |..1...console.ro
70: 6F 74 00 00 00 00 31 EE 80 10 00 00 24 EA |ot...1....$.

```



LES SERVICES TCP/IP

12	<i>Configuration automatique</i>	511
13	<i>Services d'application</i>	545
14	<i>Gestion de réseau TCP/IP</i>	623



12 Configuration automatique

Lorsque le réseau est important, il est très fastidieux de configurer les paramètres TCP/IP pour chaque station de travail. C'est également vrai lorsqu'il s'agit de modifier simplement quelques paramètres TCP/IP (les adresses IP ou les masques de sous-réseau, par exemple). Cette situation peut se présenter à la suite d'une restructuration significative de l'ensemble du réseau, ou simplement parce que les utilisateurs sont nombreux et utilisent souvent des ordinateurs portables qu'ils peuvent connecter à n'importe quel segment du réseau. De plus, les connexions physiques peuvent être tantôt des connexions directes, tantôt des connexions sans câble. Il faut donc pouvoir adapter les valeurs des paramètres TCP/IP d'une machine selon les segments.

Pour mesurer les conséquences qu'entraîne la modification des paramètres TCP/IP, un administrateur réseau aura souvent besoin de toute sa compétence. Heureusement, l'IETF (*Internet Engineering Task Force*) a développé plusieurs protocoles pour configurer automatiquement les réseaux TCP/IP, en particulier BOOTP (*Boot Protocol*) et DHCP (*Dynamic Host Configuration Protocol*).

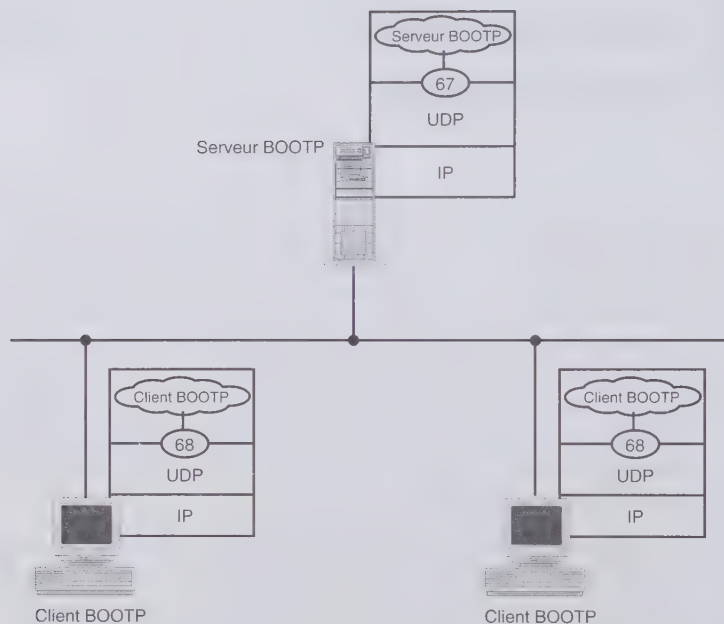
Configuration dynamique avec le protocole BOOTP

Ce protocole se sert des protocoles UDP et IP pour transporter des informations de configuration destinées aux ordinateurs qui souhaitent régler eux-mêmes les paramètres IP. Ces ordinateurs sont donc des *clients BOOTP* et la machine qui répond à la requête est un *serveur BOOTP* (voir Figure 12.1).

La requête du client BOOTP doit lui permettre d'obtenir les paramètres IP qui lui conviennent.

Le protocole BOOTP est présenté dans le document RFC 951 (*Bootstrap Protocol*), et les mises à jour RFC 2132, RFC 1532, RFC 1542 et RFC 1395. Dans les sections qui suivent, nous décrivons le fonctionnement du protocole BOOTP.



Figure 12.1*Clients et serveurs BOOTP.*

- **Les adresses IP de requêtes et de réponses dans le protocole BOOTP**

Les messages BOOTP sont encapsulés dans un datagramme UDP accompagné d'un en-tête identifiant le numéro de port utilisé par les processus BOOTP. Le datagramme UDP est à son tour encapsulé dans un datagramme IP. Lorsque le client BOOTP envoie une requête, il lui faut une adresse de destination et une adresse d'origine. Or, précisément, le client BOOTP ignore généralement quelle est son adresse IP. C'est pourquoi il utilise souvent l'adresse IP 0.0.0.0. Bien entendu, il utilise son adresse IP lorsqu'il la connaît.

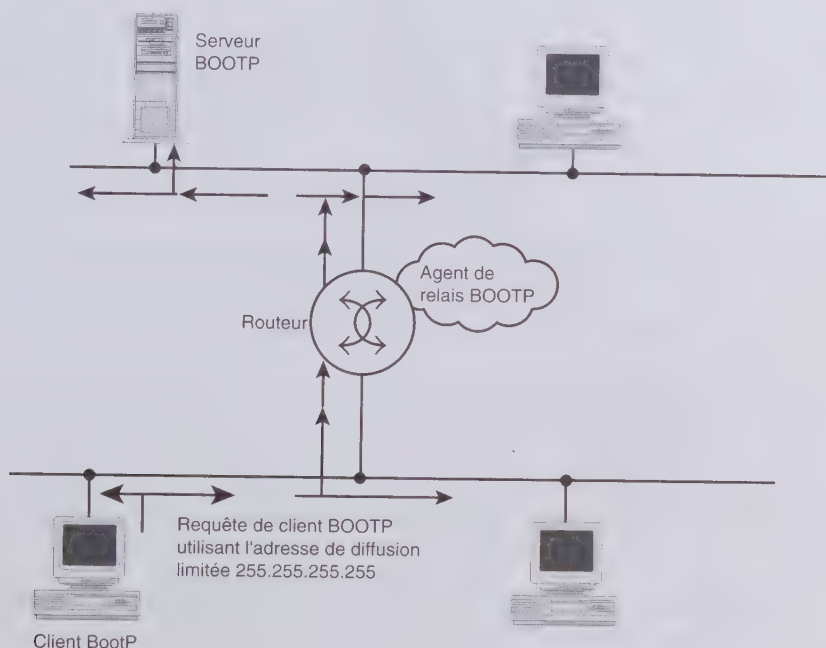
De même, le client BOOTP ignore le plus souvent l'adresse du serveur BOOTP. C'est particulièrement vrai pour les stations de travail dépourvues de disque local, qui n'ont aucun moyen de connaître l'adresse du serveur BOOTP. Mais le client se servira de l'adresse du serveur si on a pu le configurer pour cela.

Lorsque le client ne connaît pas l'adresse du serveur, il opère une diffusion IP limitée. Nous avons vu au Chapitre 4 que la diffusion IP limitée correspond à l'adresse 255.255.255.255.

Cette adresse est commune à tous les nœuds IP du réseau local, donc c'est aussi celle du serveur BOOTP.

Mais alors que se passe-t-il lorsque le serveur BOOTP appartient à un sous-réseau IP distinct, séparé du client par un routeur ? Nous avons vu au Chapitre 4 que la technique de diffusion limitée ne passait pas la frontière de routeur. Ici, il faudra configurer le routeur et y adjoindre un agent de relais BOOTP qui va vérifier en cas de diffusion limitée si le datagramme utilise le numéro de port UDP 67 (ce numéro caractérise les protocoles BOOTP et DHCP). De cette façon, le message BOOTP sera transmis aux interfaces réseau du routeur (voir Figure 12.2).

Figure 12.2
*Transmission de messages
BOOTP au-delà de la frontière
du routeur.*



Lorsque le serveur BOOTP se trouve sur le réseau local, il peut répondre à la requête du client. Selon l'implémentation du serveur, le message de réponse peut éventuellement utiliser directement l'adresse IP du client.

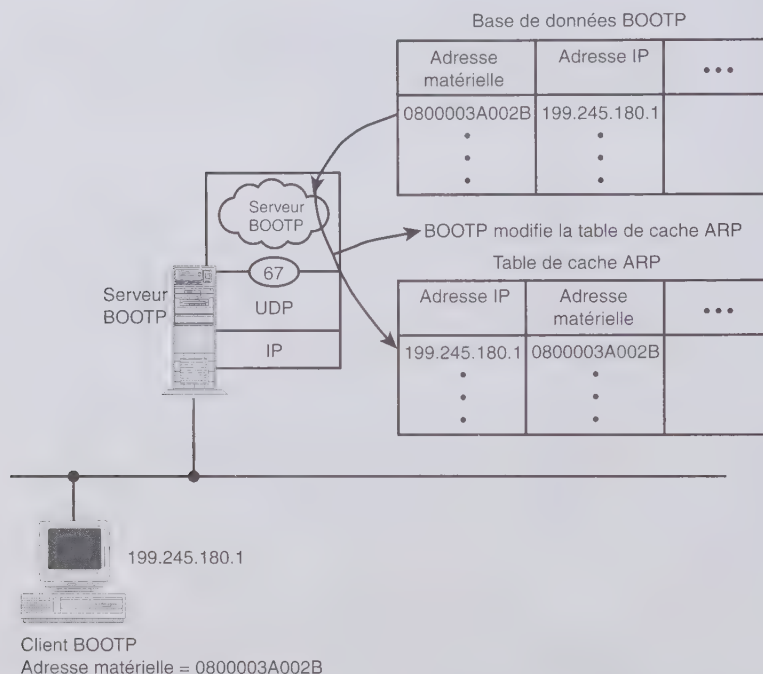
En effet, pour connaître l'adresse IP du client, le serveur consulte la base de données de configuration BOOTP. Mais le client ne connaît pas son adresse IP et ne sait pas répondre lorsque le serveur lui envoie un message de requête ARP pour obtenir son adresse matérielle. Il faudra donc implémenter une entrée spéciale dans la table de cache ARP pour que le serveur puisse envoyer sa réponse directement



à l'adresse du client BOOTP (voir Figure 12.3). L'implémentation BOOTP de l'Unix BSD en est un exemple.

Figure 12.3

Modification de la table de cache ARP par le serveur BOOTP.



Lorsque le serveur BOOTP ne modifie pas la table de cache ARP, il doit utiliser la diffusion limitée pour sa réponse.

• Gérer la perte des messages BOOTP

Les messages BOOTP utilisent les protocoles UDP et IP qui ne garantissent pas la remise des messages. Il peut donc arriver que des messages BOOTP se perdent, soient retardés, dupliqués ou finalement remis dans le désordre. Le protocole IP ne vérifie avec le total de contrôle que l'en-tête IP, pas les données. Pour détecter les erreurs dans les messages BOOTP, il faudra activer le total de contrôle du protocole UDP.

Le client BOOTP envoie ses requêtes sans les fragmenter (le flag IP DF contient la valeur 1) afin de simplifier le traitement des réponses, et aussi pour le cas où il lui manquerait la mémoire nécessaire pour réassembler les datagrammes fragmentés.

Le protocole BOOTP utilise une temporisation et une stratégie de retransmission pour gérer la perte des messages. Lorsque le client BOOTP envoie une requête, un

temporisateur se déclenche. Si la réponse ne parvient pas avant l'expiration du temporisateur, le client transmet de nouveau sa requête.

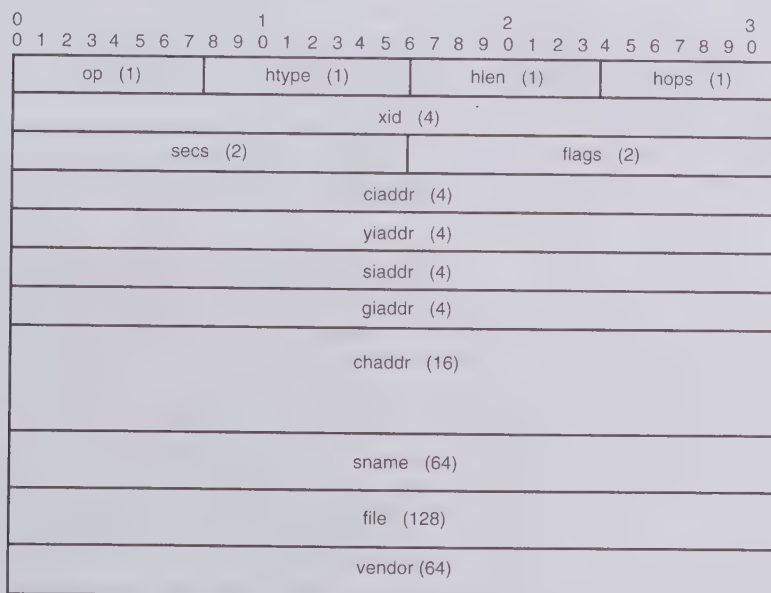
Si plusieurs clients BOOTP envoient leurs requêtes en même temps, leur diffusion risque d'inonder le réseau. C'est pourquoi la spécification BOOTP recommande l'utilisation de valeurs de temporisation aléatoires allant de zéro à quatre secondes. On double ensuite à chaque retransmission l'intervalle de valeurs que peut prendre la temporisation, jusqu'à une valeur assez importante, par exemple 60 secondes. Si la valeur est atteinte, l'intervalle aléatoire restera le même, mais la valeur de la temporisation sera toujours déterminée aléatoirement à chaque retransmission. De cette façon, on risquera moins d'observer des requêtes simultanées qui pourraient surcharger le serveur et provoquer des collisions entre paquets sur un réseau Ethernet.

• Format des messages BOOTP

La Figure 12.4 présente le format des messages BOOTP et le Tableau 12.1 donne la signification de ses différents champs.

Figure 12.4

Format des messages BOOTP.



Les nombres entre parenthèses () représentent des octets



Tableau 12.1 : Les champs des messages BOOTP

Champ	Nombre d'octets	Description
op	1	Type du message. La valeur 1 correspond à un message BOOTREQUEST, la valeur 2 à BOOTREPLY.
htype	1	Type de l'adresse matérielle. Les valeurs utilisées sont les mêmes que dans le format de paquet ARP. Par exemple, la valeur 1 correspond à une connexion Ethernet à 10 Mbit/s.
hlen	1	Longueur de l'adresse matérielle en octets. Une adresse Ethernet ou Token Ring occupe par exemple 6 octets.
hops	1	Le client BOOTP lui affecte la valeur zéro. Les éventuels agents de relais qui s'exécutent sur les routeurs s'en servent lorsqu'ils transmettent les messages BOOTP.
xid	4	ID de transaction. Il s'agit d'un numéro que le client BOOTP attribue aléatoirement lorsqu'il génère le message BOOTP. Le serveur BOOTP reprend le même identificateur dans sa réponse au client. Grâce à cela, serveur et clients peuvent associer les réponses reçues avec les messages envoyés.
secs	2	La valeur de ce champ est contrôlée par le client BOOTP. Il s'agit du nombre de secondes écoulées depuis que le client a entrepris son initialisation.
unused	2	Ce champ n'est pas utilisé.
ciaddr	4	Adresse IP du client BOOTP. Utilisée par un client BOOTP dans un message BOOTPREQUEST. Champ mis à zéro si l'adresse IP n'est pas connue.
yiaddr	4	Adresse IP du client retournée par le serveur BOOTP.
siaddr	4	Adresse IP du serveur BOOTP utilisée par le client BOOTP pour le premier contact. Cette adresse peut provenir d'une base de données de configuration TCP/IP présente sur le client BOOTP. Le serveur BOOTP retourne parfois une nouvelle adresse qui est celle du deuxième serveur intervenant dans le processus d'initialisation du client. Ce deuxième serveur peut-être la machine qui recèle l'image du système d'exploitation.
giaddr	4	Adresse IP du routeur sur lequel s'exécute l'agent de relais BOOTP.

Tableau 12.1 : Les champs des messages BOOTP (*suite*)

Champ	Nombre d'octets	Description
chaddr	16	Adresse matérielle du client BOOTP. On utilise une valeur sur 16 octets pour certains types de réseaux, mais les adresses matérielles des réseaux Ethernet ou Token Ring n'occupent que 6 octets.
sname	64	Champ facultatif contenant le nom de la machine hôte du serveur (lorsque le client BOOTP la reconnaît). La valeur est une chaîne terminée par null.
file	128	Nom du fichier de boot. C'est une chaîne terminée par null. Lorsque le client BOOTP souhaite s'initialiser avec une image de système d'exploitation, il peut spécifier un nom générique correspondant à un système d'exploitation (par exemple "unix" pour Unix). C'est dans la réponse du serveur BOOTP que l'on trouvera les informations précises, et notamment le nom de fichier et le chemin dûment qualifié menant à l'image du système d'exploitation.
vendor	64	Zone réservée à des informations concernant certains périphériques. Elle est chaque fois conforme aux exigences spécifiques du fabricant. Il peut s'agir, par exemple, dans une requête d'une valeur type/numéro du matériel, ou gestion d'un système de fichier "capacité"/distant dans une réponse. Ces informations pourront servir lors d'une troisième phase d'amorçage ou de kernel.

• Phases de la procédure BOOTP

Contrairement à ce que son nom pourrait laisser penser, le protocole BOOTP ne transfère pas du tout l'image du système d'exploitation entre le serveur et le client BOOTP. En fait, le transfert de l'image du système d'exploitation s'effectue par l'intermédiaire du protocole TFTP (*Trivial File Transfer Protocol*) qui fonctionne sur la couche de transport UDP. Ce sont les stations de travail dépourvues de disque local qui ont besoin d'une image du système d'exploitation, les autres utilisent BOOTP uniquement pour pouvoir obtenir des adresses IP.

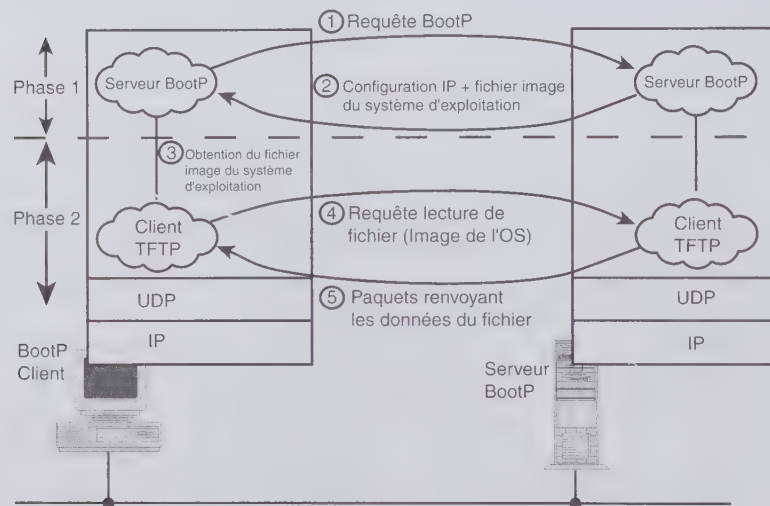
La Figure 12.5 présente les phases de toute la procédure d'amorçage nécessaire au téléchargement de l'image du système d'exploitation. A l'étape 1, le client BOOTP envoie une requête pour obtenir une image de configuration IP. A l'étape 2, le serveur



BOOTP renvoie les données de configuration ainsi que le nom du fichier contenant l'image du système d'exploitation. A l'étape 3, le client BOOTP envoie une requête pour obtenir l'image du système d'exploitation au client TFTP. A l'étape 4, le client TFTP envoie une requête au serveur TFTP pour obtenir l'image du système d'exploitation. A l'étape 5, le serveur TFTP renvoie une série de paquets de données contenant l'image du système d'exploitation. Ayant obtenu cette image, le client BOOTP charge le système d'exploitation en mémoire et s'initialise.

Figure 12.5

La procédure d'amorçage.



12ITCP05

En séparant ainsi la phase de transfert de l'image du système d'exploitation de la phase BOOTP proprement dite, on laisse la possibilité au serveur BOOTP de ne pas se trouver forcément sur la même machine que celle qui stocke les images de système d'exploitation (c'est-à-dire le serveur TFTP). En outre, on pourra encore se servir de BOOTP même si tout ce que l'on veut est l'image de configuration.

Dans certains cas, l'administrateur réseau peut souhaiter que telle ou telle station de travail ne s'initialise pas toujours avec le même système d'exploitation pour tous les utilisateurs. Le message de requête BOOTP contiendra alors un libellé générique dans le champ de nom de fichier. Par exemple, "unix" correspond au système d'exploitation Unix et "default" au système d'exploitation par défaut. Le serveur BOOTP qui lit le nom générique consulte la base des données de configuration, établit la correspondance entre le nom générique et un nom de fichier dûment qualifié et renvoie ce nom de fichier dans la réponse BOOTP. Le client BOOTP peut alors passer le nom de fichier au client TFTP local et charger le système d'exploitation correspondant.

• Le champ Vendor Specific Area

Le format des messages BOOTP décrit plusieurs paramètres IP. Mais souvent, les fabricants ont besoin de paramètres supplémentaires pour leurs périphériques. C'est le champ *Vendor Specific Area* qui contient ces paramètres.

Les données placées dans ce champ sont spécifiques de certains matériels. Elles sont destinées au client BOOTP. Le champ est facultatif et tout dépend des besoins du client BOOTP. Les quatre premiers octets contiennent une valeur définissant le format du champ. Ces quatre octets se notent à l'aide d'un système décimal à points, mais il ne faut pas les confondre avec des adresses IP. La valeur 99.130.83.99 (en hexadécimal 63.82.53.63) est une valeur arbitraire qui correspond au format standard pour ce champ. Après les quatre premiers octets, on trouve une série d'éléments :

- un champ de 1 octet pour le tag ou le type ;
- un champ facultatif de 1 octet spécifiant la longueur du champ de données qui suit ;
- un champ de données de plusieurs octets.

Le document RFC 1497 (*BOOTP Vendor Information Extensions*) définit les différents tags BOOTP. Depuis l'apparition du protocole DHCP, les extensions de données spécifiques des périphériques BOOTP et le champ d'options DHCP possèdent le même format. Ce format commun est décrit dans le document RFC 2132 (*DHCP Options and BOOTP Vendor Extensions*). Le Tableau 12.2 présente un certain nombre de tags courants.

Tableau 12.2 : Quelques tags courants pour le champ Vendor Specific Area du protocole BOOTP

Tag	Longueur	Signification
0	-	Ce tag sert uniquement pour le remplissage. Lorsqu'on l'active, les champs suivants s'aligneront sur la frontière des mots.
1	4	Champ de masque de sous-réseau.
2	4	Champ Time Offset (décalage temporel). Ce décalage en secondes est celui du sous-réseau du client par rapport à l'heure UTC. La valeur est un entier signé de 32 bits.
3	N	Option de routeur. La valeur spécifie une liste d'adresses IP correspondant à des routeurs se trouvant sur le sous-réseau du client. L'ordre des adresses dans la liste est l'ordre des préférences. La longueur minimale de ce champ est de 4 octets. La longueur du champ sera toujours un multiple de 4 octets.



Tableau 12.2 : Quelques tags courants pour le champ Vendor Specific Area du protocole BOOTP (suite)

Tag	Longueur	Signification
4	N	Option de Time Server spécifiant une liste de serveurs de synchronisation (conformes à RFC 868) possibles pour le client. L'ordre des serveurs dans la liste est l'ordre des préférences. La longueur minimale de ce champ est de 4 octets. La longueur du champ sera toujours un multiple de 4 octets.
5	N	Option de Name Server spécifiant une liste de serveurs de noms (conformes à IEN 116) possibles pour le client. L'ordre des serveurs dans la liste est l'ordre des préférences. La longueur minimale de ce champ est de 4 octets. La longueur du champ sera toujours un multiple de 4 octets.
6	N	Option de Domain Name Server spécifiant une liste de serveurs de noms de domaines (STD 13, RFC 1035) possibles pour le client. L'ordre des serveurs dans la liste est l'ordre des préférences. La longueur minimale de ce champ est de 4 octets. La longueur du champ sera toujours un multiple de 4 octets.
7	N	Option de Serveur de journal (Log Server) spécifiant une liste de serveurs de journal MIT-LCS UDP possibles pour le client. L'ordre des serveurs dans la liste est l'ordre des préférences. La longueur minimale de ce champ est de 4 octets. La longueur du champ sera toujours un multiple de 4 octets.
8	N	Option de Serveur de cookies (Cookie Server) spécifiant une liste de serveurs de cookies (conformes à RFC 865) possibles pour le client. L'ordre des serveurs dans la liste est l'ordre des préférences. La longueur minimale de ce champ est de 4 octets. La longueur du champ sera toujours un multiple de 4 octets.
9	N	Option de Serveur LPR spécifiant une liste de serveurs d'imprimantes en ligne (conformes à RFC 1179) possibles pour le client. L'ordre des serveurs dans la liste est l'ordre des préférences. La longueur minimale de ce champ est de 4 octets. La longueur du champ sera toujours un multiple de 4 octets.
10	N	Option de Serveurs Impress spécifiant une liste de serveurs d'Impress Imagen possibles pour le client. L'ordre des serveurs dans la liste est l'ordre des préférences. La longueur minimale de ce champ est de 4 octets. La longueur du champ sera toujours un multiple de 4 octets.

Tableau 12.2 : Quelques tags courants pour le champ Vendor Specific Area du protocole BOOTP (suite)

Tag	Longueur	Signification
11	N	Option de Serveurs d'emplacements de ressources (Resource Location Server) spécifiant une liste de Resource Location servers (conformes à RFC 887) possibles pour le client. L'ordre des serveurs dans la liste est l'ordre des préférences. La longueur minimale de ce champ est de 4 octets. La longueur du champ sera toujours un multiple de 4 octets.
12	N	Option de Nom d'hôte (Host Name) spécifiant le nom du client. Ce nom peut éventuellement être qualifié avec le nom de domaine local. Consultez RFC 1035 pour les restrictions concernant les jeux de caractères. La longueur minimale de ce champ est de 1 octet.
13	2	Option de Taille de fichier de boot (Boot File Size) spécifiant pour le client la longueur de l'image de boot par défaut par unités de 512 octets. Cette valeur est un entier non signé de 16 bits.
128-254	Non définie	Champ réservé aux options spécifiques relatives au site. Son interprétation dépend de l'implémentation.
255	-	Option marquant la fin des données valides du champ vendor. Les octets suivants appartenant au champ seront remplis selon l'option de remplissage (tag 0).

Configuration dynamique avec le protocole DHCP

Le protocole DHCP (*Dynamic Host Configuration Protocol*) est une extension du protocole BOOTP qui offre plus de souplesse dans la gestion des adresses IP. On peut s'en servir pour configurer de manière dynamique les paramètres TCP/IP fondamentaux des différents hôtes d'un réseau (stations de travail ou serveurs). Le protocole DHCP se compose de deux éléments :

- un mécanisme d'allocation pour les adresses IP et les autres paramètres TCP/IP ;
- un protocole pour négocier et transmettre des informations spécifiques relatives à l'hôte.

L'hôte TCP/IP qui requiert les données de configuration s'appelle un *client DHCP*, et l'hôte TCP/IP qui lui fournit ces informations s'appelle un *serveur DHCP*.



Le protocole DHCP est décrit dans le document RFC 2131. Les sections qui suivent expliquent son fonctionnement.

• Gestion des adresses IP avec le protocole DHCP

DHCP fait appel aux trois méthodes suivantes pour allouer les adresses IP :

- allocation manuelle ;
- allocation automatique ;
- allocation dynamique.

La méthode d'allocation manuelle permet à l'administrateur réseau d'attribuer lui-même au niveau du serveur DHCP une adresse IP pour le client DHCP. Le protocole DHCP pourra servir ensuite à faire parvenir cette adresse jusqu'au client.

La méthode d'allocation automatique ne demande aucune intervention manuelle. L'adresse IP du client lui sera attribuée lorsqu'il contactera pour la première fois le serveur DHCP. Les adresses attribuées avec cette méthode sont définitives et ne pourront pas être utilisées par un autre client DHCP.

La méthode d'allocation dynamique permet d'attribuer une adresse temporaire au client DHCP. Cette adresse sera une adresse d'emprunt réservée au client DHCP pour une certaine durée. A l'expiration de cette période, l'adresse IP devient caduque et le client DHCP doit la restituer. Si le client a encore besoin d'une adresse IP, il doit alors en requérir une nouvelle.

La méthode d'allocation dynamique n'est que l'une des trois méthodes autorisant le réemploi automatique d'une adresse IP. A l'expiration du délai, le client DHCP n'est pas obligé de restituer l'adresse. Lorsqu'il n'en a plus besoin, il restitue l'adresse IP au serveur DHCP qui pourra la réattribuer à un nouveau client.

Cette méthode est particulièrement utile pour les clients DHCP qui ont besoin d'une adresse temporaire pour une nouvelle connexion au réseau. Considérons, par exemple, que 300 utilisateurs utilisent des portables et que l'on a affecté une adresse de classe C à un réseau. Le réseau doit pouvoir contenir 254 nœuds (256 moins 2 adresses spéciales, soit 254). Or une connexion TCP/IP suppose chaque fois une adresse IP distincte. Les 300 utilisateurs ne peuvent donc pas se connecter simultanément à ce réseau. Mais, même en considérant qu'il existe 200 connexions physiques, on peut réutiliser les adresses qui ne sont pas en cours d'utilisation grâce à l'allocation dynamique de DHCP.

Mais l'allocation dynamique d'adresses IP est aussi très utile pour lorsque l'on ne dispose pas d'un nombre suffisant d'adresses en réserve. Ainsi, on pourra attribuer une adresse à un nouvel hôte qui doit rester connecté de façon permanente et réutiliser l'adresse dès que l'hôte sortira du réseau.

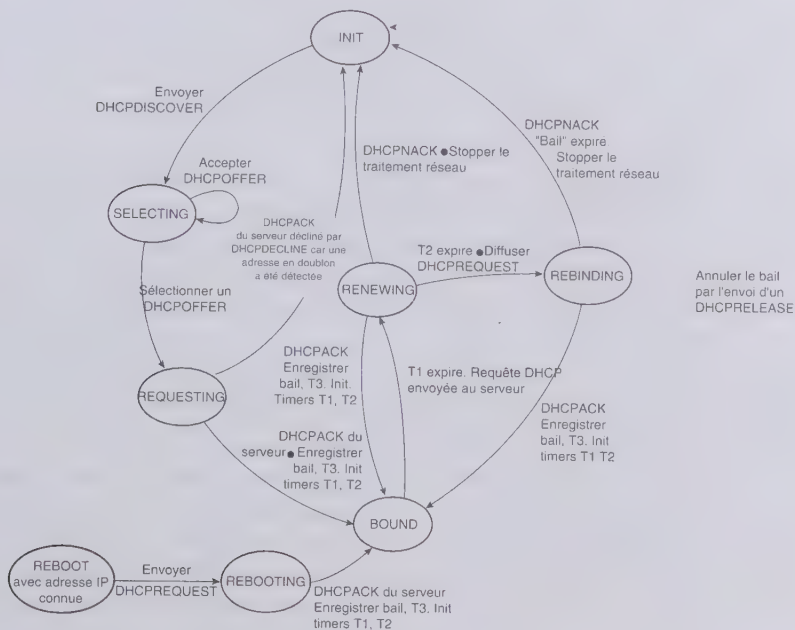
Quelle que soit la méthode utilisée vous n'avez pas à configurer chaque machine, il suffit de configurer une fois les paramètres TCP/IP sur le serveur DHCP.

• Processus d'acquisition des adresses IP dans le protocole DHCP

A partir du moment où le client DHCP a contacté le serveur DHCP, il passe par toute une série d'états internes au cours desquels il négocie l'utilisation de l'adresse IP et sa durée. Le meilleur moyen d'en rendre compte est encore de se rapporter au diagramme de transition d'état (machine à états finis) de la Figure 12.6.

Figure 12.6

Diagramme de transition d'état montrant les interactions entre le client et le serveur DHCP.



Au lancement du client DHCP, celui-ci est dans l'état INIT (*initialize*). Pour l'instant, il ne connaît pas ses paramètres IP, il envoie donc la diffusion DHCPDISCOVER. Celle-ci est encapsulée dans un paquet UDP/IP. Le numéro de port UDP pour la destination reçoit la valeur décimale 67. C'est la même que pour un serveur BOOTP, car DHCP est une extension du protocole BOOTP. L'adresse de



diffusion IP sera 255.255.255.255. Si le serveur DHCP ne se trouve pas sur le réseau local, il faudra que le routeur dispose d'un agent de relais DHCP capable de transmettre la requête DHCPDISCOVER à d'autres sous-réseaux. La prise en charge des agents de relais DHCP est expliquée dans le document RFC 1542.

Avant d'envoyer le paquet de diffusion DHCPDISCOVER, le client marque une pause d'une durée aléatoire pouvant aller de 1 à 10 secondes. Cela permet d'éviter que tous les clients DHCP n'envoient leurs requêtes en même temps au moment où on rétablit l'alimentation.

Après l'envoi de la diffusion DHCPDISCOVER, le client DHCP passe à l'état SELECTING. Dans cet état, il attend les messages DHCPOFFER en provenance des serveurs DHCP configurés pour lui répondre. Cette attente se prolonge aussi longtemps que l'implémentation l'aura spécifiée. Le client DHCP devra parfois sélectionner une réponse DHCPOFFER parmi plusieurs. Ce choix effectué, il enverra un message DHCPREQUEST au serveur correspondant. Le serveur répondra à son tour par un message DHCPACK.

Eventuellement, le client DHCP pourra vérifier que l'adresse IP que lui envoie le serveur n'est pas en cours d'utilisation. Sur un réseau de diffusion, il enverra une requête ARP à l'adresse indiquée et attendra une réponse ARP. S'il en reçoit une c'est que l'adresse est en cours d'utilisation : le client ignorera l'adresse se trouvant dans le message DHCPACK, enverra un message DHCPDECLINE et repassera ARP sur le réseau local. Le client spécifie bien son adresse matérielle dans le champ prévu pour l'adresse matériel de l'expéditeur dans le paquet ARP, mais il spécifie 0 pour l'adresse IP de l'expéditeur. Ainsi, on ne risque pas de troubler les caches ARP de l'hôte TCP/IP avec une adresse déjà en cours d'utilisation.

Si le client accepte le message DHCPACK il initialise trois temporisateurs et passe à l'état BOUND. Le premier temporisateur, T1, correspond au délai de renouvellement des adresses temporaires. Le deuxième temporisateur, T2, correspond au rebinding. Enfin, le troisième temporisateur, T3, détermine la durée de validité des adresses temporaires. La valeur de T3 est fixée automatiquement et se retrouve toujours dans le paquet DHCPACK. Mais vous pouvez configurer les valeurs de T2 et de T1 au niveau du serveur DHCP. Si vous ne le faites pas, on leur attribuera des valeurs par défaut calculées en fonction de T3. Voici le mode de calcul des valeurs par défaut :

T1 = temporisateur de renouvellement

T2 = temporisateur de rebinding

T3 = durée de validité

$T1 = 0.5 \times T3$

$T2 = 0.875 \times T3$

La date effective d'expiration se calcule en ajoutant la valeur des temporisateurs et la date d'envoi de la requête DHCPREQUEST à l'origine de la réponse

DHCPACK. Si on note T_0 cette date d'envoi, le calcul des dates d'expiration se fait comme suit :

Expiration de $T_1 = T_0 + T_1$

Expiration de $T_2 = T_0 + T_2$

Expiration de $T_3 = T_0 + T_3$

La documentation RFC 2131 recommande l'ajout d'un facteur aléatoire aux temporisateurs T_1 et T_2 afin d'empêcher que tous les clients DHCP ne voient expirer leurs temporisateurs au même moment.

A l'expiration de T_1 , le client DHCP passe de l'état BOUND à l'état RENEWING. Dans cet état, il lui faut négocier le prolongement de l'attribution de son adresse IP avec le serveur DHCP qui lui avait alloué cette attribution. Si le serveur refuse, il le signifie par un message DHCPNAK et le client repasse à l'état INIT pour tenter d'obtenir une nouvelle adresse IP. Lorsque le client reçoit un message DHCPACK, celui-ci contient la durée d'expiration de la nouvelle attribution. Alors, le client réinitialise les temporisateurs et passe à l'état BOUND.

Si le temporisateur T_2 arrive à expiration alors que le client DHCP attend toujours dans l'état RENEWING un message DHCPACK ou DHCPNAK en provenance du serveur, le client DHCP passe à l'état REBINDING. En effet, il est possible que l'absence de réponse du serveur soit due à un dysfonctionnement du serveur ou de la liaison entre le client et le serveur. Remarquez bien que $T_2 > T_1$: l'attente du renouvellement dure en fait $T_2 - T_1$.

A l'expiration du temporisateur T_2 , le client dans l'état REBINDING diffuse sur le réseau une requête DHCPREQUEST à destination d'un serveur DHCP, toujours pour obtenir la prolongation de l'attribution de son adresse IP. En effet, le client considère après $T_2 - T_1$ secondes que le serveur DHCP responsable de la première attribution étant indisponible il lui faut contacter un nouveau serveur. S'il y parvient et reçoit un message DHCPACK, le client réinitialise ses temporisateurs et repasse à l'état BOUND. Mais s'il n'arrive à contacter aucun serveur avant expiration du temporisateur T_3 , son adresse IP arrive aussi à expiration et le client repasse à l'état INIT.

A l'expiration du temporisateur T_3 , le client DHCP doit restituer l'adresse IP et stopper tout traitement en cours qui impliquerait cette adresse.

Le client DHCP peut aussi restituer l'adresse IP avant expiration du temporisateur T_3 . Prenons l'exemple d'un ordinateur portable ayant reçu l'autorisation d'utiliser une adresse IP pendant une heure. Au bout d'une demi-heure, l'utilisateur ayant terminé son travail décide de couper la connexion. Le client DHCP envoie alors un message DHCPRELEASE au serveur pour libérer l'adresse. L'adresse IP sera aussitôt disponible pour un autre client DHCP.



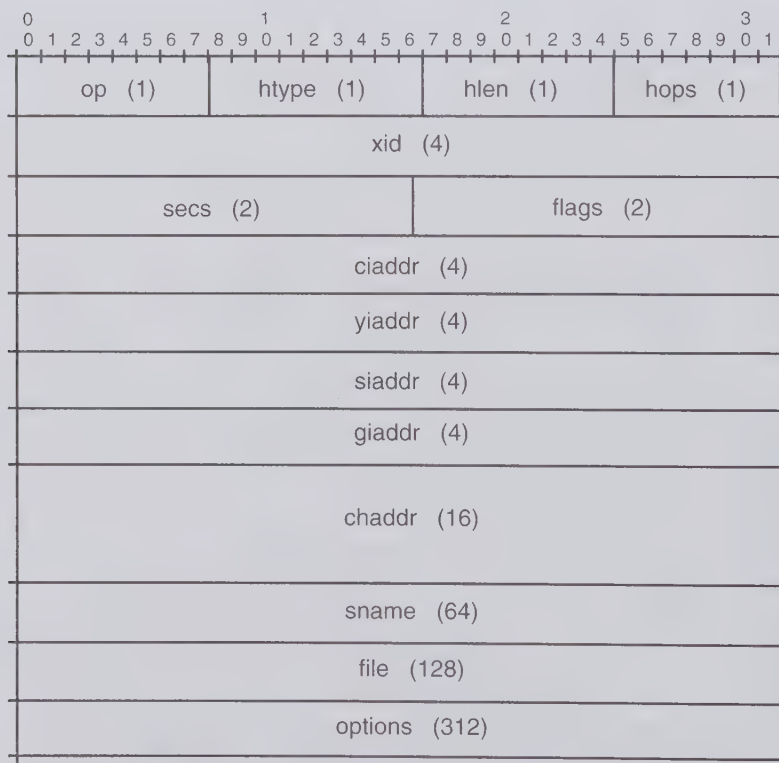
Lorsque les clients DHCP se trouvent sur des machines pourvues d'un disque, on peut y stocker l'adresse. A la prochaine initialisation l'ordinateur pourra requérir la même adresse. Ce processus est illustré à la Figure 12.6, pour l'état étant intitulé "REBOOT avec adresse IP connue".

• Format des paquets DHCP

La Figure 12.7 présente le format des paquets DHCP. Les messages DHCP utilisent un format fixe pour tous les champs, à l'exception du champ options, lequel occupe au minimum 312 octets. Le lecteur attentif constatera que ce format est identique à celui des messages BOOTP (sauf le champ flags et le champ options). En fait, on peut même configurer le serveur DHCP pour répondre à des requêtes BOOTP. Les détails de configuration sont spécifiques du système d'exploitation.

Figure 12.7

Le format des paquets DHCP.



Les nombres entre parenthèses représentent le nombre d'octets.

Le Tableau 12.3 explique les champs qui interviennent dans le protocole DHCP. On n'utilise que le bit de poids fort du champ options (voir Figure 12.8), les autres devant avoir zéro pour valeur.

Figure 12.8
Le champ options
du protocole DHCP.



B = 1 Broadcast

B = 0 Unicast

Tableau 12.3 : Les champs du protocole DHCP

Champ	Octets	Description
op	1	Type du message. La valeur 1 correspond à un message BOOTREQUEST, la valeur 2 à BOOTREPLY.
htype	1	Type de l'adresse matérielle. Les valeurs utilisées sont les mêmes que dans le format de paquet ARP. Par exemple, la valeur 1 correspond à une connexion Ethernet à 10 Mbit/s.
hlen	1	Longueur de l'adresse matérielle en octets. Une adresse Ethernet ou Token Ring occupe, par exemple, 6 octets.
hops	1	Le client DHCP lui affecte la valeur zéro. Les éventuels agents de relais qui s'exécutent sur les routeurs s'en servent lorsqu'ils transmettent les messages DHCP.
xid	4	ID de transaction. Il s'agit d'un numéro que le client DHCP attribue aléatoirement lorsqu'il génère le message DHCP. Le serveur DHCP reprend le même identificateur dans sa réponse au client. Grâce à cela, serveur et clients peuvent associer les réponses reçues avec les messages envoyés.
flags	2	Ce flag permet de spécifier un message de diffusion en affectant la valeur 1 à son bit de poids fort, les autres bits restant à zéro.



Tableau 12.3 : Les champs du protocole DHCP (*suite*)

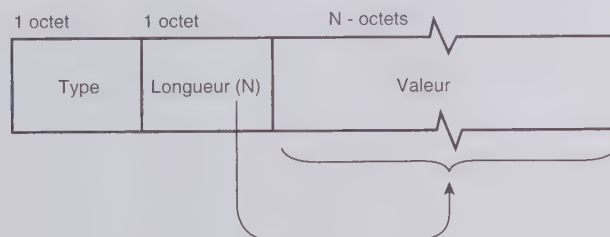
Champ	Octets	Description
secs	2	La valeur de ce champ est contrôlée par le client DHCP. Il s'agit du nombre de secondes écoulées depuis que le client a entrepris son initialisation.
unused	2	Ce champ n'est pas utilisé.
ciaddr	4	Adresse IP du client DHCP. Utilisée par le client DHCP dans un message DHCPREQUEST. Champ mis à zéro si l'adresse IP n'est pas connue.
yiaddr	4	Adresse IP du client retournée par le serveur DHCP.
siaddr	4	Adresse IP du serveur DHCP spécifique que le client DHCP souhaite contacter. Cette adresse peut provenir de la suite de messages DHCPOFFER et DHCPACK renvoyés par le serveur. Il peut s'agir d'un serveur secondaire participant au processus d'initialisation (par exemple, le serveur contenant l'image du système d'exploitation).
giaddr	4	Adresse IP du routeur sur lequel s'exécute l'agent de relais DHCP.
chaddr	16	Adresse matérielle du client DHCP. On utilise une valeur sur 16 octets pour certains types de réseaux, mais les adresses matérielles des réseaux Ethernet ou Token Ring n'occupent que 6 octets.
sname	64	Champ facultatif contenant le nom de la machine hôte du serveur (lorsque le client DHCP la reconnaît). La valeur est une chaîne terminée par null.
file	128	Nom du fichier de boot. C'est une chaîne terminée par null. Lorsque le client DHCP souhaite s'initialiser avec une image de système d'exploitation, il peut spécifier un nom générique correspondant à un système d'exploitation (par exemple "unix" pour Unix) dans le message DHCPDISCOVER. Le serveur DHCP détient les informations précises concernant l'image de système d'exploitation requise. Le serveur retourne le chemin dûment qualifié et le nom du fichier dans le message DHCPOFFER.
options	312	Champ contenant des paramètres optionnels.

La plupart des messages DHCP envoyés par le serveur sont des message unicast (ils sont envoyés à une seule adresse). En effet, l'existence du client DHCP n'est révélée au serveur DHCP que par l'adresse matérielle du serveur dans les premiers messages qu'il lui adresse. Cependant, le client peut requérir que le serveur DHCP lui réponde avec une adresse de diffusion. Pour cela, il affecte la valeur 1 au bit de poids fort du champ options. Il le fait notamment lorsqu'il ignore encore quelle est sa propre adresse IP. Le module du protocole IP du client DHCP rejettera tout datagramme dont l'adresse de destination ne correspond pas à l'adresse IP de son interface réseau. Ainsi, le datagramme sera aussi rejeté lorsque le client ignore encore sa propre adresse. Mais un client DHCP acceptera toujours un datagramme IP en diffusion. C'est pourquoi il requiert du serveur DHCP de lui répondre en diffusion plutôt qu'en unicast.

La longueur du champ options est variable, mais vaut au minimum 312 octets, ce qui porte la taille minimale d'un message DHCP à 576 octets, c'est-à-dire la taille minimale des datagrammes IP qu'un hôte peut accepter. Lorsque le client doit utiliser des messages DHCP plus grands, il faut qu'il le négocie avec l'option Taille de message DHCP maximale. Etant donné que les champs sname et file sont relativement importants et pas toujours utiles, on peut les incorporer au champ options en spécifiant l'option Option Overload. Dans ce cas, les champs sname et file seront simplement ignorés et deviendront des options. On exprime toujours les options au format TLV (type-longueur-valeur).

La Figure 12.9 présente une option dans le protocole DHCP. Le premier octet correspond au champ Type (le type), et l'octet suivant au champ Length (la longueur). La valeur de Length est la longueur en octets du champ Value (la valeur). Pour exprimer un message DHCP on utilise la valeur 53 pour le champ Type. La Figure 12.10 présente les valeurs d'options permettant de décrire les messages DHCP.

Figure 12.9
*Le format des options
dans les messages DHCP.*



• La trace du protocole DHCP

Dans cette section, nous présentons le format des messages DHCP correspondant à des paquets que l'on capture sur un réseau. Etant donné que le protocole DHCP est une extension de BOOTP, il ne sera question ici que des traces DHCP. En réalité, clients et serveurs DHCP utilisent les mêmes numéros de port que les clients et les serveurs BOOTP. Le client DHCP utilise le port UDP 68 et le serveur utilise le port UDP 67. La plupart des analyseurs de protocole tenteront d'interpréter les messages DHCP et BOOTP avec le même format.

Figure 12.10

Valeurs d'options pour les messages DHCP.

	1 octet	1 octet	1 octet
DHCPDISCOVER	53 Type	Longueur 1	Valeur 1
DHCPOFFER	53 Type	Longueur 1	Valeur 2
DHCPREQUEST	53 Type	Longueur 1	Valeur 3
DHCPDECLINE	53 Type	Longueur 1	Valeur 4
DHCKACK	53 Type	Longueur 1	Valeur 5
DHCPNAK	53 Type	Longueur 1	Valeur 6
DHCPRELEASE	53 Type	Longueur 1	Valeur 7

Trace DHCP pour une réponse dirigée

Les Figures 12.11 et 12.12 présentent une requête en diffusion émise par le client DHCP et la réponse du serveur DHCP. La réponse est dirigée, car l'implémentation du serveur DHCP avait introduit directement une entrée pour le client dans la table de cache ARP.

Figure 12.11

*Une requête DHCP
en diffusion.*

```

Packet Number : 4          7:28:11 PM
Length : 346 bytes
ether: ===== Ethernet Datalink Layer =====
Station: 00-A0-24-AB-D1-E6 ----> FF-FF-FF-FF-FF-FF
Type: 0x0800 (IP)
ip: ===== Internet Protocol =====
Station: 0.0.0.0 ----> 255.255.255.255
Protocol: UDP
Version: 4
Header Length (32 bit words): 5
Precedence: Routine
          Normal Delay, Normal Throughput, Normal Reliability
Total length: 328
Identification: 0
Fragmentation allowed, Last fragment
Fragment Offset: 0
Time to Live: 128 seconds
Checksum: 0x39A6(Valid)
udp: ===== User Datagram Protocol =====
Source Port: 800TPC
Destination Port: 800TPS
Length = 308
Checksum: 0xA382(Valid)

Data:
0: 01 01 06 00 82 6E 77 69 00 00 00 00 00 00 00 00 .....nwi.....
10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....A0 24 AB
20: D1 E6 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 63 82 53 63 .....C.Sc
F0: 35 01 01 3D 07 01 00 A0 24 AB D1 E6 0C 06 54 50 5...$....TP
100: 4E 54 57 00 37 07 01 0F 03 2C 2E 2F 06 FF 00 00 NTW.7.../...
110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
120: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

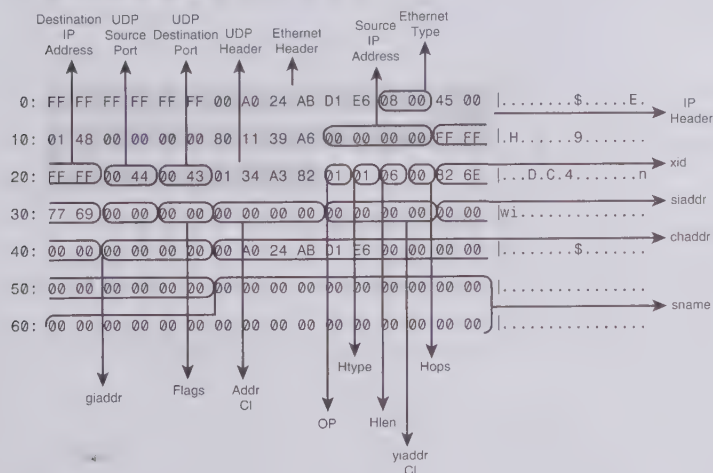
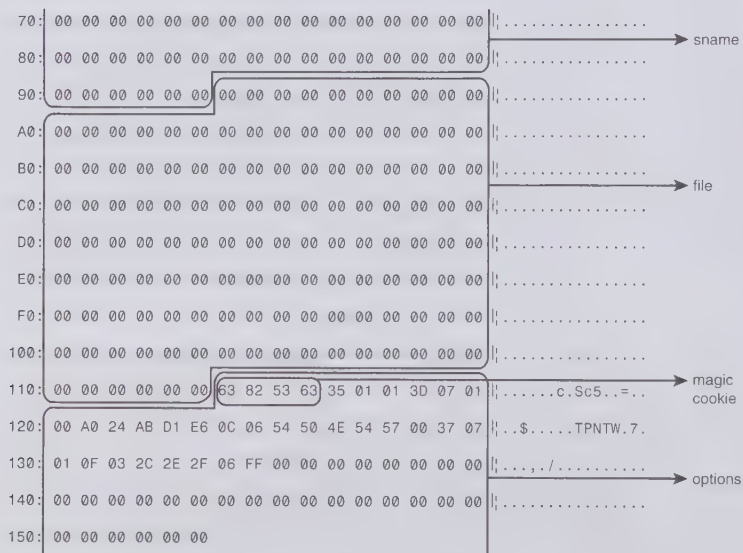


Figure 12.11

(suite)



Voici les paramètres de la requête en diffusion :

```

op = 1
htype = 1
hlen = 6
hops = 0
xid = 826E7769 (hexadécimal)
secs = 0
flags = 0
ciaddr = 0.0.0.0
yiaddr = 0.0.0.0
siaddr = 0.0.0.0
giaddr = 0.0.0.0
chaddr = 00A024ABD1E6 000000000000000000000000
sname = 0 (64 octets)
file = 0 (128 octets)
options = 0 (312 octets)

```


Figure 12.12*La réponse DHCP dirigée.*

```

Packet Number : 5          7:28:11 PM
Length : 594 bytes
ether: ===== Ethernet Datalink Layer =====
      Station: 00-00-C0-DD-14-5C ---> 00-A0-24-AB-D1-E6
      Type: 0x0800 (IP)
ip: ===== Internet Protocol =====
      Station: 199.245.180.10 ---> 199.245.180.41
      Protocol: UDP
      Version: 4
      Header Length (32 bit words): 5
      Precedence: Routine
          Normal Delay, Normal Throughput, Normal Reliability
      Total length: 576
      Identification: 47444
      Fragmentation allowed, Last fragment
      Fragment Offset: 0
      Time to Live: 128 seconds
      Checksum: 0x8739(Valid)
udp: ===== User Datagram Protocol =====
      Source Port: BOOTPS
      Destination Port: BOOTPC
      Length = 556
      Checksum: 0x0000(checksum not used)

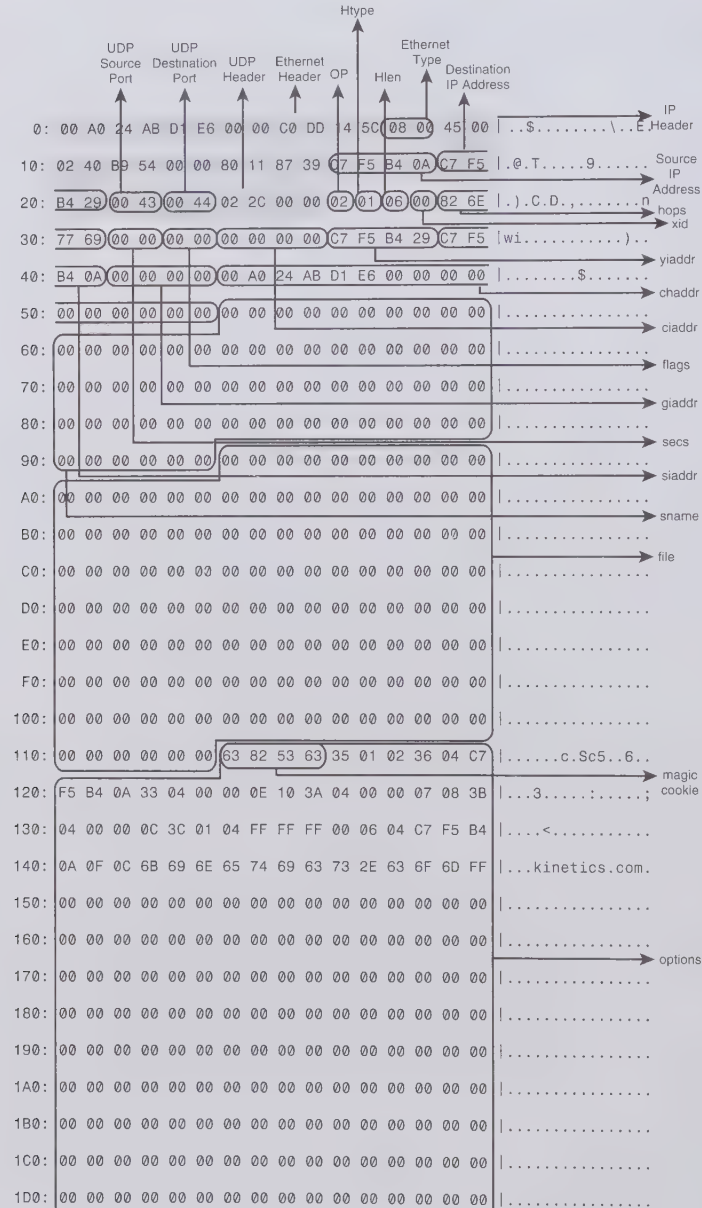
Data:
  0: 02 01 06 00 82 6E 77 69 00 00 00 00 00 00 00 00 .....nwl.....
 10: C7 F5 B4 29 C7 F5 B4 0A 00 00 00 00 00 00 00 00 A0 24 AB ...).$..
 20: D1 E6 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
 30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
 40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
 50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
 60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
 70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
 80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
 90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
 A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
 B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
 C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
 D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
 E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 63 82 53 63 .....c.Sc
 F0: 35 01 02 36 04 C7 F5 B4 0A 33 04 00 00 0E 10 3A 3A S..6.....3....:
100: 04 00 00 07 08 3B 04 00 00 0C 3C 01 04 FF FF FF .....j...<.....
110: 00 06 04 C7 F5 B4 0A 0F 0C 6B 69 6E 65 74 69 63 .....kinetic
120: 73 2E 63 6F 6D FF 00 00 00 00 00 00 00 00 00 00 s.com.....
130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
140: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
1A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
1B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
1C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
1D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
1E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
1F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
200: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
210: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
220: 00 00 00 00

```



Figure 12.12

(suite)



1E0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
1F0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
200:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
210:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
220:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
230:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
240:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	

→ options

Voici les paramètres de la réponse dirigée :

```

op = 2
htype = 1
hlen = 6
hops = 0
xid = 826E7769 (hexadécimal)
secs = 0
flags = 0
ciaddr = 0.0.0.0
yiaddr = 199.245.180.41
siaddr = 199.245.180.10
giaddr = 0.0.0.0
chaddr = 00A024ABD1E6 000000000000000000000000
sname = 0 (64 octets)
file = 0 (128 octets)
options: magic cookie = 63 82 53 63 (hex)

```

Trace DHCP pour une réponse en diffusion

Les Figures 12.13 et 12.14 présentent une requête en diffusion émise par le client DHCP et la réponse du serveur DHCP. Cette fois, la réponse est en diffusion. Ici, l'implémentation du serveur DHCP ne crée pas directement une entrée dans la table de cache ARP.



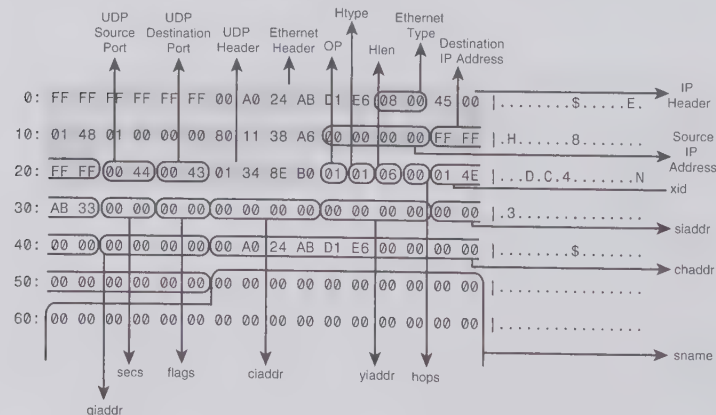
Figure 12.13

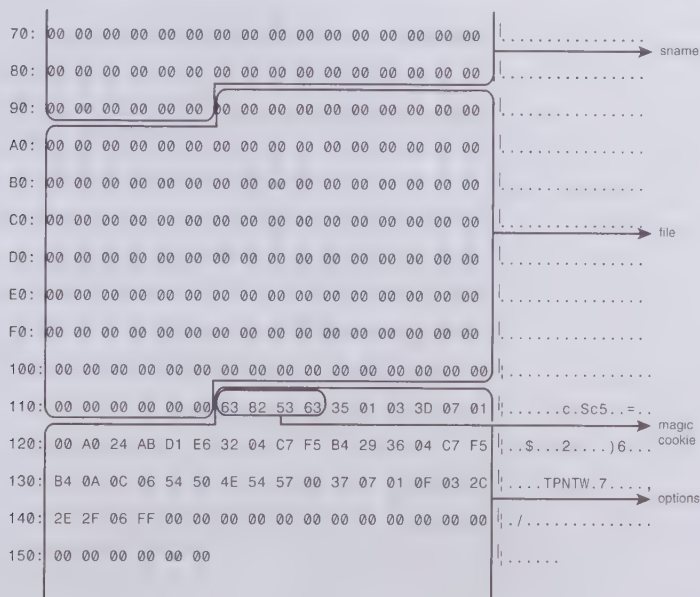
Une requête DHCP
en diffusion.

```

Packet Number : 6           7:28:11 PM
Length : 346 bytes
ether: ===== Ethernet Datalink Layer =====
      Station: 00-A0-24-AB-D1-E6 ----> FF-FF-FF-FF-FF-FF
      Type: 0x0800 (IP)
ip: ===== Internet Protocol =====
      Station: 0.0.0.0 ----> 255.255.255.255
      Protocol: UDP
      Version: 4
      Header Length (32 bit words): 5
      Precedence: Routine
          Normal Delay, Normal Throughput, Normal Reliability
      Total length: 328
      Identification: 256
      Fragmentation allowed, Last fragment
      Fragment Offset: 0
      Time to Live: 128 seconds
      Checksum: 0x38A6(Valid)
udp: ===== User Datagram Protocol =====
      Source Port: 800TPC
      Destination Port: 800TPS
      Length = 308
      Checksum: 0x8EB0(Valid)
Data:
0: 01 01 06 00 01 4E AB 33 00 00 00 00 00 00 00 00 | .....N.3.....
10: 00 00 00 00 00 00 00 00 00 00 00 00 A0 24 AB | .....$.
20: D1 E6 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
E0: 00 00 00 00 00 00 00 00 00 00 00 00 63 82 53 | .....c.Sc
F0: 35 01 03 3D 07 01 00 A0 24 AB D1 E6 32 04 C7 F5 | 5..=...$.2...
100: B4 29 36 04 C7 F5 B4 0A 0C 06 54 50 4E 54 57 00 | .)6.....TPNTW.
110: 37 07 01 0F 03 2C 2E 2F 06 FF 00 00 00 00 00 00 | 7...../.....
120: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....

```





Voici les paramètres de la requête en diffusion :

```

op = 1
htype = 1
hlen = 6
hops = 0
xid = 826E7769 (hexadécimal)
secs = 0
flags = 0
ciaddr = 0.0.0.0
yiaddr = 0.0.0.0
siaddr = 0.0.0.0
giaddr = 0.0.0.0
chaddr = 00A024ABD1E6 000000000000000000000000
sname = 0 (64 octets)
file = 0 (128 octets)
options = 0 (312 octets)

```



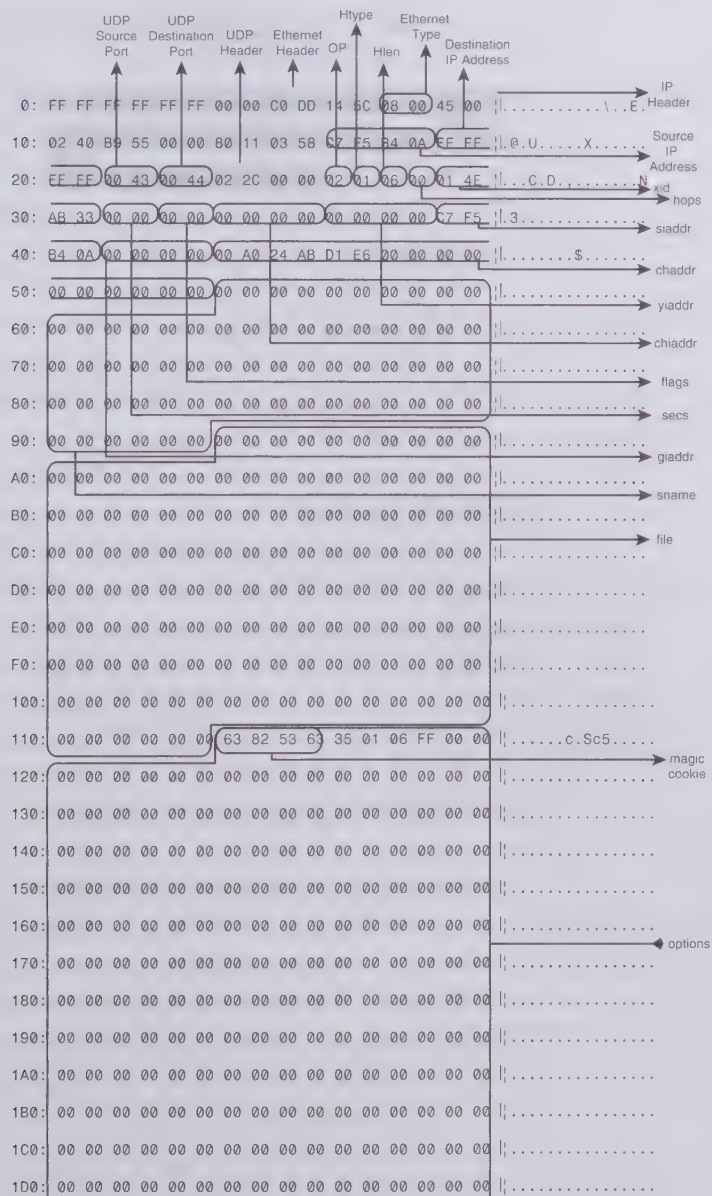


Figure 12.14
(suite)

1E0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
1F0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
200:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
210:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	→ options
220:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
230:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
240:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	

Voici les paramètres de la réponse en diffusion :

```

op = 2
htype = 1
hlen = 6
hops = 0
xid = 826E7769 (hexadécimal)
secs = 0
flags = 0
ciaddr = 0.0.0.0
yiaddr = 199.245.180.41
siaddr = 199.245.180.10
giaddr = 0.0.0.0
chaddr = 00A024ABD1E6 000000000000000000000000
sname = 0 (64 octets)
file = 0 (128 octets)
options: magic cookie = 63 82 53 63 (hexadécimal)

```

Récapitulatif

Les protocoles BOOTP et DHCP permettent de configurer automatiquement sur les différents périphériques du réseau les paramètres IP tels que les adresses ou les masques de sous-réseau. Ces deux protocoles sont construits sur le modèle client/serveur. Ils utilisent les mêmes numéros de port UDP. Le protocole DHCP est en fait une extension de BOOTP destinée à le remplacer.

L'avantage de DHCP est de permettre l'attribution d'adresses IP pour une certaine durée au bout de laquelle elles sont facilement réutilisables. Une adresse n'est plus bloquée et associée à l'adresse matérielle comme c'était le cas lorsqu'on allouait une adresse permanente à une station de travail. Retenez que l'allocation d'adresse de BOOTP est moins souple que celle de DHCP puisque les adresses IP y sont associées aux adresses matérielles de l'interface réseau.

Testez vos connaissances

1. En général pourquoi envoie-t-on une requête BOOTP en diffusion ? Quelle adresse de destination utilise-t-on alors ?
2. Quelle valeur est affectée à l'adresse d'origine de la requête BOOTP ? Pourquoi ?
3. Comment transmet-on une requête BOOTP en diffusion lorsque le serveur BOOTP se trouve sur un sous-réseau IP distinct ?
4. Pourquoi la plupart des serveurs BOOTP n'envoient-ils pas des réponses dirigées à leurs clients ?
5. Que doit pouvoir faire le serveur BOOTP pour envoyer une réponse dirigée ?
6. S'il ne peut pas envoyer une réponse dirigée, comment le serveur répond-il au client BOOTP ?
7. Comment le protocole BOOTP se protège-t-il contre d'éventuelles pertes de messages ?
8. Expliquez ce qui pourrait se passer si les clients BOOTP se lançaient tous en même temps.
9. Grâce à quel mécanisme peut-on empêcher la surcharge du réseau par des requêtes BOOTP simultanées ?
10. Faites le diagramme du format des messages BOOTP et expliquez brièvement la signification des différents champs.
11. Se sert-on de BOOTP pour télécharger l'image du système d'exploitation ?
12. Pourquoi distingue-t-on plusieurs phases dans la procédure d'amorçage d'une station utilisant BOOTP ? Décrivez ces phases.
13. Pourquoi utilise-t-on TFTP plutôt que FTP avec BOOTP ?
14. Comment peut-on configurer BOOTP de manière à autoriser le téléchargement de différentes images de système d'exploitation pour une même station de travail ?
15. Quel est le rôle du champ vendor-specific (zone spécifique constructeur) dans le protocole BOOTP ? Comment ce champ s'appelle-t-il dans le protocole DHCP ? Son format est-il identique en DHCP ?
16. Quelle valeur particulière trouve-t-on dans le champ option/vendor pour les protocoles DHCP et BOOTP ? Quel est le format de ce champ lorsque l'on trouve cette valeur ?

* 25 *



17. Quel est pour le protocole BOOTP ou DHCP le rôle des éléments option/vendor dont les valeurs de tag sont prises entre 0 et 255 ?
18. Quelles valeurs de tag sont réservées aux options spécifiques relatives au site dans les protocoles BOOTP et DHCP ?
19. Quels sont les deux éléments primaires du protocole DHCP ?
20. Citez les trois méthodes d'allocation d'adresse IP du protocole DHCP. Décrivez-les.
21. Comment le client DHCP restitue-t-il au serveur une adresse qui lui a été allouée ?
22. Soit un réseau possédant 400 connexions temporaires, mais n'utilisant jamais plus de 250 connexions simultanées. Combien d'adresses de classe C sont nécessaires pour allouer les adresses IP à l'aide de BOOTP ? Justifiez votre réponse.
23. Soit un réseau possédant 400 connexions temporaires, mais n'utilisant jamais plus de 250 connexions simultanées. Combien d'adresses de classe C sont nécessaires pour allouer les adresses IP à l'aide de DHCP ? Justifiez votre réponse.
24. Si un hôte fournit des services aux utilisateurs en dehors du réseau, quelle serait la conséquence d'allouer dynamiquement des adresses IP pour cet hôte à l'aide de DHCP ?
25. Un routeur peut-il initialiser son interface réseau dynamiquement à l'aide de DHCP ?
26. Quel est la fonction du message en diffusion DHCPDISCOVER dans le protocole DHCP ?
27. Quel mécanisme empêche les clients DHCP d'envoyer les messages DHCPDISCOVER en même temps ?
28. Dans DHCP quel est le rôle du message DHCPOFFER ?
29. Dans DHCP quel est le rôle du message DHCPACK ?
30. Quelle précaution le client DHCP peut-il prendre pour s'assurer que l'adresse qu'on lui a attribué n'est pas en cours d'utilisation ?
31. Dans quel cas le client DHCP refuse-t-il l'adresse IP que lui a attribuée le serveur DHCP ? De quel message DHCP se sert-il pour cela ? Que fait ensuite le client DHCP ?
32. Expliquez le rôle des temporisateurs T1, T2 et T3 dans le protocole DHCP. Quelles sont les relations qui existent entre eux ?
33. Pourquoi suggère-t-on dans le document RFC 2131 d'introduire un facteur aléatoire dans le calcul de la valeur des temporisateurs ?

34. Qu'arrive-t-il lorsque le temporisateur T1 arrive à expiration ?
35. Qu'arrive-t-il lorsque le temporisateur T2 arrive à expiration ?
36. Qu'arrive-t-il lorsque le temporisateur T3 arrive à expiration ?
37. Tracez un diagramme du format des messages DHCP et expliquez brièvement le rôle des différents champs.
38. En quoi le format des messages DHCP diffère-t-il du format des messages BOOTP ?
39. Comment le client DHCP peut-il spécifier que le serveur lui renvoie les réponses en unicast ou en diffusion (broadcast) ? Dans quels cas le fait-il ?
40. Quelle méthode le protocole DHCP emploie-t-il pour agrandir le champ options ?
41. Analysez le message DHCP ci-après en identifiant les différents champs et leurs valeurs :

```

0: FF FF FF FF FF FF 00 A0 24 AB D1 E6 08 00 45 00 | .....$. ....E.
10: 01 48 00 00 00 00 80 11 39 A6 00 00 00 00 FF FF | .H.....9.....
20: FF FF 00 44 00 43 01 34 A3 82 01 01 06 00 82 6E | ...D.C.4.....n
30: 77 69 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | wi.....
40: 00 00 00 00 00 00 00 00 A0 24 AB D1 E6 00 00 00 00 | .....$. ....
50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
100: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
110: 00 00 00 00 00 00 63 82 53 63 35 01 01 3D 07 01 | .....C.Sc5...=.
120: 00 A0 24 AB D1 E6 0C 06 54 50 4E 54 57 00 37 07 | ..$. ....TPNTW.7.
130: 01 0F 03 2C 2E 2F 06 FF 00 00 00 00 00 00 00 00 | ...,. /.....
140: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
150: 00 00 00 00 00 00 | .....

```



42. Analysez le message DHCP ci-après en identifiant les différents champs et leurs valeurs :

```

0: 00 A0 24 AB D1 E6 00 00 C0 DD 14 5C 08 00 45 00 |..$......\..E.
10: 02 40 B9 54 00 00 80 11 87 39 C7 F5 B4 0A C7 F5 |. @.T.....9.....
20: B4 29 00 43 00 44 02 2C 00 00 02 01 06 00 82 6E |.) .C.D.,.....n
30: 77 69 00 00 00 00 00 00 00 00 00 C7 F5 B4 29 C7 F5 |wi.....).
40: B4 0A 00 00 00 00 00 00 A0 24 AB D1 E6 00 00 00 00 |.....$.
50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
100: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
110: 00 00 00 00 00 00 63 82 53 63 35 01 02 36 04 C7 |.....c.Sc5..6..
120: F5 B4 0A 33 04 00 00 0E 10 3A 04 00 00 07 08 3B |...3.....!.....;
130: 04 00 00 0C 3C 01 04 FF FF FF 00 06 04 C7 F5 B4 |....<.....
140: 0A 0F 0C 6B 69 6E 65 74 69 63 73 2E 63 6F 6D FF |...kinetics.com.
150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
1A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
1B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
1C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
1D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
1E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
1F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
200: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
210: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
220: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
230: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
240: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....

```

43. Les clients DHCP et BOOTP utilisent le port UDP 68. Pourquoi ne prennent-ils pas un numéro de port UDP au hasard ?
44. Tracez un diagramme montrant les couches et l'encapsulation pour le protocole BOOTP. Faites la même chose pour le protocole DHCP.

13 Services d'application

Sous TCP/IP, les services d'application implémentent les fonctionnalités des couches 5, 6 et 7 du modèle OSI.

Toutes les applications TCP/IP n'ont pas besoin des services des couches 5 et 6. Les services d'application concernent le formatage, l'envoi et la réception de données au niveau des applications. Ils ne concernent pas les interactions avec l'utilisateur final qui sont gérées par les applications de niveau supérieur.

Ce chapitre abordera les principaux services d'application utilisant l'architecture TCP/IP décrite dans les chapitres précédents :

- système de nom de domaine (DNS) ;
- protocoles de courrier : SMTP, POP3, IMAP4 ;
- protocoles d'accès distant : Telnet, Berkeley r* ;
- protocoles de transfert de fichiers : FTP, TFTP ;
- protocoles d'accès fichier : NFS, Web NFS ;
- protocoles d'accès Web : HTTP/HTML, Gopher.

DNS

Dans l'Internet actuel, les machines sont identifiées par le biais du mécanisme de système de nom de domaine (DNS). DNS permet d'utiliser des noms symboliques pour accéder aux hôtes. DNS est exploité indirectement par presque tous les services d'application, les utilisateurs attribuant en général des noms symboliques qui seront résolus grâce à DNS. A titre d'exemple, un utilisateur peut ouvrir une session Telnet sur un hôte distant au moyen des commandes suivantes :

```
% telnetarchie.ans.net
Trying 147.225.1.2...
Connected to nis.ans.net.
Escape character is '^]'.
AIX telnet (nis.ans.net)
IBM AIX Version 3 for RISC System/6000
(c) Copyrights by IBM and by Others 1982, 1990
login:
```

• 38 •



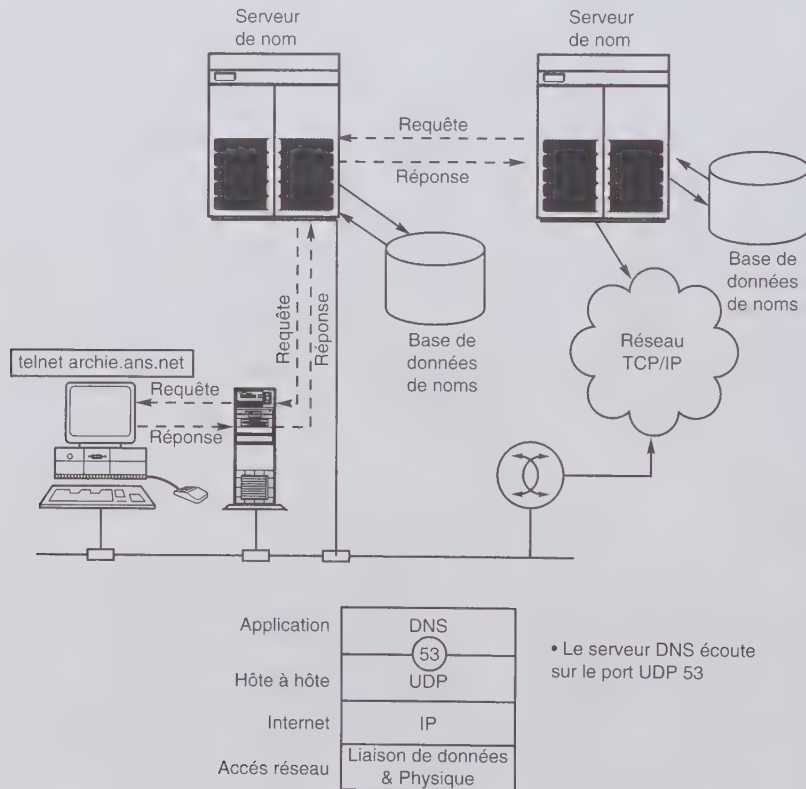
La première réponse de la session Telnet est la suivante :

Trying 147.225.1.2...

Les couches d'application TCP/IP ont transformé le nom d'hôte *archie.ans.net* en adresse IP sur 32 bits 147.225.1.2. Cette transformation a été effectuée par DNS (voir Figure 13.1).

Figure 13.1

Exemple de résolution DNS.



Les logiciels utilisant TCP/IP peuvent être configurés pour utiliser DNS afin de résoudre les noms symboliques d'hôtes. Lorsqu'une application TCP/IP rencontre un nom d'hôte, elle envoie une requête à un *résolveur de noms* afin de transformer le nom symbolique en adresse IP. Sur la plupart des plates-formes, le résolveur de noms est situé sur la station de travail d'où la requête a été émise. Le résolveur de noms est le plus souvent implémenté sous forme d'un ensemble de routines système.

Si le résolveur de noms ne parvient pas à trouver l'adresse, il transmet la requête à un *serveur de noms* recensé. Généralement, ce serveur de noms est situé sur le même réseau que la station de travail. S'il ne trouve pas la réponse à la requête, il peut la transmettre à un autre serveur de nom du réseau TCP/IP.

Le système DNS fonctionne sur le principe requête/réponse et s'appuie sur le protocole de transport UDP. Le protocole UDP est bien adapté à ce principe de requête/réponse puisqu'il ne perd pas de temps à maintenir une connexion pour la transmission des données. Le protocole TCP peut aussi convenir pour les applications à base de requête/réponse, mais le coût d'établissement et de fermeture d'une connexion ne se justifie pas pour un seul échange requête/réponse.

Dans l'exemple de Telnet décrit dans cette section, DNS est utilisé par le service d'application afin de résoudre le nom DNS en l'adresse IP équivalente. De fait, cette résolution de nom est la principale utilisation de DNS, qui permet d'établir une correspondance entre un nom symbolique et une adresse IP. DNS a une autre fonction importante, en rapport avec les services de courrier électronique. DNS sert à indiquer le serveur de mail utilisé pour gérer les adresses e-mail.

L'implémentation majoritaire de DNS est BIND (*Berkeley Internet Domain Name*), initialement développée pour la plate-forme Unix BSD. Elle est à présent disponible sur la plupart des environnements Unix, où on la retrouve le plus souvent sous l'appellation *named* (name daemon).

Nous avons vu dans cette section que les noms d'hôte comportaient des points. DNS utilise en effet un système de nommage hiérarchique à structure arborescente. A la racine de l'arbre, se trouve le domaine racine, dont le nom est un point (.). Comme tous les noms (ayant la même racine dans l'espace de nommage) se terminent par ce point, vous pouvez dans la plupart des applications TCP/IP l'omettre lorsque vous spécifiez un nom. Sous cette racine (voir Figure 13.2), on trouve les domaines de premier niveau. Ces domaines permettent de regrouper les noms de domaines (voir Tableau 13.2).

Tableau 13.1 : Exemples de domaines de premier niveau

Domaine	Description
COM	Organisations commerciales
EDU	Education (USA) : universités, écoles, etc.
MIL	Militaires (USA)
GOV	Gouvernement américain

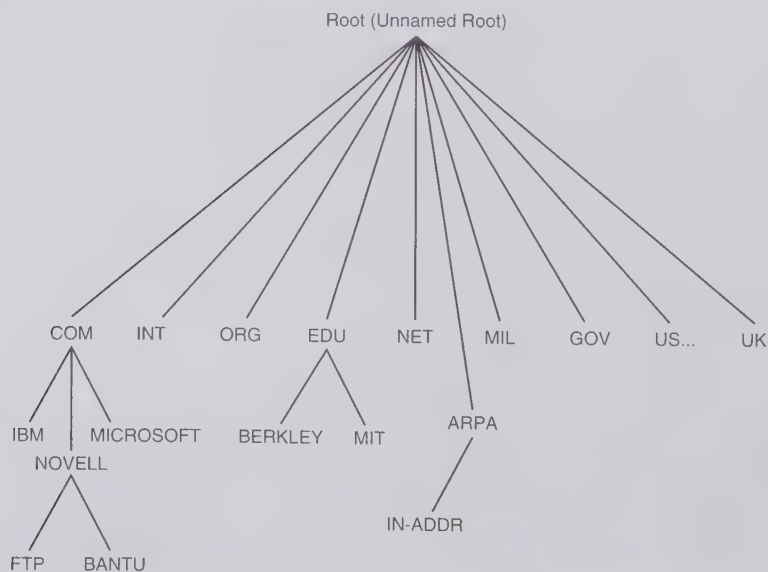
* 20 4



Tableau 13.1 : Exemples de domaines de premier niveau (*suite*)

Domaine	Description
NET	Organisations de gestion de l'Internet
ORG	Autres organisations
ARPA	ARPAnet : obsolète, encore utilisé pour la résolution inverse de noms
INT	Organisations internationales
US	Pays : Etats-Unis
CA	Pays : Canada
UK	Pays : Angleterre
DE	Pays : Allemagne
SE	Pays : Suède
FR	Pays : France
IN	Pays : Inde
CN	Pays : Chine
JA	Pays : Japon

Figure 13.2

Hiérarchie des noms sous DNS.

La désignation de chaque pays sur deux lettres est définie dans un standard CCITT (à présent appelé ITU, pour *International Telecommunications Union*), et par le standard ISO-3166 (à l'exception de la Grande-Bretagne, qui utilise UK au lieu de GB). Ces désignations sont les mêmes que celles utilisées par les services NetWare d'annuaire. Sous les domaines de premier niveau, on trouve des domaines intermédiaires. Il peut y en avoir un nombre quelconque. Chaque nom est séparé de son supérieur hiérarchique par un point (qui ne peut jamais faire partie d'un nom de domaine). Les noms de domaines s'écrivent indifféremment en majuscules ou en minuscules. La longueur totale d'un nom de domaine ne doit jamais excéder 255 caractères, comme dans l'exemple suivant :

archie.ans.net

Dans archie.ans.net, le nom d'hôte est archie, qui appartient au domaine ans.net. Supposons qu'un autre hôte s'appelle sparky, et qu'il fasse aussi partie du domaine ans.net. Son nom complet (souvent abrégé en FQDN, pour *Fully Qualified Domain Name*) sera :

sparky.ans.net

La plupart des noms de niveau intermédiaire reprennent le nom d'une organisation (entreprise, école, ministère, etc.). Les organisations peuvent librement définir des sous-domaines, mais il est alors de leur ressort de mettre en place des serveurs DNS afin de résoudre les noms de ces sous-domaines. Par exemple, imaginons une organisation, SCS, à laquelle on a attribué le nom de domaine suivant :

SCS.COM

Supposons que cette organisation dispose de réseaux distincts pour ses branches administrative, marketing et recherche. Elle peut définir trois sous-domaines nommés ADMN, MKTG et RECH, et mettre en place un ou plusieurs serveurs DNS afin de résoudre les noms de ces réseaux. Les domaines seraient les suivants :

ADMN.SCS.COM

MKTG.SCS.COM

RECH.SCS.COM

Bien qu'il ne soit pas nécessaire de disposer d'un serveur DNS par domaine, il est d'usage d'en avoir un ou plusieurs par domaine. Dans la Figure 13.2, il y aurait plusieurs serveurs DNS pour le domaine racine. Ces serveurs connaissent les noms des domaines de premier niveau, par exemple COM, EDU, MIL, ORG, NET, FR etc.



Il existe un certain nombre de serveurs de noms gérant les noms de domaines du niveau racine. Le Tableau 13.2 en propose quelques-uns.

Tableau 13.2 : Exemples de serveurs DNS du domaine racine

Nom d'hôte	Adresse IP	Programme serveur
NS.INTERNIC.NET	198.41.0.4	BIND (Unix)
NS.NIC.DDN.MIL	192.112.36.4	BIND (Unix)
NS1.ISI.EDU	128.9.0.107	BIND (Unix)
AOS.ARL.ARMY.MIL	128.63.4.82	BIND (Unix)
	192.5.25.82	
C.NYSER.NET	192.33.4.12	BIND (Unix)
TERP.UMD.EDU	128.8.10.90	BIND (Unix)
NS.NASA.GOV	192.52.195.10	BIND (Unix)
	128.102.16.10	
NIC.NORDU.NET	192.36.148.17	BIND (Unix)

Plusieurs serveurs DNS peuvent être utilisés pour le même domaine afin d'équilibrer la charge, de minimiser le trafic sur le réseau et d'assurer un service continu en cas de défaillance d'une machine. Le domaine COM peut disposer d'un ou plusieurs serveurs DNS connaissant le nom de toutes les organisations commerciales qu'il regroupe. A l'intérieur du domaine COM, un sous-domaine tel que IBM.COM disposera de ses propres serveurs DNS. A titre d'exemple, l'hôte WORLD.STD.COM interrogera le serveur DNS du domaine STD.COM afin de connaître l'adresse IP des hôtes FTP.NOVELL.COM, ou ATHENA.SCS.ORG. Une fois la requête résolue, l'adresse obtenue est en général conservée dans un cache local pendant une période de temps paramétrable.

Les serveurs DNS doivent résoudre les noms des hôtes de leur domaine. Un serveur DNS secondaire doit connaître l'adresse IP du serveur primaire à contacter afin de résoudre les requêtes. Un serveur DNS doit aussi connaître l'adresse IP de son serveur DNS parent.

DNS utilise UDP comme protocole de transport pour envoyer des requêtes DNS et recevoir des réponses en provenance des serveurs. Les serveurs DNS attendent les requêtes sur le port UDP numéro 53.

DNS est décrit dans les RFC 1034 (STD 13), 1982, 1876 et 1101.

Les RFC 2137 et 2136 décrivent les modifications récentes de DNS.

Protocoles de courrier

Le service de courrier est probablement l'application la plus utilisée sur l'Internet. Plusieurs protocoles sont disponibles, le plus courant étant SMTP (*Simple Mail Transfer Protocol*). En raison du nombre croissant de portables et d'utilisateurs mobiles, d'autres protocoles ont été développés, principalement POP3 (*Post Office Protocol*, version 3) et IMAP4 (*Internet Message Access Protocol*, version 4).

• SMTP

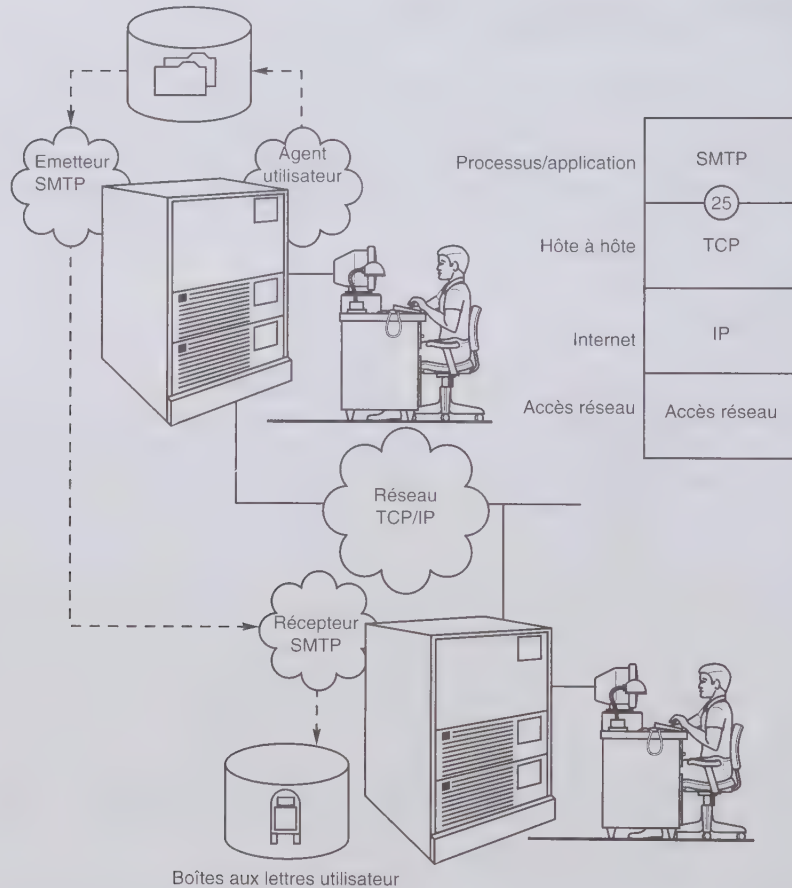
SMTP permet d'envoyer des messages texte ASCII vers des hôtes disposant de services de courrier. La Figure 13.3 montre une session de mail utilisant SMTP. Un utilisateur désireux d'envoyer un courrier interagit avec le système local de courrier en utilisant le composant UA (*User Agent*, ou "Agent Utilisateur"). Le courrier est ensuite déposé dans une boîte à lettres locale d'envoi. Un processus d'envoi SMTP parcourt régulièrement cette boîte à lettres et établit une connexion TCP avec le destinataire des messages qu'il y trouve. Le processus de réception SMTP du destinataire accepte la connexion et le courrier est alors envoyé sur cette même connexion. Le processus de réception SMTP dépose le courrier dans la boîte à lettres de destination. Si la boîte à lettres n'existe pas, un message est renvoyé à l'expéditeur pour l'en informer. Les processus d'envoi et de réception SMTP s'appellent des MTA (*Message Transfer Agent*).

Les adresses mail utilisées par SMTP se conforment au standard RFC 882. Les en-têtes sont souvent appelés des *en-têtes* 882. Voici un exemple d'adresse 882 :

KSS@SHIVA.COM

La chaîne précédant le symbole @ indique le nom de boîte à lettres et le reste de la chaîne spécifie le nom d'hôte. Si le nom de boîte à lettres inclut des caractères spéciaux, par exemple %, le nom de boîte à lettres incorpore un encodage particulier qui sera utilisé par les routeurs de mail. Dans l'adresse mail KSS@SHIVA.COM, la chaîne KSS est le nom d'une boîte à lettres située sur l'hôte SHIVA.COM.

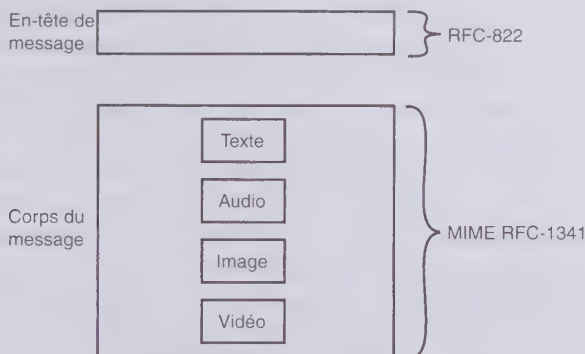


Figure 13.3*Une session SMTP.*

• Le récepteur SMTP écoute sur le port TCP 25

Pour envoyer par le biais de SMTP des messages non textuels, par exemple des fichiers binaires, audio ou vidéo, vous pouvez encoder le message sous forme de texte grâce à l'utilitaire UUENCODE, disponible sur la plupart des plates-formes. Le destinataire de votre message devra le décoder en utilisant un utilitaire nommé UUDECODE. Une autre technique d'envoi de messages non textuels consiste à utiliser le protocole MIME (*Multipurpose Internet Mail Extensions*), décrit dans les RFC 1896, 2045, 2046 et 2049. MIME permet d'encoder différents types de contenus, notamment du texte pur, du texte formaté en RTF, des images, des sons, etc. (voir Figure 13.4).

Figure 13.4
 Un message MIME
 (source : Learning Tree).



Le corps d'un message MIME admet des contenus imbriqués, et les UA (*User Agent*, ou "Agent Utilisateur") MIME peuvent en profiter pour représenter le contenu véhiculé de différentes façons. Par exemple, un terminal rudimentaire incapable d'afficher des images ou des sons pourra ainsi afficher la portion de texte du message MIME. Un autre intérêt de MIME est la possibilité de référencer des données stockées de façon externe. Par exemple, un message MIME peut inclure un pointeur vers un document sur un site FTP. Cette technique permet notamment de gérer une liste de diffusion contenant des fichiers volumineux sans imposer à tous les membres de la liste le téléchargement de tous les fichiers.

La session de courrier électronique suivante utilisant le programme Unix mail montre comment l'utilisateur interagit avec l'UA :

```
% mail
Mail version SMI 4.0 Thu Jul 23 13:52:20 PDT 1992 Type ? for
help.
"/usr/spool/mail/karanjit": 1 message
> 1 kss@RAMA.COM Mon Apr 25 19:32 5148/153370
& ?
cd [directory]      chdir to directory or home if none given
d [message list]    delete messages
e [message list]    edit messages
f [message list]    show from lines of messages
h                  print out active message headers
m [user list]       mail to specific users
n                  goto and type next message
p [message list]    print messages
```



```

pre [message list]    make messages go back to system mailbox
q                    quit, saving unresolved messages in mbox
r [message list]      reply to sender (only) of messages
R [message list]      reply to sender and all recipients of
messages
s [message list] file  append messages to file
t [message list]      type messages (same as print)
top [message list]    show top lines of messages
u [message list]      undelete messages
v [message list]      edit messages with display editor
w [message list] file  append messages to file, without from
line
x                    quit, do not change system mailbox
z [-]                display next [previous] page of headers
!                    shell escape
A [message list] consists of integers, ranges of same, or user
names separated
by spaces. If omitted, Mail uses the current message.
& m karanjit@kscs.com
Subject: Mail demonstration message
This is a demonstration on using the
simple mail program interface.
When done you must type period (.)
:-)
.
EOT
& h

```

(Affiche les en-têtes de message)

```

> 1 kss@RAMA.COM Mon Apr 25 19:32 5148/153370
& p

```

(Message Print. Le message suit)

```

Message 1:
From kss@RAMA.COM Mon Apr 25 19:32:35 1994
Return-Path: <kss@RAMA.COM>
Received: by world.std.com (5.65c/Spike-2.0)
    id AA01519; Mon, 25 Apr 1994 19:31:33 -0400
Received: from sita.RAMA.COM by relay1.UU.NET with SMTP
    (5.61/UUNET-internet-primary) id AAwnfh17101; Mon, 25 Apr
94 19:21:26 -0
400

```

```

Received: by sita.RAMA.COM (5.67/PERFORMIX-0.9/08-16-92)
      id AA03921; Mon, 25 Apr 94 16:21:21 -0700
Date: Mon, 25 Apr 94 16:21:21 -0700
From: kss@RAMA.COM (K S)
Message-Id: <9404252321.AA03921@learn1.Lrntree.COM>
To: karanjit@world.std.com
Status: R0
X-Status: D
#! /bin/sh
# This is a shell archive. Remove anything before this line,
# then unpack
# it by saving it into a file and typing "sh file". To overwrite
# existing
# files, type "sh file -c". You can also feed this as standard
# input via
# unshar, or by typing "sh <file", e.g.. If this archive is
# complete, you
# will see the following message at the end:
#      "End of shell archive."
# Contents: INSTALL Makefile Prospero RCS README aalloc.c
# archie.c
# archie.man atalloc.c dirsend.c get_pauth.c get_vdir.c
# p_err_text.c
# pauthent.h pcompat.h perrno.h pfs.h pmachine.h pprot.h
# ptalloc.c
# regex.c stcopy.c support.c uw-copyright.h vl_comp.c vlalloc.c
# Wrapped by darwin@king.csri on Wed Jan 5 20:28:52 1994
PATH=/bin:/usr/bin:/usr/ucb ; export PATH
if test -f 'INSTALL' -a "${1}" != "-c" ; then
    echo shar: Will not clobber existing file \"'INSTALL'\"
else
    echo shar: Extracting \"'INSTALL'\" \ (1725 characters\ )
    sed "s/^X/" >'INSTALL' <<'END_OF_FILE'
X[Last changed: 07/31/91]

```

(Reste du message....)

& x

(Quitter et ne pas modifier la boîtes à lettres système)

%



Les messages de courrier sont envoyés en utilisant les commandes SMTP reprises dans les Tableaux 13.3 et 13.4. Ces tableaux contiennent les commandes SMTP d'envoi et de réception d'une implémentation minimale de SMTP. Toutes les commandes SMTP sont constituées de quatre lettres. Le processus de réception est souvent un serveur mail qui répond aux commandes SMTP par des groupes de trois chiffres selon la syntaxe suivante :

NNN Message texte

où NNN est le code de retour.

Tableau 13.3 : Commandes SMTP d'envoi (implémentation minimale)

Commande	Fonctionnalité
HELO exp	Requête de connexion en provenance d'un expéditeur SMTP.
MAIL FROM: adr_exp	Permet de lancer une transaction mail dans laquelle les données seront transmises à une ou plusieurs boîtes à lettres.
RCPT TO: ad_dest	Permet de spécifier un destinataire du courrier. Pour indiquer plusieurs destinataires, on répète cette commande.
DATA	Le récepteur interprète toutes les données suivantes comme les données du mail en provenance de l'expéditeur, et ce jusqu'à l'apparition d'une ligne constituée d'un seul point, à savoir la séquence <CRLF> . <CRLF>.
QUIT	Demande au récepteur d'envoyer la réponse OK et de refermer la connexion.
RSET	Demande l'annulation du mail en cours.
NOOP	Ne fait rien, mais demande au récepteur de renvoyer la réponse OK, par exemple à fin de diagnostic.

Tableau 13.4 : Exemple de commandes SMTP côté récepteur (serveur de mail)

Commande	Fonctionnalité
250	Action demandée bien effectuée (message OK)
251	Utilisateur non local, le message sera retransmis
450	Action demandée non effectuée, boîte à lettres inaccessible (par exemple, occupée)
550	Action demandée non effectuée, boîte à lettres inaccessible

Tableau 13.4 : Exemple de commandes SMTP côté récepteur (serveur de mail) (suite)

Commande	Fonctionnalité
451	Action demandée annulée : erreur pendant le traitement
551	Utilisateur non local : redirigez le message
452	Action demandée non effectuée : espace de stockage insuffisant
552	Action demandée non effectuée : dépassement de quota disque
553	Action demandée non effectuée : nom de boîte à lettres illégal (par exemple, syntaxe erronée).
354	Commencez à transmettre le mail, terminez le flux par <CRLF> .<CRLF>
554	Echec de la transaction

La Figure 13.5 montre une session SMTP utilisant quelques-unes de ces commandes. Remarquez que dans cet exemple, le courrier est envoyé par l'utilisateur **kss@scs.psi.com**, identifié par la commande mail.

Le courrier est transmis à **jones@scsi.psi.com** et à **bob@ftree.psi.com**. Remarquez que la boîte à lettres **bob@ftree.psi.com** n'existe pas, d'où un message d'erreur (code d'état 550) renvoyé pour indiquer cet état de fait. Le serveur de mail répond à toutes les autres commandes avec un code d'état de 250.

Les documents de standardisation de SMTP sont décrits dans le Tableau 13.5

Tableau 13.5 : RFC traitant de SMTP

Protocole	Nom	Etat	RFC No	STD No
SMTP	Simple Mail Transfer Protocol	Rec	821	10
SMTP-SIZE	SMTP Service Ext for Message Size	Rec	1870	10
SMTP-EXT	SMTP Service Extensions	Rec	1869	10
MAIL	Format of Electronic Mail Messages	Rec	822	11



Figure 13.5

*Exemple de session SMTP
montrant les commandes SMTP
(source : Learning Tree).*

Example SMTP Session

```

S:  HELO machine
R:  HELO machine,  Pleased to meet you

S:  MAIL FROM:   kss@scs.psi.com
R:  250 OK

S:  RCPT To:     jones@scs.psi.com
R:  250 OK

S:  RCPT To:     bob@ltree.psi.com
R:  550 No such user here
} User bob does not have a mailbox at ltree.psi.com

S:  RCPT To:     john@ltree.psi.com
R:  250 OK

S:  DATA
R:  354 Strart mail input; end with <CRLF>.<CRLF>
S:  message text
S:  message text
S:  message text_etc.
S:  <CRLF>.<CRLF>
R:  250 OK

```

R = SMTP receiver
S = SMTP sender

• POP3 (Post Office Protocol Version 3)

SMTP s'attend à ce que l'hôte de destination, à savoir le serveur recevant le courrier, soit sur le réseau. Dans le cas contraire, il sera impossible d'établir une connexion TCP avec lui. Pour cette raison, il est peu souhaitable d'établir une session SMTP avec les ordinateurs qui ne sont pas allumés en permanence.

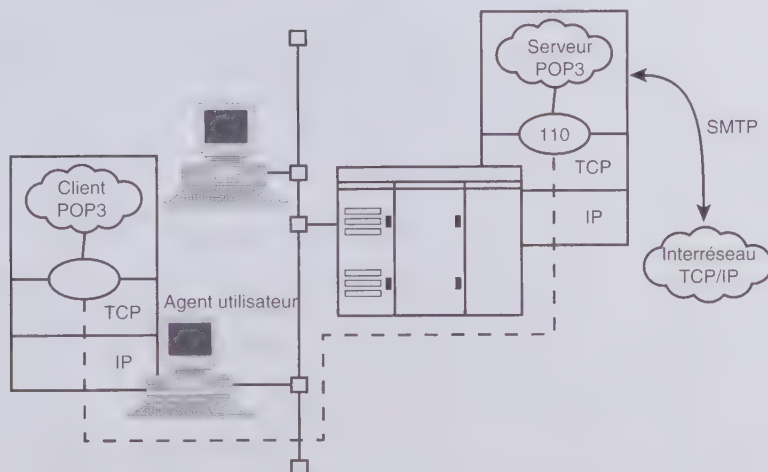
Dans la plupart des environnements réseau, les messages SMTP sont reçus par un hôte SMTP qui est toujours disponible (voir Figure 13.6). Cet hôte SMTP dispose d'un système de dépose de courrier. Les stations de travail interagissent avec l'hôte SMTP et récupèrent les messages qui leur sont destinés en utilisant un protocole client/serveur de courrier, par exemple POP3, décrit dans la RFC 1939. POP3 utilise TCP comme protocole de transport et le serveur POP3 écoute sur le port TCP 110.

Bien que POP3 soit utilisé pour télécharger des messages en provenance du serveur, on utilise encore SMTP pour envoyer des messages vers le serveur SMTP.

Les Tableaux 13.6 à 13.8 donnent la liste des commandes POP3 tirées de la spécification RFC 1939. Bien que les commandes USER et PASS (voir Tableau 13.7) soient déclarées facultatives par la RFC 1939, la plupart des implémentations les

reconnaissent. Ces commandes sont remplacées par la méthode d'authentification MD5 (*Message Digest* version 5) utilisée dans la commande APOP.

Figure 13.6
Architecture client/serveur
POP3.



- L'agent MTA s'exécute sur un ordinateur qui possède davantage de ressources qu'une station de travail
- Il offre ainsi aux stations un service de dépose de courrier.
- POP3 fournit un accès dynamique au serveur de dépose de courrier.

Tableau 13.6 : Les commandes POP3 indispensables

Commande	Signification
STAT	Permet de spécifier une réponse positive, composée de +OK suivi d'un espace, du nombre de messages en attente, d'un espace et de la taille en octets des messages. Exemple de réponse : +OK msgid size.
LIST [msgid]	Lorsqu'un numéro de message est indiqué, le serveur POP3 renvoie des informations plus précises sur le message, notamment son numéro et sa taille. Lorsque aucun numéro de message n'est précisé, une réponse positive est renvoyée, composée d'une ligne par message en attente dans la boîte à lettres et donnant les mêmes informations que ci-avant.



Tableau 13.6 : Les commandes POP3 indispensables (*suite*)

Commande	Signification
RETR msg	Permet de récupérer une liste de messages dans la boîte aux lettres de l'utilisateur sur le serveur POP3. Le serveur POP3 transmet tout d'abord une réponse positive +OK, suivie du message. S'il est impossible de renvoyer le message, le serveur renvoie -ERR.
DELE msg	Demande la suppression du message spécifié.
NOOP	Ne fait rien et demande simplement au serveur POP3 de répondre par +OK, souvent à fin de diagnostic.
RSET	Annule toutes les demandes en attente de suppression de messages. Le serveur renvoie alors +OK. Exemple de réponse : +OK mail drop has 3 messages (430 bytes)
QUIT	Le serveur POP3 supprime tous les messages demandés et renvoie +OK (réponse positive) ou -ERR (en cas de problème). Le serveur POP3 libère alors tout verrou qu'il aurait pu poser sur les fichiers de la boîte à lettres et referme la connexion TCP.

Tableau 13.7 : Commandes POP3 facultatives

Commande	Signification
USER nom	Permet de spécifier un nom de boîte à lettres.
PASS mot_de_passe	Permet de transmettre le mot de passe de la boîte à lettres.
TOP msg n	Le serveur POP3 renvoie une réponse +OK suivie des en-têtes du message spécifié, puis une ligne vide suivie des n premières lignes du corps du message.
UIDL [msg]	Permet de renvoyer un identifiant unique de listing de message (UIDL) pour le message. Le serveur POP3 envoie une réponse +OK avec une ligne contenant les informations du message. Cette ligne est appelée ID unique de listing du message. Sans argument, le serveur POP3 renvoie un +OK suivi d'une réponse sur plusieurs lignes. Après le +OK initial, le serveur renvoie pour chaque message une ligne contenant des informations relatives à ce message. Un UIDL se compose du numéro du message suivi d'un espace puis de l'identifiant unique du message.
APOP nom digest	nom est une chaîne permettant d'identifier la boîte à lettres et digest est la chaîne de digest de MD5. Cette commande est utilisée pour échanger les mots de passe autrement que par USER/PASS, qui utilise une transmission en clair. La méthode d'authentification APOP protège la transmission dans les deux sens et le mot de passe ne transite jamais en clair sur le réseau.

Tableau 13.8 : Réponses du serveur POP3

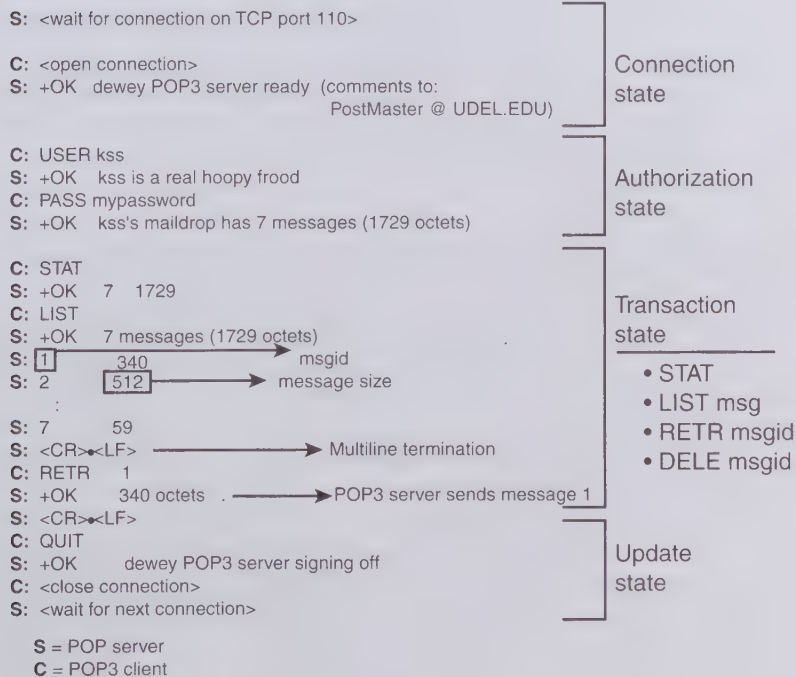
Commande	Signification
+OK	La commande a été exécutée correctement.
-ERR	L'exécution de la commande a provoqué une erreur.

La Figure 13.7 donne un exemple de session entre un client POP3 et un serveur POP. Cette session utilise certaines des commandes vues dans les tableaux précédents.

Figure 13.7

Exemple de session POP3
(source : Learning Tree).

Example POP3 Session



Cet exemple montre que la session POP3 est initialement dans l'état de connexion. Dans cet état, on établit la connexion TCP avec le serveur POP3. Ensuite, la session passe dans un état d'autorisation, dans lequel l'utilisateur doit fournir un nom d'utilisateur et un mot de passe afin d'être identifié par le serveur POP3.



Dans les implémentations précédentes de POP3, ces informations transitaient en clair sur le réseau. Il est maintenant possible d'éviter cet inconvénient en utilisant MD5. Pour en savoir plus à ce sujet, reportez-vous à la RFC 1939.

Une fois l'utilisateur identifié, la session POP3 entre dans l'état de transaction. Dans cet état, on peut utiliser un certain nombre de commandes (par exemple STAT, LIST, RETR, DELE, RSET, etc.) Dans la Figure 13.7, le client POP3 envoie une commande STAT et le serveur renvoie le nombre de messages, ainsi que leur taille totale (1729 octets). Le client POP3 utilise alors la commande LIST pour demander la liste des messages. Le serveur POP3 affiche le numéro de message de chaque message ainsi que sa taille. Le client envoie alors des commandes RETR et spécifie l'identifiant de chacun des messages à lui faire parvenir. En fonction du paramétrage, le client POP3 enverra une commande DELE par message récupéré.

Une fois les messages transférés, la session POP3 entre dans l'état de mise à jour. Dans cet état, le client POP3 envoie au serveur une commande QUIT afin de refermer la connexion. Le client et le serveur POP3 mettent alors à jour leurs configurations internes afin de prendre en compte la session. La connexion TCP est ensuite refermée.

• IMAP4 (Internet Message Access Protocol Rev 4)

Bien que POP3 soit un système client/serveur bien adapté pour que des stations de travail puissent récupérer leur courrier, il présente certains points faibles. Par exemple, le courrier doit être téléchargé sur la station de travail avant d'être exploité, POP3 ne permettant pas de manipuler directement les messages sur le serveur. Pour ces raisons, IMAP4 est une alternative satisfaisante à POP3 de plus en plus souvent employée.

IMAP4 permet à un protocole client/serveur l'accès à des messages et leur manipulation directement sur le serveur. Le protocole permet de manipuler des répertoires, appelés boîtes à lettres, d'une façon fonctionnellement équivalente aux boîtes à lettres locales. IMAP4 permet aussi à un client de se resynchroniser avec le serveur.

Les fonctionnalités IMAP4 sont les suivantes :

- accéder et manipuler des portions de messages e-mail sur le serveur sans avoir à les télécharger ;
- voir des messages et les pièces jointes sans les télécharger ;
- télécharger tous les messages pour les consulter hors connexion ;
- resynchroniser les boîtes à lettres locales avec celles du serveur.

IMAP4 dispose des opérations suivantes :

- créer, supprimer et renommer une boîte à lettres ;

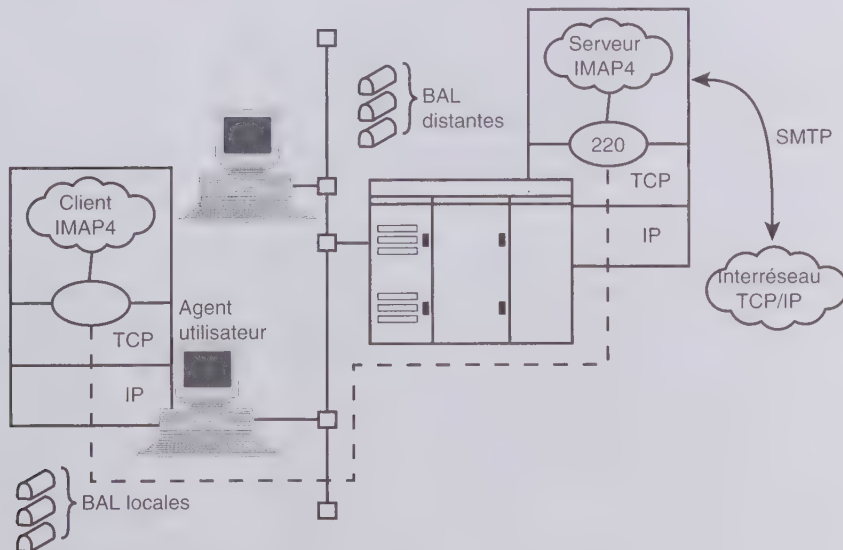
- vérifier l'arrivée de nouveaux messages ;
- supprimer des messages des boîtes à lettres ;
- définir et modifier des flags définissant l'état des messages ;
- reconnaître les en-têtes RFC 822 et décoder des messages MIME ;
- rechercher et récupérer de façon sélective des attributs de messages, des textes et des portions de messages.

IMAP4 identifie chaque message par un numéro. Ces numéros sont soit des identifiants uniques soit des valeurs incrémentales. IMAP4 ne fonctionne qu'avec un seul serveur. Un mécanisme permettant de configurer IMAP4 pour fonctionner avec plusieurs serveurs est en cours d'étude par l'IETF.

Tout comme POP3, IMAP4 ne spécifie pas de méthode d'envoi de courrier. Cette fonctionnalité est gérée par un autre protocole, par exemple SMTP. La Figure 13.8 montre les interactions client/serveur de IMAP4.

Figure 13.8

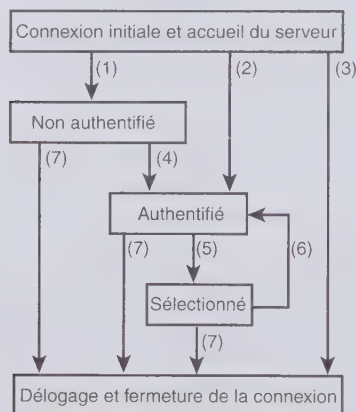
Architecture client/serveur
de IMAP4 (source : *Learning
Tree*).



Le comportement de IMAP4 peut être décrit en utilisant un diagramme d'états (voir Figure 13.9).

Figure 13.9

Le diagramme d'états de IMAP4.



- 1) Connexion sans préauthentification (accueil OK,
- 2) Connexion préauthenticée (accueil PREAUTH)
- 3) Connexion rejetée (accueil BYE)
- 4) Commande LOGIN ou AUTHENTICATE réussie
- 5) Commande SELECT ou EXAMINE réussie
- 6) Commande CLOSE ou échec d'une commande SELECT ou EXAMINE
- 7) Commande LOGOUT, extinction du serveur ou fermeture de la connexion

Comme l'illustre la Figure 13.9, IMAP4 peut être dans quatre états différents. La plupart des commandes IMAP4 ne sont valides que dans un état particulier. Une erreur de protocole est générée lorsqu'un client demande une commande illégale dans l'état considéré. Les états IMAP4 sont les suivants :

- état non authentifié ;
- état authentifié ;
- état sélectionné ;
- état logout.

Dans l'état *non authentifié*, le client IMAP4 communique au serveur les informations permettant de l'identifier. La plupart des commandes sont interdites tant que cette phase n'a pas été franchie. On entre dans cet état lorsque la connexion est établie, à moins que la connexion n'ait été pré-authenticée par d'autres moyens.

Dans l'état *authentifié*, le client est authentifié et doit sélectionner une boîte à lettres avant de pouvoir exécuter des commandes affectant des messages. On entre dans cet état une fois l'authentification réussie. Lorsqu'une erreur se produit lors de la sélection d'une boîte à lettres, on retourne dans cet état afin qu'une autre boîte à lettres puisse être sélectionnée.

L'*état sélectionné* est atteint lorsqu'une boîte à lettres valide a été sélectionnée.

Dans l'*état logout*, la connexion est parvenue à son terme et IMAP4 la referme. Cet état peut être atteint à l'initiative du client ou du fait du serveur.

Le Tableau 13.9 décrit les RFC relatives à IMAP4.

Tableau 13.9 : RFC relatives à IMAP4

RFC	Etat	Titre de la RFC
2095	PS	IMAP/POP AUTHorize Extension for Simple Challenge/Response
2088	PS	IMAP4 nonsynchronizing literals
2087	PS	IMAP4 QUOTA extension
2086	PS	IMAP4 ACL extension
2061	I	IMAP4 COMPATIBILITY WITH IMAP2BIS
2060	PS	INTERNET MESSAGE ACCESS PROTOCOL VERSION 4rev1
1733	I	DISTRIBUTED ELECTRONIC MAIL MODELS IN IMAP4
1732	I	IMAP4 COMPATIBILITY WITH IMAP2 AND IMAP2BIS
1731	PS	IMAP4 authentication mechanisms

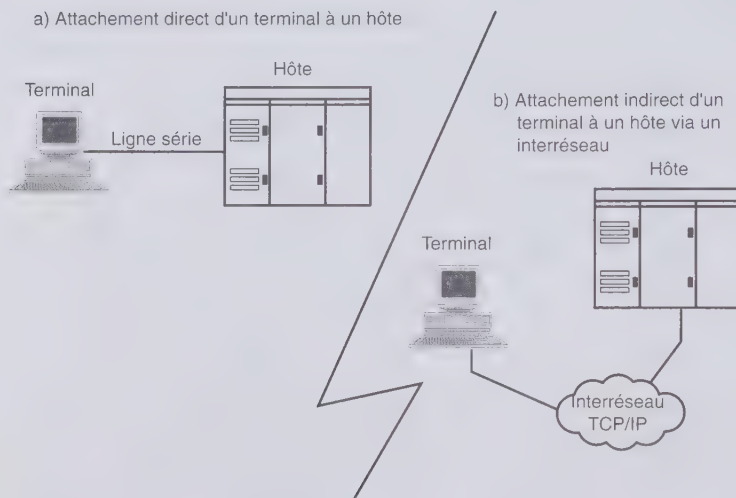
PS = Proposition de standard

I = Information

Protocoles d'accès distant

L'interaction la plus classique avec un mini-ordinateur ou un mainframe se fait via un terminal. Les caractéristiques du terminal dépendent du matériel qui le compose et du système d'exploitation de l'hôte. Une fois la connexion établie entre le terminal et l'hôte (voir Figure 13.10), l'utilisateur effectue des frappes clavier représentant des commandes. Le pilote de terminal de l'hôte reçoit les frappes clavier et les stocke pour en déduire les commandes utilisateur. La commande est alors exécutée par l'hôte, qui renvoie sa réponse.



Figure 13.10*Architecture terminal/hôte.*

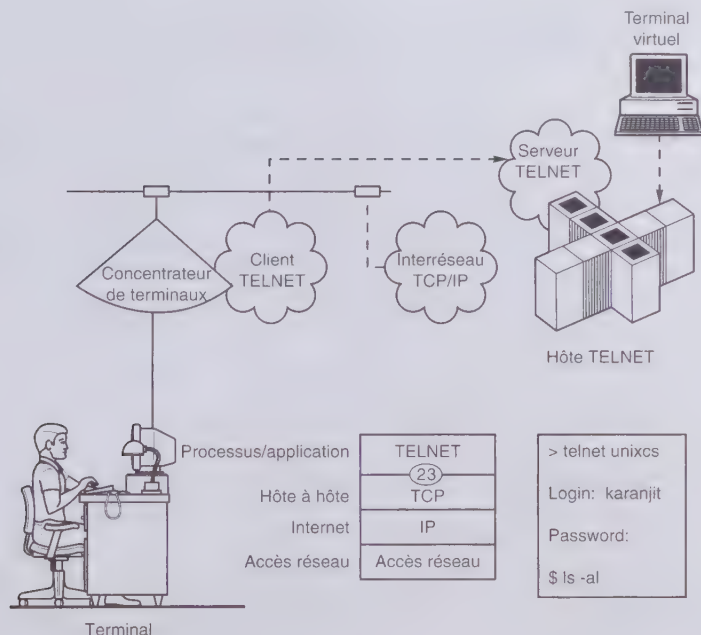
Lorsque le terminal et l'hôte sont séparés par un réseau, un protocole d'accès distant doit être utilisé pour retrouver les fonctionnalités disponibles lors de la connexion directe du terminal avec l'hôte (voir Figure 13.10). Les principaux protocoles d'accès distant de la suite TCP/IP sont :

- Telnet ;
- Les utilitaires Berkeley r*.

• Telnet

Le protocole Telnet est utilisé pour émuler une connexion de terminal à un hôte distant. Telnet utilise TCP comme protocole de transport afin de transmettre les informations entre le clavier de l'utilisateur et l'hôte distant, ainsi que pour afficher des informations en provenance de l'hôte distant sur le périphérique d'affichage de la station de travail de l'utilisateur.

La Figure 13.11 vous montre une session Telnet. Afin d'établir une session Telnet, il faut disposer d'un client Telnet sur la station de travail, et d'un serveur Telnet sur l'hôte distant. Une session TCP/IP est alors établie entre le client Telnet et le serveur Telnet. Le serveur Telnet écoute sur le port TCP 23. Sur les plates-formes Unix, le serveur Telnet s'appelle `telnetd` (*Telnet daemon*) d'après le nom du programme implémentant le serveur Telnet.

Figure 13.11*Exemple de session Telnet.*

Au fur et à mesure que l'utilisateur tape sur son clavier, les caractères sont reçus par le serveur Telnet et transmis au système d'exploitation sur lequel il s'exécute, comme si les caractères étaient saisis depuis un terminal local. Le résultat des commandes est alors transmis par le serveur au client Telnet. Le client Telnet affiche les résultats ainsi reçus sur le périphérique d'affichage de la station de travail de l'utilisateur. Du point de vue de l'utilisateur du client Telnet, la réponse semble venir d'une machine rattachée localement à sa station de travail. Une fois connecté et authentifié, vous pouvez taper n'importe quelle commande sur le système d'exploitation distant. Afin de vous connecter à un hôte distant, vous devez disposer d'un compte sur cette machine.

Voici un exemple de session Telnet sur un hôte Unix :

```

% telnet world.std.com
Trying 192.74.137.5 ...
Connected to world.std.com.
Escape character is '^]'.
* To create an account on The World login as new, no password.
SunOS Unix (world)
login: karanjit

```



Password: *password*

Last login: Tue May 10 17:52:29 from karanjit-slip.cs

OS/MP 4.1C Export(STD/arlle)#15: Fri Mar 18 17:25:40 1994

Welcome to the World! A 6 CPU Solbourne 6E/900.

Public Access Unix - Home of the Online Book Initiative

Type 'help' for help! - Stuck? Try 'help HINTS'.

When you see MORE, hit the space bar for the next page.

Use 'exit' or 'logout' to leave the system.

Still Stuck? Send mail to 'staff'.

You have mail.

Over disk quota on /home/ie, time limit has expired, remove 120K

TERM = (vt100)

Erase is Backspace

No new messages.

An authority is a person who can tell you more about something
than you

really care to know.

from: postel@isi.edu [no subject]

% ls -alr

total 1011

```
-rw----- 1 karanjit 21852 Apr 13 21:56 veronica
drwx----- 2 karanjit 512 Dec 28 1992 temp
-rw----- 1 karanjit 4780 Jun 4 1993 snapip.dmp
-rw----- 1 karanjit 1319 Jul 9 1993 Index
-rw----- 1 karanjit 8 Apr 13 20:43 .sh_history
-rw----- 1 karanjit 7035 May 10 17:53 .pinerc
drwx----- 2 karanjit 512 Apr 13 20:01 .nn
-rw----- 1 karanjit 138402 Apr 13 20:01 .newsrsrc.bak
-rw-r--r-- 1 karanjit 138102 Apr 13 20:01 .newsrsrc
-rw-r--r-- 1 karanjit 4 Apr 7 12:55 .msgsrc
-rw----- 1 karanjit 11 Dec 28 1992 .mh_profile
-rw-r--r-- 1 karanjit 386 Dec 19 1992 .login
-rw----- 1 karanjit 0 Apr 7 17:35 .gopherrc
-rw-r--r-- 1 karanjit 538 Dec 19 1992 .emacs
drwx----- 2 karanjit 512 Jan 22 1993 .elm
-rw-rw-r-- 1 karanjit 799 Feb 1 1993 .cshrc
-rw----- 1 karanjit 0 Dec 19 1992 .addressbook
drwxrwxr-x 1911 root 61952 May 24 12:55 ..
drwx----- 7 karanjit 1024 May 10 17:56 .
```

% ping novell.com

novell.com is alive

% logout

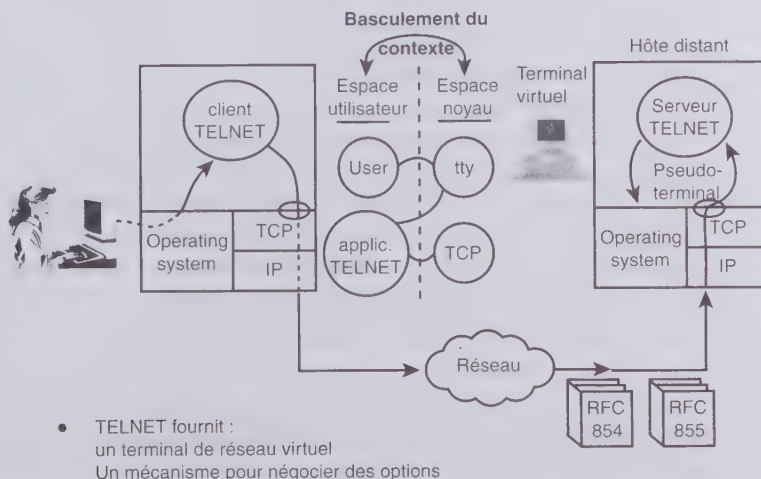
Une fois que vous êtes logué par Telnet, les commandes que vous tapez sont celles que vous taperiez si vous étiez directement présent sur la machine distante.

Architecture Telnet

La Figure 13.12 montre le modèle client/serveur de Telnet. Lorsqu'un utilisateur lance une application Telnet et spécifie un nom d'hôte de destination, une connexion TCP est établie sur le port 23 de l'hôte de destination. Les données entre le client Telnet et l'hôte Telnet (encore appelé serveur Telnet) sont transmises le long de cette connexion TCP. La connexion ne sera en temps normal refermée que lorsque l'utilisateur exécute la commande `logout` qui met fin à la session Telnet.

Figure 13.12

*Le modèle client/serveur Telnet
(source : Learning Tree).*



Une fois la connexion TCP établie, le client et le serveur Telnet entrent dans une phase de négociation visant à déterminer le type du terminal utilisé, ainsi que le mode opératoire de la session Telnet. Le client peut notamment être configuré pour transmettre chaque ligne saisie par l'utilisateur sous la forme d'un seul segment TCP. Par défaut, chaque caractère saisi est transmis dans un segment TCP.

Au fur et à mesure des saisies de l'utilisateur, les frappes clavier sont prises en compte par le pilote de terminal et transmises au client Telnet. Le client Telnet transmet en général chaque caractère dans un nouveau segment TCP. Ce mode opératoire est très coûteux en bande passante, mais ne pose pas de problèmes lorsque celle-ci est élevée et le nombre d'utilisateurs réduit. Le serveur Telnet renvoie ses réponses



dans des segments de message TCP. La réponse du serveur est composée de plusieurs caractères et son efficacité est en conséquence supérieure à celle du client.

Dans la plupart des implémentations, le serveur Telnet est lancé par un daemon Internet (*super daemon*) qui reste en écoute sur une liste de ports définis dans une base de données de configuration. Lorsqu'une connexion est reçue sur le port TCP 23, le processus serveur Telnet commence son écoute sur le port 23. Il peut y avoir plusieurs instances du daemon serveur Telnet fonctionnant simultanément afin de gérer un grand nombre de clients Telnet. Il est aussi possible d'utiliser une seule instance du serveur Telnet, qui utilisera plusieurs threads pour gérer les sessions Telnet.

Dans la Figure 13.12, le client et le serveur Telnet fonctionnent en mode utilisateur. Les systèmes d'exploitation modernes connaissent deux modes d'exécution pour faire fonctionner un programme : le mode *utilisateur* et le mode *kernel* (ou mode noyau). La plupart des applications s'exécutent en mode utilisateur et ont un accès restreint aux ressources du système. Cette restriction permet d'empêcher les applications de provoquer une panne du système. Les accès aux ressources et aux services sont gérés par des appels système qui, eux, s'exécutent en mode *kernel*. Si le client et le serveur Telnet fonctionnent en mode *utilisateur* et doivent accéder à des services du système tels que le protocole TCP/IP, la machine doit effectuer un basculement de contexte pour passer en mode *kernel*. Un tel basculement de contexte nécessite l'exécution de plusieurs cycles machine et l'utilisation de ressources du processeur.

Lorsqu'un caractère est fourni par le terminal, un basculement de contexte se produit pour permettre l'accès au pilote télétype (tty) qui s'exécute en mode *kernel*. Le pilote délivre le caractère au client Telnet, qui fonctionne en mode *utilisateur*, ce qui nécessite un autre basculement de contexte. Lorsque le client Telnet envoie à la pile TCP/IP les données correspondant au caractère, un nouveau basculement de contexte est nécessaire, puisque TCP/IP fonctionne en mode *kernel*. Dans notre exemple, on constate trois changements de contexte pour chaque caractère transmis par le client Telnet.

Du côté du serveur, un changement de contexte a lieu lorsque le caractère reçu est envoyé dans la pile TCP/IP du programme serveur Telnet. Le serveur Telnet envoie les caractères au système d'exploitation via une interface pseudoteletype (ptty), faisant intervenir un ou plusieurs changements de contexte. Lorsqu'un grand nombre de caractères est transmis par le client Telnet, la machine verra ses performances se dégrader. Si d'autres applications fonctionnent sur l'hôte Telnet, elles seront aussi affectées par cette baisse de performances.

Le standard Telnet, appelé STD 8, est défini dans les RFC suivantes :

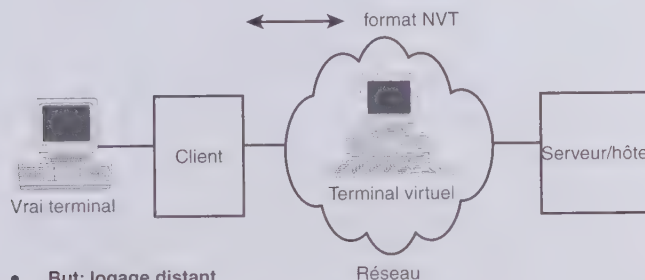
- RFC 854, "*Telnet Protocol Specification*" ;
- RFC 855, "*Telnet Option Specification*".

Format NVT et terminal NVT

Afin de prendre en compte un grand nombre de terminaux, le protocole Telnet dispose d'un mode de négociation des paramètres. Les paramètres négociés permettent de définir un terminal réseau virtuel (ou NVT, pour *Network Virtual Terminal*), comme indiqué sur la Figure 13.13.

Figure 13.13

Terminal Réseau Virtuel (ou NVT) (source : Learning Tree).



- **But: logage distant**
- **protocole Network Virtual Terminal (NVT)**
 - Définit un langage commun pour le transfert de données et d'infos de contrôle.
- **Le client fait la conversion entre le format NVT et celui de la machine client.**
- **Le serveur fait la conversion entre le format NVT et celui de la machine serveur**

Le périphérique NVT est associé à un protocole NVT qui définit la méthode permettant de transférer les données et les informations de contrôle. Les différences matérielles des terminaux, celles liées au contrôle et au transfert des informations, sont gérées par NVT.

Le client Telnet transforme les frappes utilisateur en un flux de caractères au format NVT. Ce format gère le codage des données en provenance de l'utilisateur et des données de contrôle. Les *données utilisateur* sont les touches pressées par lui, et les *données de contrôle* sont les caractères permettant d'interrompre un processus (Ctrl+C par exemple), effacer des caractères (Retour arrière), effacer des lignes, etc. Le serveur Telnet accepte les données au format NVT et les transforme en fonction du système d'exploitation du serveur Telnet. La réponse renvoyée par le serveur au client est convertie en flux NVT par le serveur Telnet. Lors de la réception de cette réponse, le client convertit le flux NVT en format natif du terminal.



Pour résumer,

- le client Telnet effectue les conversions entre NVT et le format de la machine client ;
- le serveur Telnet effectue les conversions entre NVT et le format de la machine serveur.

Le format NVT utilise des codes ASCII 7 bits pour les données utilisateur et réserve le huitième bit aux données de contrôle. La table ASCII standard (Etats-Unis) contient 95 caractères imprimables et 33 codes de contrôle. Les caractères imprimables ont la même signification que leur équivalent ASCII. Le Tableau 13.10 donne la définition des caractères de contrôle les plus courants.

Tableau 13.10 : Caractères de contrôle NVT

Nom	Code	Signification
NUL	0	Pas d'opération
BEL	7	Bip
BS	8	Curseur vers la gauche
HT	9	Tabulation horizontale
LF	10	Saut de ligne
VT	11	Tabulation verticale
FF	12	Saut de page
CR	13	Retour chariot

La Figure 13.14 donne un exemple de format NVT. L'utilisateur doit saisir la séquence de caractères suivante pour exécuter une commande :

```
cp inpo<bs>ut.doc x
```

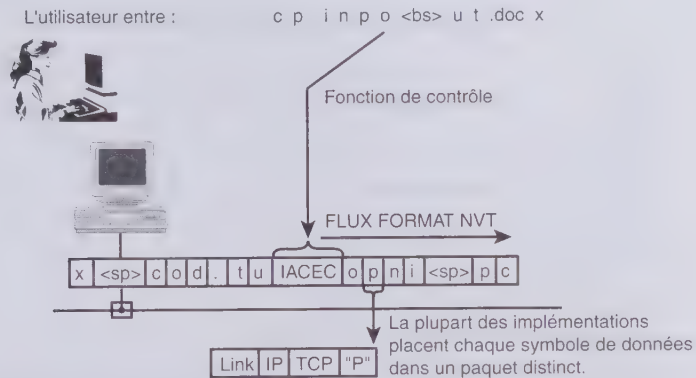
<bs> représente le caractère BackSpace, qui a pour effet d'effacer le dernier caractère saisi (le o tapé par erreur). Les caractères NVT envoyés par le client utilisent la représentation ASCII des caractères et sont transmis à raison d'un segment TCP par caractère saisi. Remarquez ce qui se produit lorsque le caractère <bs> est rencontré : il est recodé en la séquence suivante :

IAC EC

IAC signifie *Interpret As Command* ("interpréter en tant que commande"), et correspond au code ASCII 255. Le caractère EC est le caractère d'effacement, de valeur ASCII 247. Remarquez que le bit de poids fort de chacun de ces caractères est positionné à 1.

Figure 13.14

Exemple de format NVT
(source : *Learning Tree*).



- **Les fonctions de contrôles sont passées avec le flux de données.**
 - La séquence d'échappement est le caractère IAC suivi par le code de commande

IAC : Interpret As Command ("interprète comme une commande") code ASCII 255

Négociation des options de Telnet

Une fois la connexion TCP établie pour la session Telnet, les deux participants (le client et le serveur Telnet) négocient les options qui seront utilisées tout au long de la session Telnet. Pendant cette négociation d'options, certains paramétrages de configuration doivent être acceptés par les deux participants (type de terminal, half-duplex ou full-duplex, mode ligne, mode caractère, etc.).

Le client Telnet transmet des requêtes d'option en utilisant des codes Telnet spéciaux (DO, WILL, DON'T, ou WON'T). Ces codes sont transmis sous la forme de séquences de caractères commençant par le caractère IAC (code ASCII 255), servant à indiquer qu'une commande suit. Vient ensuite un caractère indiquant le type de requête, par exemple 253 pour DO et 252 pour DON'T (notez que, ici encore, le bit de poids fort est toujours positionné à 1). Voici comment coder DO et DON'T :

255 253 = IAC DO = Indicateur DO d'option

255 252 = IAC DON'T = Indicateur DON'T d'option

La Figure 13.15 donne une liste de commandes Telnet ; le Tableau 13.11, les codes de quelques-unes de ces options Telnet.



Figure 13.15

Exemples de commandes Telnet
(source : Learning Tree).

TELNET Commands Reference List

IAC (255)		Interprète l'octet suivant comme une commande
IAC (255)	DON'T (254)	L'émetteur souhaite que le récepteur désactive l'option
IAC (255)	DO (253)	L'émetteur souhaite que le récepteur active l'option
IAC (255)	WON'T (252)	L'émetteur souhaite désactiver l'option
IAC (255)	WILL (251)	L'émetteur souhaite activer l'option
IAC (255)	GA (249)	Signal "Go ahead" (poursuivre)
IAC (255)	EL (248)	Signal "Erase line" (suppression de ligne)
IAC (255)	EC (247)	Signal "Erase character" (suppression de caractère)
IAC (255)	AYT (246)	Signal "Are you There ?" (Tu es là ?)
IAC (255)	AO (245)	Signal "Abort Output" (annuler la sortie)
IAC (255)	IP (244)	Signal "Interrupt process" (Interruption du processus)
IAC (255)	BRK (243)	Signal de break
IAC (255)	DMARK (242)	La partie du flux de données du signal SYNC, accompagné par la notification Urgent de TCP fait que les données sont éliminées jusqu'au caractère DMARK
IAC (255)	NOP (241)	Signal NoOp
IAC (255)	SB (250)	Début de la sous-négociation de l'option indiquée
IAC (255)	SE (240)	Fin des paramètres de sous-négociation

Tableau 13.11 : Valeurs ASCII des codes de commandes Telnet

Nom	Code	RFC	Signification
Echo	1	857	Permet au récepteur de faire un écho des données
SGA	3	858	Suppression du signal "Go Ahead" à la fin des données
Status	5	859	Requête de l'état de l'option TELNET de l'extrémité distante
Timing	6	860	Demande l'insertion de codes de synchronisation dans le flux de retour
Terminal	24	884	Echange d'informations relatives au type du terminal
NAWS	31	1073	Négociation de la taille de la fenêtre

Tableau 13.11 : Valeurs ASCII des codes de commandes Telnet (suite)

Nom	Code	RFC	Signification
Tspeed	32	1079	Transmission d'information relatives à la vitesse du terminal
TFC	33	1080	Contrôle de flux de terminal (Distant)
Linemode	34	1116	Envoie des lignes complètes au lieu de simples caractères
Xdisploc	35	1096	X Display Location

• Utilitaires Berkeley r*

Unix BSD intègre dans sa distribution standard un ensemble de commandes appelées les *commandes r*, ou encore les *utilitaires r**. Sur des ordinateurs distants, ces commandes effectuent diverses tâches, allant du login distant à l'exécution distante de programmes. Ces commandes sont conçues pour offrir un accès transparent à une machine distante sur un réseau dans lequel les utilisateurs sont connus. Cette situation est très fréquente dans les universités, et l'était tout particulièrement à Berkeley (University of California) où les étudiants et enseignants disposaient de comptes sur nombre de machines et avaient besoin d'un accès transparent aux données et programmes d'une machine à une autre. Ces utilitaires ont été portés sur nombre d'autres plates-formes : implémentations diverses d'Unix, VMS, MVS, DOS, NT, etc., et ils sont maintenant disponible dans quasiment tous les environnements.

Le Tableau 13.12 reprend les commandes "r" les plus courantes.

Tableau 13.12 : Les utilitaires Berkeley r*

Commande	Fonctionnalité
rlogin	Identique à Telnet, permet aux utilisateurs de se connecter à des ordinateurs distants.
rexec	Permet d'exécuter une commande sur un ordinateur distant.
rsh	Permet de lancer un shell (un interpréteur de commande) sur un ordinateur distant afin d'exécuter la commande spécifiée.
rcp	Permet de copier des fichiers entre machines distantes ou entre une machine distante et la machine locale.
rwho	Affiche la liste des utilisateurs connectés sur le réseau.



Tableau 13.12 : Les utilitaires Berkeley r* (suite)

Commande	Fonctionnalité
rwall	Transmet un message à tous les utilisateurs des machines spécifiées.
ruptime	Affiche la liste des ordinateurs du réseau, leur charge, le nombre d'utilisateurs et le temps écoulé depuis le dernier redémarrage.

Les sections suivantes vous en apprendront plus sur ces utilitaires.

La commande *rlogin*

Cette commande ne dispose pas d'autant d'options négociées entre le client et le serveur que Telnet, et suppose donc que l'hôte distant connaît la configuration locale. Certaines implémentations de *rlogin* exportent l'environnement local de l'utilisateur vers la machine distante au cours de la session de login.

L'accès transparent des utilitaires r* est implémenté grâce à la notion de confiance. Pour qu'un utilisateur ait un accès transparent à un hôte, l'adresse IP de sa machine doit être référencée dans l'un des fichiers suivants de l'ordinateur distant :

- **hosts.equiv.** Sur les systèmes Unix, ce fichier est le plus souvent conservé dans le répertoire */etc*. Ce fichier permet de faire confiance aux hôtes qui y figurent, ce qui signifie en d'autres termes que ces hôtes ont un accès transparent à l'hôte distant.
- **.rhosts.** Un utilisateur peut créer ce fichier pour permettre à des utilisateurs d'autres ordinateurs d'utiliser son login.

On peut aussi spécifier, en plus des adresses IP des fichiers *.rhosts* et *hosts.equiv*, la liste des utilisateurs ayant un accès transparent à la machine. Faute de précision, tous les utilisateurs de la machine indiquée sont concernés. Le détail de l'utilisation de ces fichiers et de leur contenu varie d'une implémentation à l'autre, et vous devrez vous référer à la documentation de la machine hôte pour utiliser ces commandes.

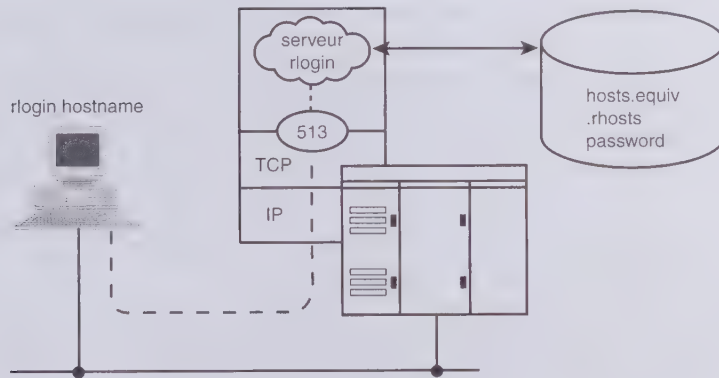
Sur les systèmes Unix, le fichier */etc/passwd* (qui contient la liste des comptes utilisateurs valides) est aussi utilisée pour vérifier que l'utilisateur dispose d'un compte sur le système.

Le plus souvent, un utilisateur exécutera la commande en spécifiant directement le nom de l'hôte distant :

```
rlogin nom_machine
```

Il est inutile de préciser un nom d'utilisateur ou un mot de passe, puisque le système est authentifié par le serveur *rlogin* (appelé *rlogind* sur les systèmes Unix). Le serveur *rlogin* s'exécute sur le système distant, qui vérifie les fichiers *hosts.equiv*, *.rhosts*, et */etc/passwd* (voir Figure 13.16).

Figure 13.16
Exemple d'utilisation
de rlogin.



L'utilitaire *rsh*

La commande *rsh* peut s'utiliser pour exécuter une commande sur l'ordinateur distant et renvoyer son résultat. Par exemple, la commande *rsh* suivante lance la commande *ps -a* sur la machine *rama.kinetics.com* et renvoie le résultat à l'ordinateur local :

```
rsh rama.kinetics.com ps -a
```

Les utilitaires *r** comprennent les environnements des deux machines, ainsi que les concepts de flux d'entrée standard (*stdin*), de flux de sortie standard (*stdout*) et de flux d'erreur standard (*stderr*) qui sont implémentés sur la majorité des systèmes d'exploitation. Il est donc possible de sauvegarder le résultat de la commande *rsh* précédente dans un fichier local en utilisant la commande suivante :

```
rsh rama.kinetics.com ps -a > fichier_local
```

L'utilitaire *rcp*

Cet utilitaire sert à copier des fichiers entre deux systèmes distants, ou entre un système distant et le système local. Les fichiers distants sont désignés en utilisant la syntaxe suivante :

```
nom_hote:chemin
```

nom_hote spécifie le nom de l'hôte (ou son adresse IP) qui contient le fichier recherché, et *chemin* est l'emplacement de ce fichier sur l'hôte indiqué.



La commande suivante permet de copier un fichier depuis le système `rama.kinetics.com` dans un fichier local :

```
rcp rama.kinetics.com:/etc/passwd passwd.loc
```

La commande suivante permet de copier un fichier depuis le système `rama.kinetics.com` vers le système `kama.kinetics.com` :

```
rcp rama.kinetics.com:/etc/ftpusers kama.kinetics.com:/ftpusers
```

Sécurité et utilitaires Berkeley r*

Le problème de l'accès transparent est qu'il suffit qu'un intrus s'introduise dans les fichiers contenant les noms d'hôte des machines de confiance et les modifie pour se donner accès à l'intégralité du système.

Certaines implémentations de r* ne fonctionnent pas avec des adresses IP, mais uniquement avec des noms de domaine. Cette sécurité supplémentaire permet de s'assurer que les ordinateurs sont enregistrés dans le DNS. Les intrus qui tentent d'accéder au système doivent alors aussi s'enregistrer dans le DNS, ce qui les rend plus faciles à repérer.

La plupart des implémentations non Unix de r* demandent à l'utilisateur un mot de passe local avant d'exécuter la commande `r`. Cependant, les mots de passe valides étant stockés dans un fichier local, le risque d'intrusion demeure dès lors que l'intrus gagne l'accès à ce fichier.

Protocoles de transfert de fichiers

Les données sur la majorité des plates-formes sont regroupées en unités appelées fichiers. Un fichier sur un système dispose des attributs suivants : le nom du fichier, les dates de création, de dernière modification et de dernier accès, la taille du fichier et d'autres encore qui varient selon la plate-forme.

Le transfert de données entre deux systèmes implique en général tout ou partie d'un fichier. Les services d'application TCP/IP intègrent deux protocoles standards pour échanger entre systèmes le contenu complet d'un fichier : FTP (*File Transfer Protocol*) et TFTP (*Trivial File Transfer Protocol*).

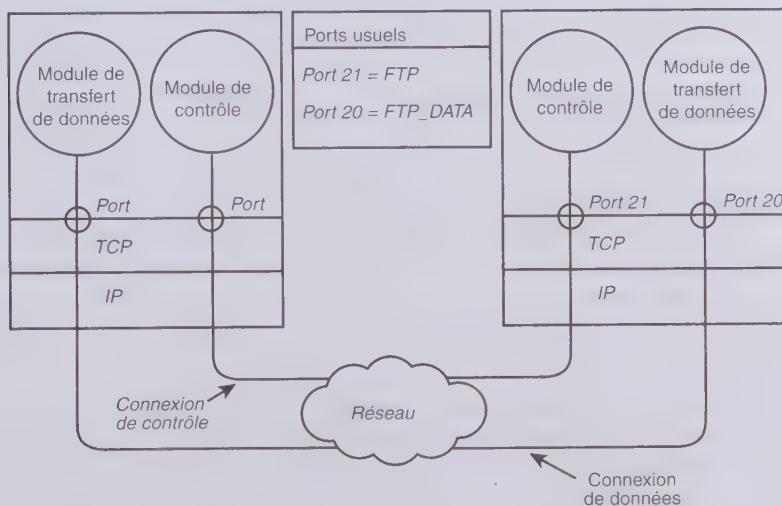
Pour un accès direct aux fichiers, lorsqu'on intègre le système de fichiers distant dans le système de fichiers local, on utilise un autre protocole standard appelé NFS (*Network File System*).

Ces protocoles seront passés en revue dans les sections suivantes.

• File Transfer Protocol

FTP (*File Transfer Protocol*) utilise TCP pour transférer des fichiers entre deux systèmes au travers d'un réseau IP. L'implémentation met en jeu deux composants : le composant client et le composant serveur. La Figure 13.17 explique le mode opératoire de FTP.

Figure 13.17
Mode opératoire de FTP
(source : Learning Tree).



- **Connexion de contrôle**
 - Créée quand la connexion au serveur FTP est établie
 - Utilisée seulement pour les commandes/réponses FTP
- **Connexion de transfert de données**
 - Créée à la demande pour chaque transfert de données
 - Rompue à la fin de chaque transfert de données

Le client FTP s'exécute sur la machine de l'utilisateur et initie une connexion avec le serveur FTP. Le serveur FTP est en écoute sur le port 21, où il attend les requêtes de connexion. Lorsqu'une requête de connexion est reçue, un échange en trois étapes a lieu afin d'établir la connexion TCP. Cette connexion TCP est appelée la *connexion de contrôle*, et sert à échanger les commandes FTP et les réponses. Lorsque le client FTP appelle une commande de démarrage de transfert de données, le client lance un processus de transfert de fichiers côté client sur un port TCP local. Ce numéro de port local est transmis au serveur FTP en utilisant la connexion de



contrôle. Lorsque le serveur FTP reçoit le numéro de port du processus de transfert de données, le serveur lance un processus de transfert de données sur le port 20, qui émet une requête de connexion TCP sur le port indiqué par le client FTP. Cette connexion est appelée *connexion de données*, et ne sert que pour transférer les données des fichiers demandés. FTP permet d'envoyer vers et de recevoir depuis la machine hôte. Les fichiers sont envoyés sur la connexion de données. En résumé, les deux connexions FTP suivantes sont créées pendant une session FTP :

- connexion de contrôle (TCP, FTP_CLIENT_IP, CLIENT_PORT1, FTP_SERVER_IP, 21) ;
- connexion de données (TCP, FTP_CLIENT_IP, CLIENT_PORT2, FTP_SERVER_IP, 20).

La connexion de données n'existe que pendant la durée du transfert de fichiers. Elle est refermée dès que le transfert de données est terminé. Une nouvelle connexion de données est établie chaque fois qu'on doit transférer un fichier. La connexion FTP reste ouverte jusqu'à ce que l'utilisateur ou le serveur FTP referme la connexion. En général, la session FTP est terminée par l'utilisateur FTP, mais un serveur FTP peut très bien choisir de fermer une connexion, par exemple en cas de problème de charge trop importante.

Dans le mode opératoire de FTP, le processus de données est créé hors du processus principal. Cependant, sur les plates-formes ne pouvant pas créer de multiples processus, ou lorsque le FTP est exécuté dans un système intégré sans lien avec le système d'exploitation, FTP alors est un programme monolithique implémentant les différents composants FTP. Quels que soient les environnements d'exécution du client FTP et du serveur FTP, le protocole FTP requiert deux connexions TCP : une pour la connexion de contrôle, l'autre pour la connexion de données.

Le standard FTP est décrit dans la RFC 959 (STD 9), intitulée "*File Transfer Protocol*". Les sections suivantes décrivent la structure des commandes FTP et en donnent des exemples d'utilisation.

Commandes FTP

Les commandes de contrôle d'une connexion FTP utilisent le même format de données que celui d'une session Telnet. Rappelez-vous que Telnet avait recours à NVT pour formater les données. FTP n'exploite pas toutes les possibilités du format NVT, puisqu'il occulte les phases de négociation.

Les commandes de contrôle FTP sont envoyées sous la forme de chaînes de quatre caractères suivies éventuellement d'arguments. Ces commandes et leur signification sont reprises dans le Tableau 13.13.

Tableau 13.13 : Les commandes de contrôle FTP

Commande de contrôle	Fonctionnalité
USER nomutil	Spécifie le nom de l'utilisateur initiant la session FTP.
PASS motdepasse	Spécifie le mot de passe de l'utilisateur indiqué par la commande USER.
ACCT compte	Spécifie des informations supplémentaires de compte après exécution correcte de la commande PASS. Cette commande est facultative.
CWD repertoire	Permet de changer le répertoire de travail sur le serveur distant.
CDUP	Remonte d'un répertoire sur le serveur distant.
SMNT chemin	Demande de monter un autre système de fichiers sans modifier le login ou le mot de passe.
REIN	Réinitialise la session. La prochaine commande devra être USER.
QUIT	Met fin à la session FTP.
PORT h1, h2, h3, h4, p1, p2	Spécifie les informations concernant le point d'arrivée TCP. Utilise pour identifier auprès du serveur FTP le numéro de port de processus de données du client. h1, h2, h3 et h4 sont les nombres en notation décimale à point de l'adresse IP du client. p1 et p2 correspondent à la notation décimale à point du numéro de port de processus de données.
PASV	Spécifie que le serveur FTP doit se mettre en écoute de la connexion de données qui sera transmise par le client FTP. La réponse à cette commande est le port TCP sur lequel écoute le serveur.
TYPE code	Spécifie le type de représentation des données.
STRU code	Spécifie la structure de fichier, par exemple file, record, ou page.
MODE code	Spécifie le mode de transfert, par exemple stream, block ou compressed.
RETR chemin	Récupération du fichier désigné.
STO chemin	Envoi du fichier désigné.
STOU chemin	Identique à STO si ce n'est que le fichier qui sera créé aura un nom unique dans le répertoire.



Tableau 13.13 : Les commandes de contrôle FTP (*suite*)

Commande de contrôle	Fonctionnalité
APPE chemin	Identique à STO si ce n'est que dans le cas où le fichier existe déjà sur le serveur, le contenu du nouveau fichier lui sera ajouté.
ALLO argument	Permet d'allouer de l'espace pour le fichier sur le serveur.
REST argument	Permet de spécifier au serveur à partir de quel octet il faut reprendre un transfert interrompu.
RNFR chemin	Spécifie l'ancien chemin à renommer. Doit être suivie de la commande RNT0.
RNT0 chemin	Spécifie le chemin du nouveau fichier. Doit être précédée de la commande RNFR.
ABOR	Demande au serveur FTP l'annulation de la dernière commande.
DELE chemin	Demande la suppression du fichier spécifié.
RMD repertoire	Demande la suppression du répertoire spécifié.
MKD repertoire	Demande la création du répertoire spécifié.
PWD	(<i>Print Working Directory</i>). Renvoie le nom du répertoire de travail courant sur le serveur FTP.
LIST [repertoire]	Demande la liste des fichiers et de leurs attributs présents dans le répertoire indiqué, ou par défaut dans le répertoire courant.
NLST [repertoire]	Demande la liste des noms de fichiers du répertoire indiqué (ou du répertoire courant). Les autres attributs de fichiers ne sont pas renvoyés. Cette commande est utilisée dans l'implémentation de la commande MGET.
SITE parametres	Permet de spécifier des commandes spécifiques au site du serveur FTP.
SYST	Permet de connaître la plate-forme du serveur FTP.
STAT [chemin]	Permet de renvoyer des informations d'état sur les fichiers ou le transfert FTP.
HELP [commande]	Permet d'obtenir de l'aide sur une commande particulière, ou bien par défaut la liste des commandes implémentée sur le serveur.
NOOP	Pas d'opération, le serveur FTP renvoie une réponse.

En réponse aux commandes FTP, le serveur FTP renvoie des codes de trois chiffres suivis d'un espace et de texte optionnel, comme suit :

NNN Texte du message

NNN représente la réponse sur trois chiffres.

Voici l'un des retours possibles du serveur :

200 Command okay.

Le Tableau 13.14 donne la liste des codes de réponse courants accompagnés de leur signification.

Tableau 13.14 : Codes de réponse d'un serveur FTP

200 Command okay.

500 Syntax error, command unrecognized.

501 Syntax error in parameters or arguments.

202 Command not implemented, superfluous at this site.

502 Command not implemented.

503 Bad sequence of commands.

504 Command not implemented for that parameter.

110 Restart marker reply.

211 System status, or system help reply.

212 Directory status.

213 File status.

214 Help message.

215 NAME system type.

120 Service ready in nnn minutes.

220 Service ready for new user.

221 Service closing control connexion.

421 Service not available, closing control connexion.

125 Data connexion already open; transfer starting.

225 Data connexion open; no transfer in progress.



Tableau 13.14 : Codes de réponse d'un serveur FTP (suite)

425	Can't open data connexion.
226	Closing data connexion.
426	Connection closed; transfer aborted.
227	Entering Passive Mode (h1,h2,h3,h4,p1,p2).
230	User logged in, proceed.
530	Not logged in.
331	User name okay, need password.
332	Need account for login.
532	Need account for storing files.
110	Restart marker reply.
120	Service ready in nnn minutes.
125	Data connexion already open; transfer starting.
150	File status okay; about to open data connexion.
200	Command okay.
202	Command not implemented, superfluous at this site.
250	Requested file action okay, completed.
257	"PATHNAME" created.
350	Requested file action pending further information.
450	Requested file action not taken.
451	Requested action aborted: local error in processing.
452	Requested action not taken.
550	Requested action not taken.
551	Requested action aborted: page type unknown.
552	Requested file action aborted.
553	Requested action not taken.

FTP du point de vue de l'utilisateur

FTP permet à un utilisateur d'accéder aux fichiers et répertoires d'un hôte distant de façon interactive et d'y effectuer un certain nombre d'opérations :

- Lister les fichiers d'un répertoire local ou distant.

- Renommer et supprimer des fichiers.
- Transférer des fichiers depuis l'hôte distant vers l'hôte local (*download*).
- Transférer des fichiers depuis l'hôte local vers l'hôte distant (*upload*).

En règle générale, vous devez disposer d'un client FTP pour utiliser FTP. Ces applications TCP/IP clientes sont disponibles sur un grand nombre de plates-formes. Certaines plates-formes disposent d'une interface graphique, d'autres n'ont que la ligne de commande. D'un point de vue purement pédagogique, il est recommandé d'utiliser l'interface en ligne de commande qui permet de mieux comprendre la correspondance entre les commandes FTP de l'utilisateur et les commandes FTP internes du Tableau 13.13.

Pour lancer une session FTP, tapez la commande suivante :

```
ftp [nom_hote]
```

`nom_hote` est le nom symbolique ou l'adresse IP de l'hôte auquel vous désirez vous connecter. Si vous omettez de le préciser, vous aurez quand même accès à certaines commandes FTP, notamment les commandes d'aide, accessibles en utilisant un point d'interrogation. Le listing suivant vous montre les commandes disponibles une fois que vous êtes connecté sur l'hôte INTERNIC.NET :

```
ftp> ?
```

Commands may be abbreviated. Commands are:

!	cr	macdef	proxy	send
\$	delete	mdelete	sendport	status
account	debug	mdir	put	struct
append	dir	mget	pwd	sunique
ascii	disconnect	mkdir	quit	tenex
bell	form	mls	quote	trace
binary	get	mode	recv	type
bye	glob	mput	remotehelp	user
case	hash	nmap	rename	verbose
cd	help	ntrans	reset	?
cdup	lcd	open	rmdir	
close	ls	prompt	runique	

Si vous désirez effectuer une connexion FTP avec l'hôte INTERNIC.NET, tapez la commande suivante :

```
ftp INTERNIC.NET
```



Si vous connaissez l'adresse IP de l'hôte, vous pouvez l'indiquer directement :

```
ftp 137.65.4.1
```

Il est important de comprendre que vous devez préciser un nom d'utilisateur et un mot de passe afin de vous connecter sur un hôte FTP. Le serveur FTP utilise les mécanismes sous-jacents d'identification de sa plate-forme pour vérifier les privilèges de chaque utilisateur FTP. Dès lors, si vous disposez d'un compte Bill sur la machine faisant office d'hôte FTP, vous pouvez vous connecter en tant que Bill et utiliser votre mot de passe habituel.

La plupart des hôtes FTP proposent un accès anonyme. Si vous spécifiez le nom d'utilisateur anonymous et que vous donnez comme mot de passe votre adresse e-mail, ou guest, vous pouvez vous connecter à l'hôte FTP avec des droits limités, déterminés par l'administrateur système.

L'exemple suivant est une visite guidée d'une session FTP.

1. Saisissez la commande ftp en précisant le nom de l'hôte visé.

```
% ftp internic.net
```

Le % est le prompt Unix par défaut, vous n'avez pas à le taper.

2. Si l'hôte est accessible, vous verrez apparaître un message ressemblant à ceci :

```
Connected to internic.net.
220-*****Welcome to the InterNIC Registration Host *****
*****Login with username "anonymous" and password "guest"
*****You may change directories to the following:
policy      - Registration Policies
templates   - Registration Templates
netinfo     - NIC Information Files
domain      - Root Domain Zone Files
220 And more!
Name (internic.net:karanjit):
```

Comme l'indique ce message, vous pouvez vous connecter en utilisant le nom d'utilisateur anonymous et le mot de passe guest.

3. Donnez un nom d'utilisateur et un mot de passe

```
Name (internic.net:karanjit): anonymous
331 Guest login ok, send "guest" as password.
Password: guest
230 Guest login ok, access restrictions apply.ftp>
```

4. Une fois connecté, vous pouvez utiliser la commande ? (ou help) :

```
ftp>?
```

Il est possible d'abrégier les commandes, dont la liste sera donnée un peu plus loin dans cette section.

5. La commande permettant de savoir dans quel répertoire vous êtes est `pwd`.

```
ftp> pwd
257 "/" is current directory.
```

L'état de chaque commande FTP est renvoyé sous forme d'un code numérique, par exemple 257, accompagné d'un message texte.

6. Afin de consulter la liste des fichiers du répertoire courant, utilisez la commande `ls` ou `dir`.

```
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connexion for file list.
bin
usr
dev
etc
pub
policy
templates
home
netinfo
domain
ls-ltR
netprog
archives
rfc
226 Transfer complete.
99 bytes received in 0.04 seconds (2.4 Kbytes/s)
```

La liste précédente donne juste le nom des fichiers et non leurs attributs supplémentaires (taille, fichier ou répertoire, date, etc.). Pour voir ces informations, utilisez la commande `dir` comme suit :

```
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connexion for /bin/ls.
total 22
drwxr-xr-x 2 root 1    512 Mar 22 21:40 archives
dr-xr-xr-x 2 root 1    512 Feb 25 1993 bin
```




```

drwxr-xr-x 2 root 1      512 Mar 9 1993 dev
drwxr-xr-x 2 root 1      512 Apr 1 1993 domain
dr-xr-xr-x 2 root 1      512 Feb 25 1993 etc
drwxr-xr-x 2 root 1      512 Mar 9 1993 home
-rw-r--r-- 1 root 1     9035 May 4 19:12 ls-ltR
drwxr-xr-x 2 root 1     1024 Apr 14 14:57 netinfo
drwxr-xr-x 2 root 1      512 Apr 1 1993 netprog
drwxr-xr-x 2 root 1     1024 May 4 19:11 policy
drwxr-xr-x 4 root 1      512 Apr 20 15:01 pub
lrwxrwxrwx 1 root 1       6 Aug 9 1993 rfc -> policy
drwxr-xr-x 2 root 1      512 May 3 19:54 templates
drwxr-xr-x 3 root 1      512 Feb 25 1993 usr
226 Transfer complete.
875 bytes received in 0.25 seconds (3.4 Kbytes/s)

```

Ces informations sont affichées dans un style Unix puisque vous êtes connectés à un système Unix.

7. Si vous êtes certain d'être connecté à un système Unix, vous pouvez utiliser l'astuce suivante :

Si vous utilisez la commande `ls`, la commande Unix `ls` est appelée. Vous pouvez donc lui adjoindre toutes les options habituelles, notamment l'option `-lR`, qui donne un listing sous forme longue et parcourant l'intégralité des sous-répertoires de façon récursive. Attention, il ne s'agit pas ici d'une commande ou d'une option FTP, mais bien d'une spécificité de la plate-forme sur laquelle s'exécute le serveur FTP :

```

ftp> ls -lR
200 PORT command successful.
150 Opening ASCII mode data connexion for /bin/ls.
total 22
drwxr-xr-x 2 root 512 Mar 22 21:40 archives
dr-xr-xr-x 2 root 512 Feb 25 1993 bin
drwxr-xr-x 2 root 512 Mar 9 1993 dev
drwxr-xr-x 2 root 512 Apr 1 1993 domain
dr-xr-xr-x 2 root 512 Feb 25 1993 etc
drwxr-xr-x 2 root 512 Mar 9 1993 home
-rw-r--r-- 1 root 9035 May 4 19:12 ls-ltR
drwxr-xr-x 2 root 1024 Apr 14 14:57 netinfo
drwxr-xr-x 2 root 512 Apr 1 1993 netprog
drwxr-xr-x 2 root 1024 May 4 19:11 policy
drwxr-xr-x 4 root 512 Apr 20 15:01 pub
lrwxrwxrwx 1 root 6 Aug 9 1993 rfc -> policy
drwxr-xr-x 2 root 512 May 3 19:54 templates

```

```

drwxr-xr-x 3 root  512 Feb 25 1993 usr
:
(etc.)
:
usr/lib:
total 576
-r-xr-xr-x 1 root  40960 Feb 25 1993 ld.so
-rwxr-xr-x 1 root  516096 Feb 25 1993 libc.so.1.8
-rwxr-xr-x 1 root  24576 Feb 25 1993 libdl.so.1.0
226 Transfer complete.
remote: -lR
9228 bytes received in 1.2 seconds (7.5 Kbytes/s)

```

8. Pour changer de répertoire, utilisez la commande `cd` :

```

ftp> cd rfc
250 CWD command successful.

```

9. Pour voir la liste des fichiers du répertoire `/rfc`, tapez `ls` ou `dir`, comme suit :

```

ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connexion for file list.
asn.index
domain.index
index
master.index
network.index
rfc1009.txt
rfc1011.txt
rfc1031.txt
:
(etc.)
:
226 Transfer complete.
496 bytes received in 0.045 seconds (11 Kbytes/s)

```

10. Si vous essayez de récupérer un fichier qui n'existe pas, par exemple `rfc1365.txt`, FTP vous l'indiquera de la façon suivante :

```

ftp> get rfc1365.txt
200 PORT command successful.
550 rfc1365.txt: No such file or directory.

```



11. Pour copier un fichier depuis le serveur FTP vers la machine locale, tapez la commande FTP suivante :

```
get fichierdistant [fichierlocal]
```

Dans cette commande, `fichierdistant` est le nom du fichier de l'hôte distant et `fichierlocal` est le nom du fichier sur la machine locale. Si vous ne spécifiez pas `fichierlocal`, le fichier local aura le même nom que le fichier distant.

Pour les fichiers de texte, FTP sait effectuer les conversions nécessaires sur les fins de lignes entre les différentes plates-formes, à condition que vous ayez demandé un transfert en mode ASCII. Pour demander ce mode de transfert, utilisez la commande FTP `ascii`. Pour transférer des fichiers binaires, utilisez la commande `image` ou `binary` afin de désactiver la conversion des fins de ligne.

L'exemple suivant illustre la commande FTP `get` :

```
ftp> get rfc1400.txt
200 PORT command successful.
150 Opening ASCII mode data connexion for rfc1400.txt (13009
bytes).
226 Transfer complete.
local: rfc1400.txt remote: rfc1400.txt
13404 bytes received in 0.78 seconds (17 Kbytes/s)
```

12. Pour refermer la connexion FTP, utilisez la commande `close` :

```
ftp> close
221 Goodbye.
ftp>
```

13. Pour quitter complètement FTP, tapez la commande `bye` qui fermera les connexions avant de quitter FTP.

```
ftp> bye
%
```

• Trivial File Transfer Protocol

TFTP (*Trivial File Transfer Protocol*) utilise UDP comme protocole de transport. Comme UDP ne garantit pas la bonne remise des données, TFTP doit effectuer sa propre correction d'erreurs. TFTP utilise une simple technique de PAR (*Positive Acknowledgement Retransmission*). TFTP définit des messages d'acquiescement utilisés pour signaler la bonne réception des données. Si le message d'acquiescement n'est pas parvenu dans un certain délai, le paquet de données est réémis.

TFTP a été conçu dans un esprit de simplicité et de compacité d'implémentation. La taille du code permet d'implémenter ce protocole dans les ROM d'amorçage des stations de travail. TFTP est souvent utilisé en conjonction avec BOOTP qui permet d'obtenir des informations de configuration en provenance d'un serveur BOOTP. TFTP sert ensuite à télécharger un système d'exploitation.

Le standard TFTP est défini par la RFC 1350, "*The TFTP Protocol (Revision 2)*" et complété par les RFC 1782, 1783, 1784 et 1785.

Les sections suivantes passent en revue le format des messages TFTP, puis son mode opératoire.

Format des messages TFTP

TFTP définit cinq types de messages présentés dans le Tableau 13.15.

Tableau 13.15 : Types de messages TFTP

Code d'opération	Description
1	Requête de lecture (RRQ)
2	Requête d'écriture (WRQ)
3	Données (DATA)
4	Acquittement (ACK)
5	Erreur (ERROR)

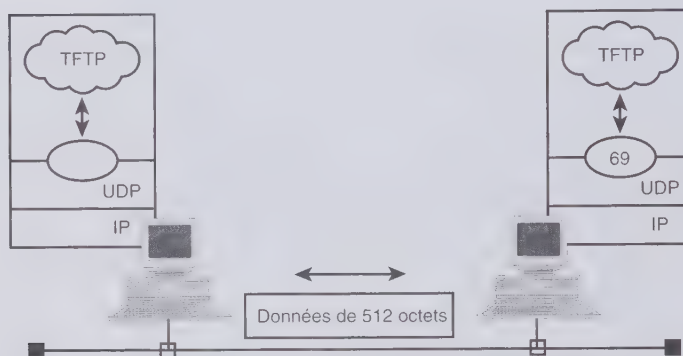
Le client TFTP envoie tout d'abord une requête de lecture (RRQ) ou d'écriture (WRQ), comme illustré dans la Figure 13.18, sur le port UDP 69 où le serveur TFTP est en attente de requêtes.

Le format des requêtes de lecture et d'écriture est repris dans la Figure 13.19. Tous les messages TFTP disposent d'un champ OpCode (Code d'opération) contenant l'une des valeurs du Tableau 13.15. Les messages de requête de lecture et d'écriture contiennent aussi des champs pour le nom de fichier et le mode opératoire. Le champ de nom de fichier est une chaîne de caractères à zéro terminal. Le champ de mode opératoire est utilisé pour indiquer les types de fichiers reconnus. Ses valeurs sont NETASCII, BINARY et MAIL, sous la forme de constantes chaînes à zéro terminal. Le format à zéro terminal de ces deux champs leur permet d'être de longueur variable.



Figure 13.18

Interaction client/serveur
TFTP (source : Learning Tree).

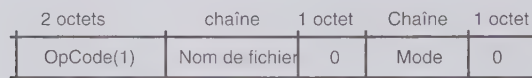


- **Trivial File Transfer Protocol (TFTP)**

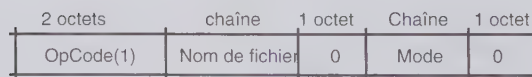
- S'exécute au-dessus de UDP et est plus simple que TCP
 - TFTP fournit sa propre gestion des erreurs
- Ne fournit pas de service d'identification
- Facilement mis en œuvre en mémoire morte (ROM)
 - Idéal pour les stations de travail sans disque

Figure 13.19

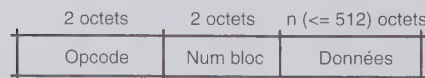
Format des messages TFTP.



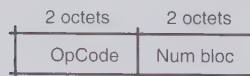
(a) paquet RRQ



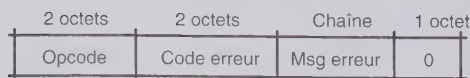
(b) paquet WRQ



(c) paquet DATA



(d) paquet ACK

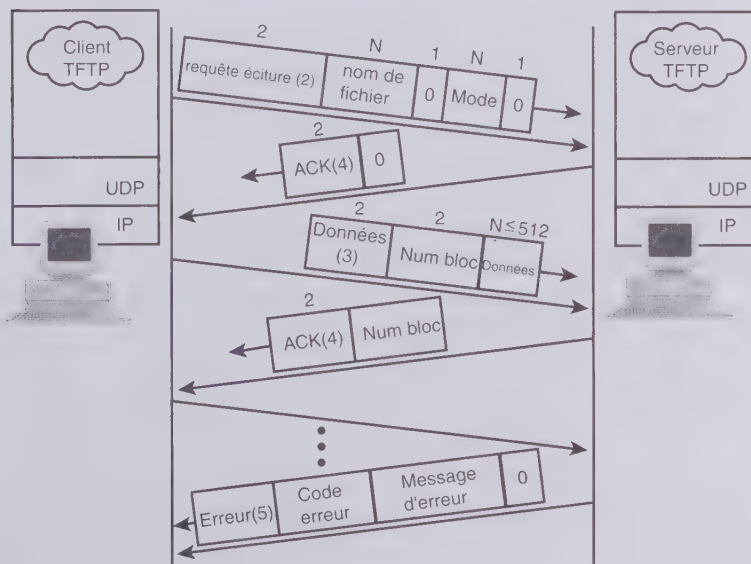


(e) paquet ERROR

Mode opératoire de TFTP

Après réception d'une requête de lecture ou d'écriture, le serveur TFTP enregistre l'adresse IP et le numéro de port du client. Dans les réponses qu'il lui fera parvenir par la suite, le serveur TFTP utilisera cette adresse IP et ce numéro de port. Le message de requête d'écriture ou de lecture est acquitté et suivi par des messages de données et leurs acquittements. TFTP transmet ses messages par blocs de 512 octets. Un bloc de données de moins de 512 octets signale la fin de la transmission. Lorsque la taille du fichier est un multiple de 512, un message de données supplémentaire est émis, de taille nulle. Les blocs de données sont numérotés en séquence. Les messages de données et d'acquiescement sont accompagnés d'un champ contenant le numéro de bloc des données transférées. La Figure 13.20 montre une transmission TFTP dans laquelle un message de requête d'écriture a été envoyé par le client TFTP.

Figure 13.20
Exemple de transaction TFTP
(source : Learning Tree).



- Seuls 5 types de messages TFTP sont définis
- Un bloc de données de taille < 512 octets indique la fin des données



TFTP peut être utilisé pour transférer des fichiers entre hôtes sans authentification. Les fichiers sont transférés en spécifiant uniquement leur nom. Comme il n'est pas nécessaire de disposer d'un nom d'utilisateur et d'un mot de passe pour transférer des fichiers avec TFTP, la plupart des administrateurs système le désactivent sur leurs systèmes ou restreignent les types de fichiers susceptibles d'être ainsi transférés. Les implémentations peuvent aussi refuser l'accès à un fichier à moins que tous les utilisateurs de l'hôte y aient accès.

Network File System

NFS (*Network File System*) est un protocole développé à l'origine par SUN Microsystems et vendu sous licence à un grand nombre d'éditeurs. NFS permet à un ordinateur exécutant le programme serveur NFS d'exporter son système de fichiers vers d'autres clients. Exporter un système de fichiers revient à le rendre accessible aux clients d'un grand nombre de plates-formes, pourvu qu'ils exécutent le programme client NFS.

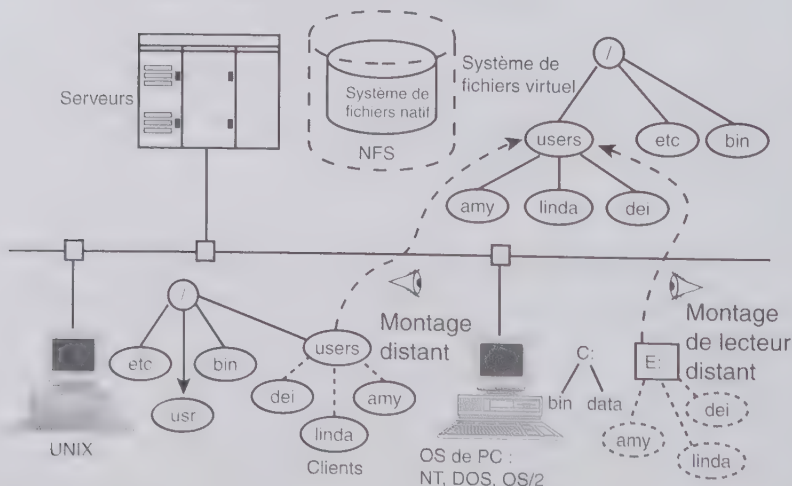
La Figure 13.21 montre que le serveur NFS exporte le répertoire `/users`. Ce répertoire exporté peut être accédé simultanément par des clients utilisant n'importe quel système d'exploitation.

Figure 13.21

Utilisation de NFS
(source : *Learning Tree*).

- **NFS permet de créer un système de fichiers virtuel**

- Généralement exécuté sur les plates-formes UNIX
- Non limité à UNIX
- VMS, MVS, NetWare, DOS
- Connexion sans état entre deux ordinateurs



Chaque client NFS voit le système de fichiers exporté par le serveur NFS comme s'il faisait partie de son propre système de fichiers. Par exemple, un client NFS PC DOS accédera au système de fichiers par le biais d'un lecteur réseau, et un client NFS Unix verra le système de fichiers distant rattaché à son propre système de fichiers.

Les sections suivantes aborderont plus en détail les différentes versions de NFS ainsi que les protocoles afférents.

Versions de NFS

NFS se compose d'un certain nombre de protocoles agissant de concert pour implémenter un système de fichiers. NFS a été développé par SUN Microsystems au début des années 80. Il est documenté dans la RFC 1094, "NFS: Network File System Protocol Specification".

Le protocole NFS original, tel que documenté par la RFC 1094, était conçu pour utiliser UDP comme protocole de transport. Ce protocole a été utilisé dans les versions 1 et 2 de NFS. L'utilisation de UDP est bien adaptée aux réseaux locaux, mais présente certains inconvénients dans les réseaux étendus. NFS version 2 savait utiliser TCP, mais à un coût élevé en termes de performances.

Les derniers développements de NFS, version 3, lui permettent de fonctionner avec TCP. NFS version 3, conçu pour fonctionner avec TCP, est plus adapté à l'utilisation sur des réseaux étendus, susceptibles de tomber en panne et d'avoir des temps de réponse très variables. NFS version 3 a supprimé la limite arbitraire de 8 Ko maximum par bloc imposée par NFS version 2.

NFS version 3 est indispensable pour utiliser Web NFS, un nouveau protocole permettant aux navigateurs Web d'accéder à des serveurs NFS. NFS version 3 est documenté dans la RFC 1813, "NFS Version 3 Protocol Specification".

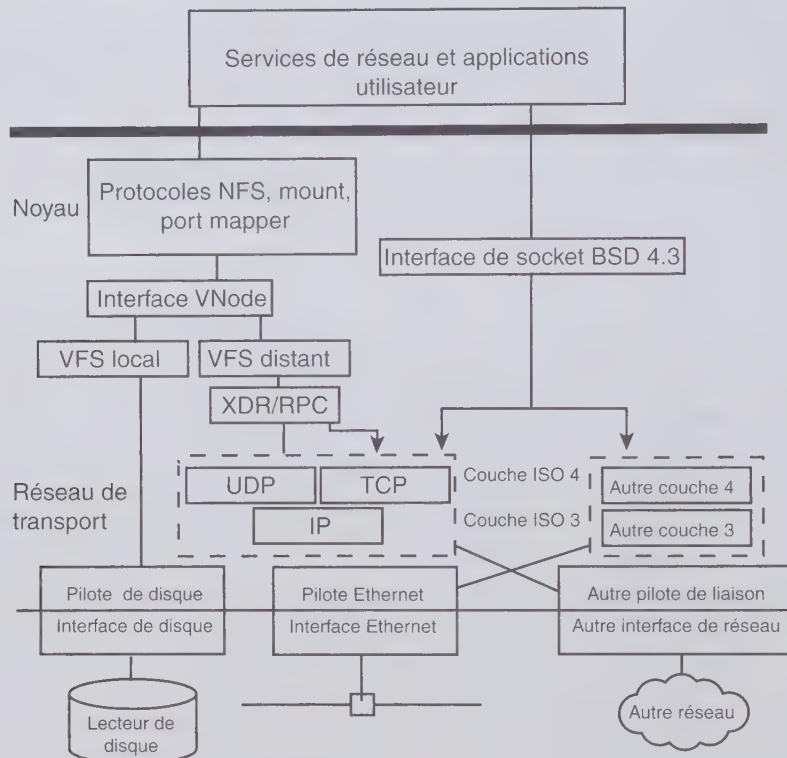
Les protocoles NFS

Le service d'application NFS est implémenté par le biais d'un ensemble de protocoles NFS. La Figure 13.22 montre ces différents protocoles. Au niveau des couches Liaison de données et Physique du modèle OSI, NFS sait tirer parti des différentes technologies disponibles. Au niveau de la couche réseau, NFS utilise IP. La couche Transport est assurée par UDP ou TCP. Lorsque NFS est utilisé sur UDP, il est recommandé d'utiliser les totaux de contrôle facultatifs (*checksum*) de UDP.



Figure 13.22

Les protocoles NFS
(source : Learning Tree).



Au niveau de la couche Session du modèle OSI, NFS utilise le protocole RPC (*Remote Procedure Call*). Il existe différents protocoles RPC, mais tous portent le même nom. Le protocole RPC de NFS est souvent appelé Sun-RPC pour éviter de le confondre avec les autres. RPC permet aux services NFS du serveur d'être utilisés grâce au paradigme d'appel de procédure, familier des développeurs. RPC fournit un accès de haut niveau aux services NFS sans rentrer dans le détail des protocoles de communication. Les développeurs utilisant RPC n'ont pas à connaître le détail des communications pour utiliser les services réseaux rendus disponibles par RPC. Le protocole Sun-RPC est présenté dans la RFC 1057, "RPC: Remote Procedure Call Protocol Specification version 2".

Au niveau de la couche Présentation du modèle OSI, NFS utilise le protocole XDR (*External Data Representation*). XDR met à disposition une technique permettant de représenter les données de façon uniforme. Par exemple, les données numériques sont représentées en utilisant une notation de complément à 2. Lorsqu'un système utilise une représentation différente des nombres, la conversion

est automatiquement effectuée par XDR. Le protocole XDR est traité dans la RFC 1832, "XDR: External Data Representation Standard".

Au niveau de la couche application du modèle OSI, NFS utilise le protocole NFS (*Network File System*). Ce protocole permet d'écrire dans un fichier, de créer un fichier, de lire un fichier, etc. La couche application se compose aussi d'un ensemble de protocoles de support pour NFS, dont les protocoles *Mount* et *Portmapper*. Le protocole Mount implémente la procédure de montage NFS, et le protocole Portmapper donne aux clients ayant besoin d'un service un numéro de port pour ledit service. Le protocole Portmapper utilise le port UDP 111. Un client NFS désireux d'accéder à un service particulier envoie une requête sur le port UDP 111. Les programmes serveurs désirant mettre à disposition un service s'enregistrent auprès de Portmapper. Lorsqu'ils reçoivent une requête pour un service, Portmapper répond en donnant le numéro de port auquel on doit transmettre les requêtes.

RPC NFS : appel de procédure distante

La communication NFS entre un client et un serveur utilise le modèle RPC (*Remote Procedure Call*, ou appel de procédure distante). NFS implémente 18 appels de procédures distantes. Le code de ces 18 procédures réside sur le serveur NFS. Les procédures NFS sont utilisées par les clients NFS pour accéder aux services réseau de fichiers implémentés par le serveur NFS. Un client NFS accède à ces procédures en utilisant le mécanisme de RPC.

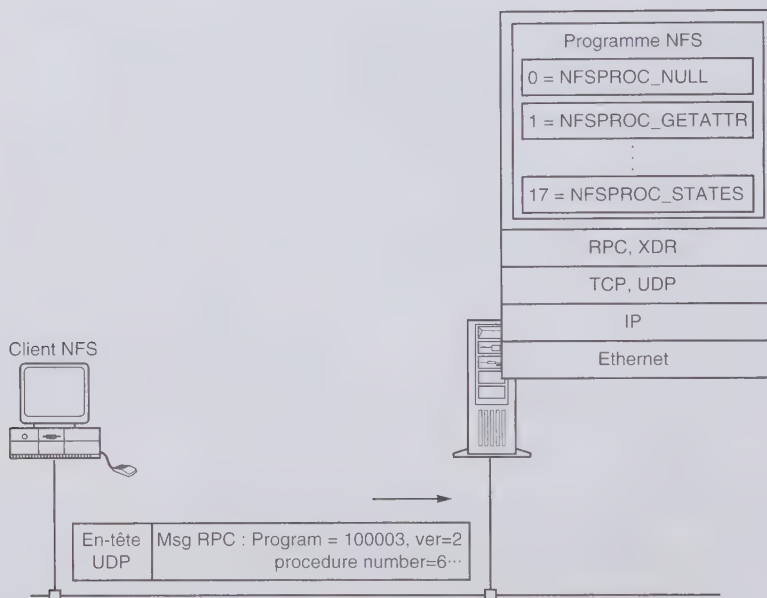
Le protocole RPC contient un numéro de programme et un numéro de procédure. Le numéro de programme identifie le service auquel on désire accéder. Par exemple, le service NFS a le numéro de programme 100003. A l'intérieur du programme, on trouve 18 procédures numérotées de 0 à 17. Par exemple, la procédure numéro 6 traite de la lecture d'un fichier. Un client NFS ayant besoin de lire un fichier enverra une requête RPC contenant le numéro de programme 100003 et le numéro de procédure 6 (voir Figure 13.23). Les autres paramètres de l'opération de lecture seront fournis par le paquet de requête RPC émanant du client.

La requête RPC spécifie aussi le numéro de version du programme devant mettre le service à disposition. Tous les programmes proposant des services comportent un numéro de version. Ce numéro de version permet de faire cohabiter de façon efficace les différentes versions d'un même service. Cette coexistence permet aux clients NFS plus anciens d'appeler le service précédent, cependant que les nouveaux clients feront appel à la nouvelle implémentation.



Figure 13.23

Exemple d'appel RPC.



Le Tableau 13.16 dresse la liste des numéros de procédures du programme NFS et donne une brève description de leurs fonctionnalités. La procédure numéro 0, NFS_PROC, est décrite comme ne faisant rien. Dans tous les programmes, la procédure numéro 0 ne fait rien. En fait, elle fait un peu plus que cela. Lorsqu'une requête RPC est transmise à la procédure numéro 0, elle se contente de renvoyer la requête à l'expéditeur. Bon nombre de clients NFS utilisent cette fonctionnalité d'écho à des fins de diagnostic. Il existe d'ailleurs un utilitaire de diagnostic appelé `nfsping`, permettant d'envoyer une requête d'écho à la procédure 0 et d'attendre une réponse. La syntaxe de `nfsping` est la suivante :

```
nfsping nom_hôte
```

Si le service NFS est actif sur l'hôte, on recevra une réponse. Sinon, le service (ou l'hôte) est inaccessible.

Tableau 13.16 : Numéros des RPC de NFS

Numéro de procédure	Procédure NFS	Description
0	NFS_PROC	Pas d'action, sert à des fins de diagnostic.
1	NFS_GETATTR	Demande les attributs d'un fichier ou d'un répertoire, notamment les permissions.

Tableau 13.16 : Numéros des RPC de NFS (*suite*)

Numéro de procédure	Procédure NFS	Description
2	NFSPROC_SETATTR	Définit les attributs d'un fichier ou d'un répertoire, notamment les permissions.
3	NFSPROC_ROOT	Obsolète ; remplacée par le protocole Mount.
4	NFSPROC_LOOKUP	Demande un descripteur de fichier dans un répertoire spécifique.
5	NFSPROC_READLINK	Lit le contenu d'un lien symbolique.
6	NFSPROC_READ	Lit un fichier.
7	NFSPROC_WRITECACHE	Utilisé par NFS version 3 pour écrire dans le cache.
8	NFSPPROC_WRITE	Ecrit dans un fichier.
9	NFSPROC_CREATE	Crée un fichier.
10	NFSPROC_REMOVE	Supprime une entrée de répertoire dans un répertoire.
11	NFSPROC_RENAME	Renomme un fichier.
12	NFSPROC_LINK	Crée un lien physique vers un fichier.
13	NFSPROC_SYMLINK	Crée un lien symbolique vers un fichier.
14	NFSPROC_MKDIR	Crée un répertoire.
15	NFSPROC_RMDIR	Supprime un répertoire.
16	NFSPROC_READDIR	Lit un répertoire.
17	NFSPROC_STATFS	Demande les attributs du système de fichiers.

Web NFS

Le protocole Web NFS a été conçu pour permettre à un navigateur Web d'accéder à des serveurs NFS à travers l'Internet. Le protocole Web NFS implique l'utilisation de NFS version 3, qui accepte les connexions TCP et utilise des descripteurs publics.

En NFS, un *descripteur* est une structure de données interne, créée par le serveur NFS lorsqu'une requête Mount est faite en utilisant le protocole NFS Mount. Le protocole NFS Mount dépend de RPC pour la communication entre le client NFS



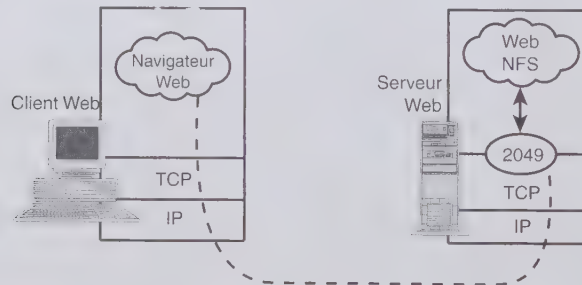
et le serveur. Le descripteur est renvoyé lorsque le daemon Mount (un daemon est un terme Unix représentant un processus indépendant) qui s'exécute sur le serveur NFS monte le système de fichiers demandé. Le processus de montage comprend notamment une vérification des droits de l'utilisateur et une initialisation des structures de données, comportant des informations indiquant comment accéder au système de fichiers demandé. Le descripteur renvoyé est ensuite utilisé pour toutes les opérations ultérieures visant à accéder au système de fichiers.

Un descripteur public est un descripteur de fichier système spécialement réservé et utilisé comme descripteur initial de système de fichiers par les navigateurs Web. L'utilisation d'un descripteur public permet au navigateur Web d'accéder au système de fichiers d'un serveur NFS sans avoir à utiliser la commande NFS Mount.

La Figure 13.24 donne un exemple d'utilisation de Web NFS. Le serveur Web NFS écoute sur le port TCP 2049 les requêtes en provenance des clients Web.

Figure 13.24

Web NFS (source : Learning Tree).



Les protocoles Web NFS sont documentés dans les RFC 2054 ("Web NFS Client Specification") et 2055 ("Web NFS Server Specification").

Protocoles d'accès Internet

Un certain nombre de protocoles ont évolué afin de donner accès à des documents sur l'Internet. Les deux plus célèbres sont HTTP (*HyperText Transfer Protocol*) et Gopher. Parmi ces protocoles, le plus répandu est HTTP, aussi connu sous le nom de protocole WWW.

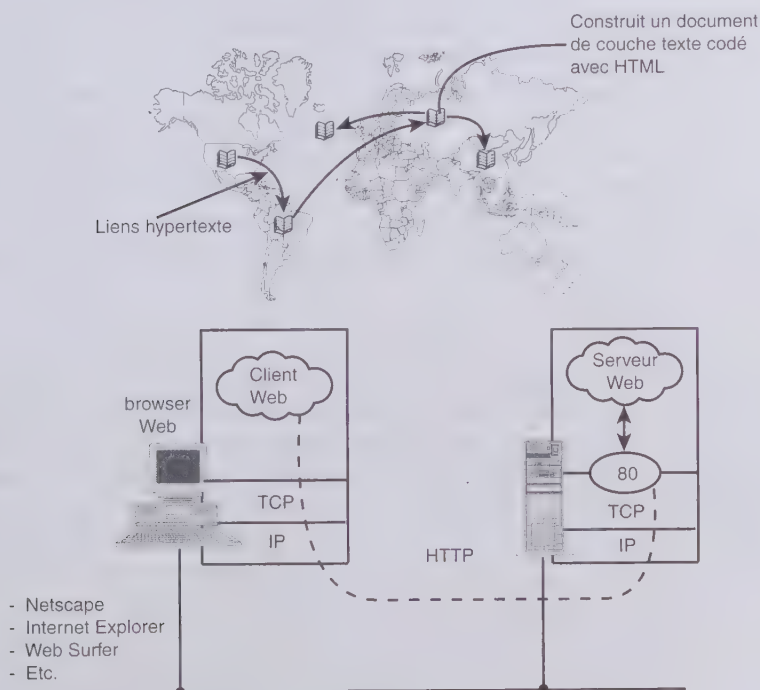
- **WWW (World Wide Web)**

La Figure 13.25 illustre l'utilisation du WWW. Le WWW, souvent appelé le Web, est composé d'un grand nombre de serveurs Web, lesquels stockent des documents contenant des informations de texte, des images, des sons et de la vidéo, organisés

sous la forme de *pages Web*. Chaque page Web contient des instructions appelées des hyperliens permettant d'y rattacher des documents Web. Les hyperliens peuvent pointer sur des documents situés sur le même serveur ou sur des documents situés sur d'autres serveurs de l'Internet. L'ensemble résultant de documents ressemble à un labyrinthe de pages pouvant s'étendre de pays en pays et de continent en continent, d'où le terme de World Wide Web (réseau maillé de taille planétaire).

Figure 13.25

Le World Wide Web
(source : Learning Tree).



Les documents Web sont consultables à partir de logiciels clients appelés des navigateurs Web. Le document est encodé en utilisant un langage de description de pages (HTML). Le navigateur Web commence par charger un document HTML, puis l'affiche graphiquement à l'écran. Le code HTML est une description des différents composants de la page Web que le navigateur Web utilise pour créer un rendu graphique des différents composants de la page. Les hyperliens prennent souvent l'apparence de textes soulignés. Lorsque vous cliquez sur ces textes, le navigateur Web télécharge le document correspondant.



Les sections suivantes aborderont l'encodage HTML et le protocole HTTP.

HTML (*HyperText Markup Language*)

HTML est une implémentation de SGML (*Standard Generalized Markup Language*). Le standard SGML, aussi appelé ISO-8879 par l'ISO, fournit une technique générique permettant de spécifier une représentation des documents structurés, incluant éventuellement des hyperliens. Un *hyperlien* est une phrase (ou une suite de mots) dans un document, mise en évidence, permettant de provoquer le chargement d'un nouveau document lorsque on la sélectionne. Les actions que l'on peut effectuer lorsqu'un hyperlien est sélectionné sont les suivantes : afficher une image, envoyer un courrier électronique, demander des informations à l'utilisateur par le biais d'un formulaire, lancer un téléchargement, interroger une base de données, exécuter un programme, etc. Les documents comportant des hyperliens sont parfois appelés des hyperdocuments.

Dans tous les documents HTML se trouvent des balises avec la structure suivante :

```
<balise>
</balise>
```

où balise vaudra l'un des mots clés utilisés pour décrire les composants d'un document HTML. La balise fermante `</balise>` marque la fin de la portée de la balise ouvrante `<balise>`. Presque toutes les balises HTML ont une balise fermante correspondante. Par exemple, les documents HTML sont toujours encadrés par les mêmes balises :

```
<HTML>
Les différents éléments du document HTML
apparaîtront entre ces balises.
</HTML>
```

Entre les balises `<HTML>` et `</HTML>`, vous pourrez spécifier l'en-tête et le corps du document HTML. L'en-tête vient entre les balises `<HEAD>` et `</HEAD>`, le corps entre les balises `<BODY>` et `</BODY>` :

```
<HTML>
<HEAD> En-tête du document </HEAD>
<BODY> Corps du document </BODY>
</HTML>
```

Les lignes vides placées entre les balises ne seront pas prises en compte. Par exemple, le code HTML précédent pourrait sans perte d'information être écrit comme suit :

```
<HTML><HEAD> En-tête du document </HEAD><BODY> Corps du document
</BODY>
</HTML>
```


Les lignes vides sont en général ajoutées pour augmenter la lisibilité du document HTML, tout comme on ajoute des commentaires dans les programmes. Les balises peuvent indifféremment être en minuscules ou en majuscules. Si vous désirez couper un paragraphe de texte ou ajouter une ligne vide visible dans le document final sur la machine du client, vous devrez ajouter des balises HTML spécifiques permettant d'obtenir ce rendu. Le titre du document est défini dans l'en-tête, en utilisant les balises <TITLE> et </TITLE>. Le texte compris entre ces balises apparaîtra dans la zone de titre du navigateur, comme suit :

```
<HTML>
<HEAD>
<TITLE>
    Ma première page HTML !
</TITLE>
</HEAD>
<BODY> Corps du document. </BODY>
</HTML>
```

Pour créer un hyperlien, utilisez la balise d'ancrage :

```
<A HREF= adresse_URL autres_parametres> texte de l'hyperlien </A>
```

La balise d'ancrage peut s'accompagner d'un certain nombre de paramètres spécifiés dans *autres_parametres*, qui modifient le comportement de la balise d'ancrage. Généralement, le paramètre *HREF* correspond à l'URL (*Uniform Resource Locator*) de la ressource appelée. Une adresse URL est une technique standard permettant de localiser une ressource sur l'Internet (syntaxe d'une URL : **protocole://nom_hote/chemin**).

Protocole peut être l'un des protocoles utilisés sur l'Internet, par exemple HTTP, FTP, Telnet, Gopher, File, etc., permettant d'accéder à la ressource. Si vous indiquez un document HTML résidant sur un autre ordinateur, le protocole que vous spécifierez sera HTTP (*HyperText Transfer Protocol*), utilisé pour récupérer un document HTML. Pour accéder à un autre document en spécifiant le protocole FTP, vous indiquerez FTP, etc. Si vous accédez à un fichier sur l'ordinateur local (c'est-à-dire, l'ordinateur sur lequel s'exécute le navigateur Web), vous pouvez spécifier le protocole FILE.

Nom_hote est l'adresse IP ou le nom de domaine de l'hôte auquel vous désirez accéder. **chemin** est le chemin d'accès à la ressource, à savoir le répertoire et le nom du fichier auquel vous désirez accéder. **chemin** peut être sensible à la différence majuscules/minuscules si le serveur Web fonctionne sous Unix. **chemin** est facultatif. Si vous ne le précisez pas dans le cas d'un document HTML, le serveur



vous renverra un document par défaut, dont le nom est souvent **index.html** (ce comportement varie en fonction du serveur Web). Voici quelques exemples d'adresses URL :

http://www.novell.com

http://www.hp.com/peripherals

http://www.tsl.org

ftp://ftp.novell.com

telnet://usa.net

file:///C:/PUBS/MONWEB/simple.html

Examinez le document HTML suivant, qui contient un lien vers différentes ressources. Notez qu'il est possible d'insérer des commentaires dans un document HTML en utilisant la syntaxe `<!-- commentaire>`.

```
<HTML>
<HEAD><TITLE>Balises d'ancrage</TITLE></HEAD>
<BODY>
<!-- Commentaire HTML. Notez la syntaxe employée.>
<!--
<H1>Démonstration des balises d'ancrage</H1>
<p>Les balises d'ancrage permettent de créer des hyperliens.
Remarquez comment nous avons commencé ce paragraphe, en
utilisant une balise de paragraphe. Les balises de paragraphe
formatent automatiquement un texte ou un paragraphe. Lorsque
vous terminez une ligne, vous devez générer un saut pour séparer
le paragraphe du texte qui suit.
<br> <!-- Cette balise insère un saut de ligne.>
Les balises d'ancrage peuvent avoir un grand nombre de
paramètres, mais nous ne passerons en revue que les plus simples.
<hr><!-- Cette balise crée une ligne horizontale>
<p> Voici les balises d'ancrage:
<p><A HREF=http://www.novell.com>Serveur Web de Novell</A>
<p><A HREF=ftp://www.novell.com> Serveur FTP de Novell </A>
<p><A HREF=telnet://usa.net>Telnet vers USA.NET</A>
<p><A HREF=mailto:tslinfo@tsl.org>Ecrivez-nous </A>
</BODY>
</HTML>
```

En étudiant le code HTML et son rendu dans un navigateur Web, vous comprendrez mieux le sens des différentes balises HTML.

Dans le document HTML précédent, la balise de paragraphe `<p>` a été utilisée pour séparer les hyperliens en les mettant chacun sur une ligne. HTML dispose de

mécanismes permettant d'afficher des listes numérotées ou non numérotées pour représenter des listes d'éléments. Les listes ordonnées affichent un numéro devant chaque nouvel élément de la liste, et les listes non ordonnées mettent une simple puce devant chaque élément. La syntaxe des listes est la suivante :

```
<!-- Liste ordonnée>
<OL>
    <LI> Élément 1
    <LI> Élément 2
    :
    <LI> Élément N
</OL>
<!-- Liste non ordonnée>
<UL>
    <LI> Élément 1
    <LI> Élément 2
    :
    <LI> Élément N
</UL>
```

Le code suivant montre comment insérer les liens dans une liste ordonnée. Remarquez qu'on a défini deux balises d'ancrage supplémentaires. Le paramètre HREF contient une constante chaîne précédée du symbole #. Ce lien se réfère à un ancrage interne à la page, défini par une balise d'ancrage interne. Cette balise d'ancrage interne dispose d'un attribut NAME dont la valeur est la constante chaîne évoquée ci-avant.

```
<HTML>
<HEAD><TITLE>Balises d'ancrage</TITLE></HEAD>
<BODY>
<!-- Commentaire HTML. Notez la syntaxe employée.>
<!--
<H1>Démonstration des balises d'ancrage</H1>
<p>Les balises d'ancrage permettent de créer des hyperliens.
Remarquez comment nous avons commencé ce paragraphe, en
utilisant une balise de paragraphe. Les balises de paragraphe
formatent automatiquement un texte ou un paragraphe. Lorsque
vous terminez une ligne, vous devez générer un saut pour séparer
le paragraphe du texte qui suit.
<br> <!-- Cette balise insère un saut de ligne.>
Les balises d'ancrage peuvent avoir un grand nombre de
```



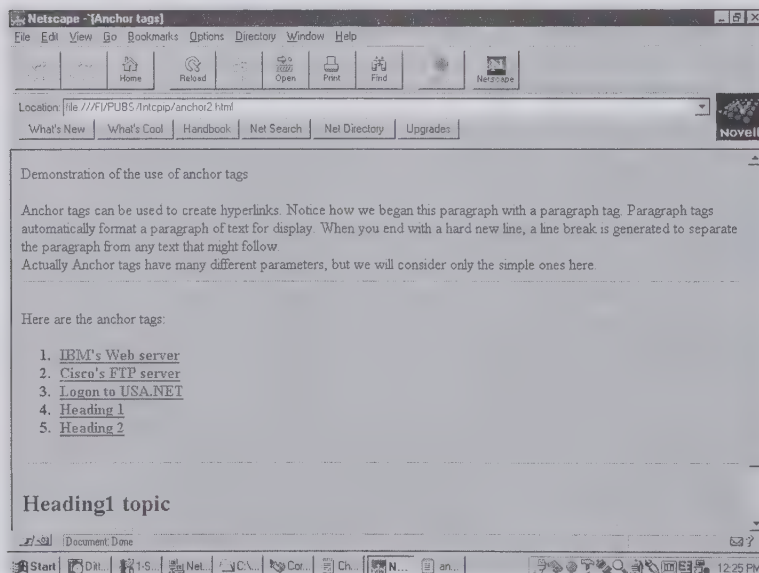
```

paramètres, mais nous ne passerons en revue que les plus simples.
<hr><!-- Cette balise crée une ligne horizontale>
<p> Voici les balises d'ancrage:
<B> <!-- Affichage en gras jusqu'à ce qu'on rencontre /B>
<OL><!-- Liste Ordonnée>
<LI><A HREF=http://www.ibm.com>Serveur Web d'IBM </A>
<LI><A HREF=ftp://ftp.cisco.org>Serveur FTP de Cisco </A>
<LI><A HREF=telnet://usa.net>Telnet vers USA.NET</A>
<!-- Les deux ancrages suivants amènent à l'emplacement indiqué
dans le document courant>
<LI><A HREF="#Lien1">Lien 1</A>
<LI><A HREF="#Lien2">Lien 2</A>
</OL><!-- Fin de la liste ordonnée>
</B><!-- Fin du gras>
<hr>
<!-- Les niveaux de titre vont de H1 à H6>
<A NAME="Lien1"><H2>Sujet du lien 1 </H2></A>
<A NAME="Lien2"><H2>Sujet du lien 2</H2></A>
</BODY>
</HTML>

```

Figure 13.26

Rendu d'un document HTML
dans un navigateur Web.



La Figure 13.26 montre le rendu final du code HTML dans un navigateur Web. Vous remarquerez l'utilisation des balises et afin de mettre le texte en gras. De la même façon, <I> et </I> permettent de mettre un texte en italique. Vous disposez enfin de six niveaux de titres, de <H1> à <H6>, afin de faire varier l'apparence et la taille des titres.

HyperText Transfer Protocol

Les commandes des navigateurs Web et les documents HTML envoyés en réponse sont transmis par le biais d'un protocole particulier appelé HTTP (*HyperText Transfer Protocol*). Le serveur Web est souvent appelé un serveur HTTP (ou un daemon HTTP, sous Unix), parce qu'il répond à des requêtes HTTP.

Lorsqu'un navigateur Web demande un document HTML, une connexion est établie entre le navigateur Web et le serveur Web, sur le port TCP 80. Vous pouvez spécifier un autre port, par exemple le port TCP 8080, mais le numéro de port par défaut est 80.

Si vous désirez utiliser un port non standard, il faudra le préciser dans l'URL. Voici une adresse URL demandant à accéder au serveur via le port 8080 : **http://www.someorg.com:8080**.

Une fois la connexion établie, le navigateur Web envoie une requête de document par l'intermédiaire du protocole HTTP. La requête contient la méthode à utiliser pour récupérer le document, le nom de l'objet demandé et la version du protocole HTTP à invoquer. Le protocole HTTP communique ses informations en format texte afin de ne pas être gêné par les différences d'implémentation des jeux de caractères d'une plate-forme à l'autre. Voici, par exemple, une requête HTTP envoyée par un navigateur Web demandant un document :

```
GET /Web/docs/index.html HTTP/1.0
```

Dans cette requête, la requête GET HTTP est utilisée pour obtenir le document /Web/docs/index.html. Le numéro de version du protocole HTTP est 1.0.

Le serveur répond à la requête en envoyant une réponse HTTP. Cette réponse du serveur se compose de trois parties :

- état de la réponse ;
- en-tête de réponse ;
- données.

L'état de la réponse est une ligne de texte avec la version de HTTP utilisée par le serveur, un code d'état décrivant le résultat du serveur, et une description texte



donnant quelques informations supplémentaires. Voici un exemple de statut de réponse renvoyé par un serveur Web :

```
HTTP/1.0 200 OK
```

Vient ensuite l'en-tête de réponse, qui contient des informations relatives au type du serveur, à la version de MIME utilisée pour décrire le type de contenu, et une ligne vide. Cette ligne vide est obligatoire, car elle sert de séparateur entre l'en-tête de réponse et les données qui suivent. Voici un exemple d'en-tête de réponse :

```
Date: Tuesday, 05-Nov-96 23:42:34 GMT
Server: Novell-HTTP-Server/2.5
MIME-version: 1.0
Content type:text/html
```

Ici les données de la réponse (notez la présence d'une ligne vide)

MIME est une méthode d'encodage des données conçue au départ pour transmettre des documents par courrier électronique. HTTP utilise MIME pour décrire les différents types de données qu'il doit manipuler. La ligne `Content type` est une description MIME des données transférées. `text/html` indique que les données sont de type texte et de sous-type HTML. Lorsque le navigateur reconnaît ce type texte, il sait qu'il doit interpréter les données comme du code HTML. Un autre type couramment reconnu par les navigateurs est le type `text/plain`, qui indique que la réponse est à traiter comme du texte brut.

Si le document demandé est un document HTML, les données de réponse contiendront le texte du document HTML. Voici un exemple de données de réponse :

```
<HTML>
<HEAD><TITLE>Document HTML simple</TITLE></HEAD>
<BODY>
<P> Cela est un document HTML minimaliste.
</BODY>
</HTML>
```

En assemblant les différents morceaux de la réponse, voici ce qui est effectivement transmis depuis le serveur jusqu'au navigateur Web :

```
HTTP/1.0 200 OK
Date: Tuesday, 05-Nov-96 23:42:34 GMT
Server: Novell-HTTP-Server/2.5
MIME-version: 1.0
Content type:text/html
```

```

<HTML>
<HEAD><TITLE>Document HTML simple</TITLE></HEAD>
<BODY>
<P> Cela est un document HTML minimaliste.
</BODY>
</HTML>

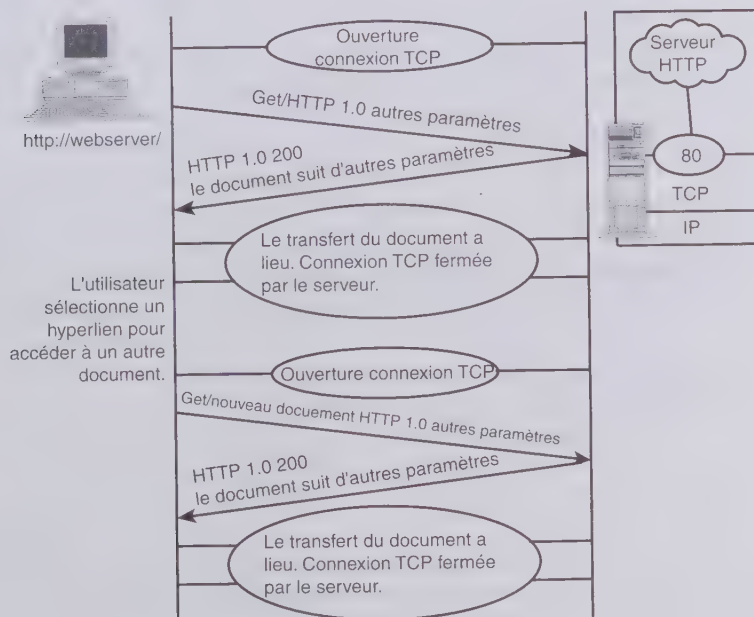
```

Une fois la séquence requête/réponse achevée, la connexion TCP/IP est refermée. Une nouvelle connexion sera établie pour chaque échange requête/réponse entre le client et le serveur.

La Figure 13.27 montre une interaction HTTP typique. Remarquez que la première étape consiste à ouvrir une connexion TCP sur le port 80 du serveur Web. Ensuite, la commande GET/HTTP 1.0 est transmise par le client Web. Le transfert du document a alors lieu. A la fin de ce transfert, la connexion est refermée par le serveur HTTP. Chaque fois que le client demandera un nouveau document, une nouvelle connexion TCP sera établie.

Figure 13.27

Interaction HTTP classique
(source : Learning Tree).



• Les paramètres HTTP sont codés en MIME.



Afin de comprendre comment fonctionne le protocole HTTP, tentez de contacter le serveur Web sur le port 80. Si vous avez accès à un client Telnet, vous pouvez essayer la commande suivante :

```
telnet hote_Web 80
```

Remplacez hote_Web par l'adresse IP ou le nom de machine du serveur Web de NetWare. Le nombre 80 indique au client Telnet d'établir la connexion sur le port TCP 80 du serveur Web. Sans cette indication explicite, Telnet tenterait de se connecter sur le port 23. Vous pouvez utiliser Telnet depuis n'importe quelle machine Unix ou depuis un client Telnet Windows. Pensez dans ce dernier cas à indiquer le port 80 dans la fenêtre de configuration de connexion.

Une fois la connexion effectuée, vous verrez apparaître le message suivant :

```
Trying 199.245.180.11...
Connected to 199.245.180.11
Escape character is '^['.
```

Cet exemple suppose que l'adresse IP du serveur est 199.245.180.11. Votre correspondant aura probablement une autre adresse.

Nous allons maintenant simuler le comportement d'un client Web émettant une requête HTTP GET. Saisissez la ligne suivante afin de récupérer le fichier index.htm :

```
GET /index.html HTTP/1.0
```

Vous remarquerez qu'il ne s'est rien passé lorsque vous avez appuyé sur la touche Entrée. En effet, la requête HTTP doit se terminer par deux séquences <CR><LF>. Appuyez donc à nouveau sur la touche Entrée. Vous verrez apparaître la réponse HTTP quasi immédiatement. Voici un exemple de réponse, en provenance du serveur Web expérimental développé par l'auteur de cet ouvrage :

```
HTTP/1.0 200 OK
Date: Sat, 02 Aug 1997 04:17:19 GMT
Server: Novell-HTTP-Server/2.5
Content type: text/html
Last-modified: Thu, 29 Feb 1996 14:26:58 GMT
Content-length: 2761
```

```
<HTML><TITLE>NetWare Web Server CGI SDK Home Page</TITLE>
<BODY TEXT="#000000" LINK="#FF0000">
<BODY BACKGROUND=images/blue_pap.gif>
<IMG SRC="/images/novlogo.gif" align="right" hspace=10> <P>
<H2>Novell NetWare Web Server</H2>
```

This NetWare Web Server product contains the CGI SDK that allows programmers to extend the functionality of the NetWare Web Server.

<P>

This is the sample home page installed on your Early Access Web Server. You can edit this file or replace it with any other HTML document. This file can be found in your document root as <I>index.htm</I>.

<HR>

<H2>About this Web server...</H2>

This Web page is served from a NetWare File server running HTTP.NLM.

The CGI implementation is beta software, and is provided to allow for developers to begin development of extensions to the NetWare Web Server.

<p>

Please send e-mail to the address at the bottom of this page to let us know that you're developing applications to the NetWare Web Server, and for any technical support issues you might have.

<hr>

Try out these three examples of applications that are written to the NetWare CGI. The SDK provides the source code for the

<I>cgiparse</I>

and <I>cginph</I> applications that you can use as development templates.

<P>

1. LoadableModule /nds/ sys:Web/nds-bin/

<DD>http://localhost/nds/ndsobj

<P>

2. LoadableModule /cgi/ sys:Web/cgi-bin

<DD>http://localhost/cgi/cgiparse

<DD><I>Use as a template for parsed-header extensions.</I><P>

3. LoadableModule /cgi/ sys:Web/cgi-bin

<DD>http://localhost/cgi/cginph

<DD><I>Use as a template for non-parsed header extensions.</I><P>

<hr>

The following is an implementation of an image map processor written to the NetWare CGI. The NetWare Web Server implements image maps natively, but this demonstrates that you can implement your own processing if you wish.

<p>



Try clicking on the pillar, a fish's eye, the plant, etc.

<p>

<HR>

<H2>Known problems</H2>

Sometimes there may be some problems when you UNLOAD the HTTP.NLM. The System Console may hang for a bit before unloading.

Also, sometimes unloading HTTP.NLM does not successfully unload your CGI extensions. If this happens, manually unload the extensions before trying to unload HTTP.NLM again.

<P>

More examples and code will be provided with later updates. Be sure to send us email to the address below to get on our update list!

<HR>

<ADDRESS>Novell, Inc., San Jose, CA

<i>nwwebdev@novell.com</i>

</ADDRESS>

</BODY>

</HTML>

Remarquez que la réponse HTTP contient des informations d'en-tête, une ligne vide, puis le contenu du document demandé. Ce document pourra varier en fonction de la machine que vous avez appelée. A la fin du transfert du document HTML, la connexion Telnet est automatiquement refermée. Vous serez obligé d'établir une nouvelle connexion si vous demandez un nouveau document en utilisant HTTP 1.0. Cette façon de procéder est assez mal adaptée à des pages contenant un grand nombre d'images. En effet, chaque élément d'une page doit être téléchargé séparément et nécessite donc d'établir une connexion particulière. Le protocole HTTP est en cours de révision, et la version 1.1 règle ce problème.

Dans l'essai précédent, nous avons demandé un document présent sur le serveur. Que se passe-t-il lorsque la ressource est inexistante ? Vérifions-le en établissant à nouveau une connexion Telnet avec le serveur HTTP, mais en saisissant la requête suivante :

```
GET /je_suis_presque_sur_que_cette_page_n_existe_pas.htm HTTP/1.0
```

Validez en tapant deux fois sur la touche Entrée, vous devriez obtenir cette réponse :

```
HTTP/1.0 404 Not Found
```

```
Date: Sat, 02 Aug 1997 04:18:29 GMT
```



```
Server: Novell-HTTP-Server/2.5
Content type: text/html
```

```
<HEAD><TITLE>404 Not Found</TITLE></HEAD>
<BODY><H1>404 Not Found</H1>
The requested URL /
je_suis_presque_sur_que_cette_page_n_existe_pas.htm was not
found on this server.<P>
</BODY>
```

Remarquez que la réponse HTTP contient un document HTML indiquant que la ressource demandée n'a pas été trouvée sur le serveur. Le code de retour est 404, qui indique que la ressource n'a pas été trouvée. Le document HTML renvoyé par le serveur est un *document virtuel*, qui n'existe pas physiquement sur le serveur : il a été créé dynamiquement par le serveur Web pour indiquer la nature de l'erreur.

Vous voudrez sans doute tenter d'envoyer d'autres commandes, par exemple une requête invalide, afin de voir le type de réponse renvoyée par le serveur :

GETX requete non valide

La réponse du serveur indique un code d'erreur 400, signifiant que la requête HTTP était non valide :

```
HTTP/1.0 400 Bad Request
Date: Sat, 02 Aug 1997 04:20:09 GMT
Server: Novell-HTTP-Server/2.5
Content type: text/html

<HEAD><TITLE>400 Bad Request</TITLE></HEAD>
<BODY><H1>400 Bad Request</H1>
Your client sent a query that this server could not
understand.<P>
Reason: Invalid or unsupported method.<P>
</BODY>
```

Indexation Web et passerelles CGI

Afin de trouver des documents spécifiques sur le serveur, un certain nombre de moteurs de recherche sont à votre disposition. Ces moteurs de recherche indexent les documents Web, suivent tous les hyperliens d'un document HTML, et indexent l'intégralité des documents ainsi obtenus. Certains d'entre eux sont réalisés et

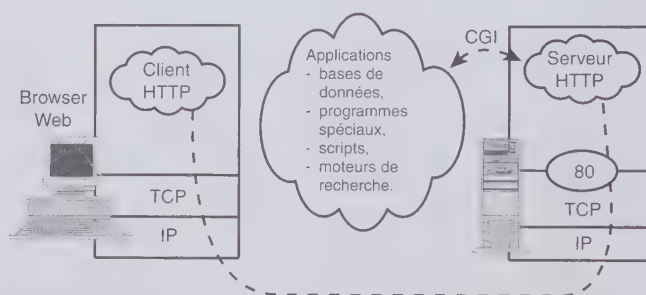


maintenus par des entreprises indépendantes. La plupart des navigateurs Web disposent de méthodes permettant d'accéder directement à ces moteurs de recherche. Vous pourrez en trouver aux adresses www.yahoo.fr, www.excite.com, et www.lycos.com.

La plupart de ces moteurs de recherche emploient l'interface CGI (*Common Gateway Interface*), qui permet d'exécuter d'autres services d'application (voir Figure 13.28). CGI est utilisé pour transférer la requête à une autre application située sur le serveur Web ou sur un autre serveur. Le résultat de cette exécution est renvoyé au serveur par le biais de CGI, qui le renvoie à son tour au client.

Figure 13.28

CGI (source : Learning Tree).



Gopher

Bien que la majorité des informations nouvelles de l'Internet soient publiées sur des serveurs Web, il existe des informations tout aussi utiles, mais plus anciennes, accessible uniquement par les vieux outils tels que FTP et Gopher. Gopher a perdu de sa popularité au fur et à mesure du déploiement du Web.

Le protocole Gopher a été développé à l'Université du Minnesota et fournit un système de menus par l'intermédiaire duquel on peut naviguer à la recherche de sujets d'intérêt. Dans Gopher, les hyperliens sont les titres des menus. Une fois un titre de menu sélectionné, vous accédez à d'autres menus ou à d'autres documents, situés sur le même serveur ou sur un autre. L'ensemble de ces menus liés entre eux s'appelle *l'espace gopher*.

La Figure 13.29 montre les interactions entre un client Gopher et un serveur Gopher. Le serveur Gopher écoute sur le port TCP 70 et attend des chaînes de recherche de texte envoyées par le client. Le serveur Gopher renvoie des éléments de menus en fonction de la dernière chaîne de recherche texte.

La Figure 13.30 montre un écran de menus renvoyé à un client Gopher.

Un outil spécial, appelé Veronica (*very easy rodent-oriented networkwide index to computerized archives*) est associé à Gopher pour rechercher des sujets précis dans les menus des serveurs Gopher. Veronica est souvent accessible sous la forme d'un choix de menu dans les menus de premier niveau du système de menus Gopher.

Figure 13.29

Le modèle client/serveur
de Gopher (source : Learning
Tree).

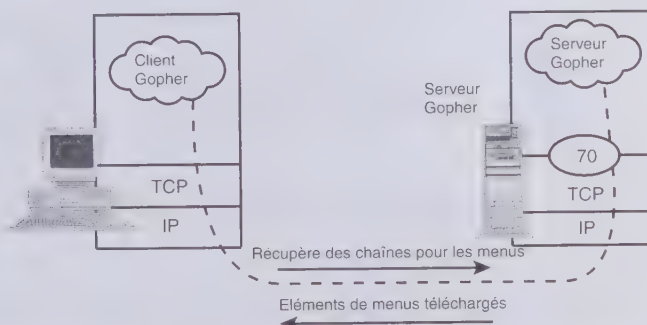
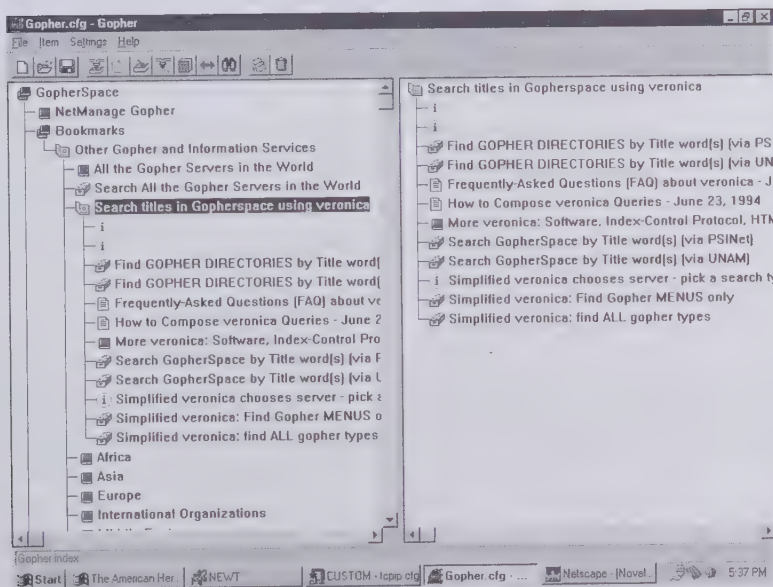


Figure 13.30

Exemple de menu Gopher
(source : Learning Tree).

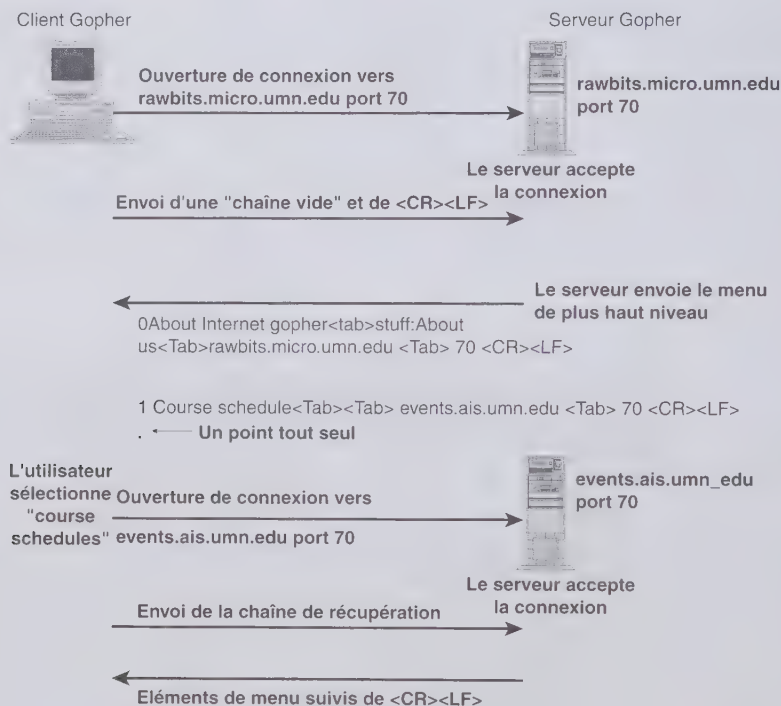


Une session Gopher est lancée par le client Gopher, qui émet une connexion TCP sur le port 70 du serveur Gopher (voir Figure 13.31). Le serveur Gopher accepte la liaison, et une connexion TCP est établie. Le client Gopher envoie ensuite une chaîne vide (""), terminée par un retour chariot et un saut de ligne. Le serveur Gopher répond à cette chaîne en envoyant les éléments de menu de son menu de premier niveau. Ces éléments sont envoyés selon le format indiqué dans la Figure 13.32.



Figure 13.31

Mode opératoire de Gopher
(source : Learning Tree).



Le champ sélecteur (voir Figure 13.32) peut prendre l'une des valeurs définies dans le Tableau 13.17. Le champ chaîne d'affichage est l'élément de menu qui sera affiché par le client Gopher. Le champ référence de fichier est une chaîne spéciale qui doit être envoyée au serveur Gopher afin de récupérer les éléments associés à la sélection courante. Les champs nom d'hôte et numéro de port hôte définissent le serveur Gopher TCP et permettent de trouver le fichier spécifié.

Figure 13.32

Format des éléments
de menu Gopher.

Sélecteur	<Tab>	Chaîne d'affichage	<Tab>	Référence de fichier	<Tab>	Nom d'hôte	<Tab>	Numéro de port hôte
-----------	-------	-----------------------	-------	-------------------------	-------	---------------	-------	------------------------

<Tab> = Table de caractères ASCII

Le protocole Gopher est décrit dans la RFC 1436.

Récapitulatif

Ce chapitre a abordé les services d'application utilisant l'infrastructure de protocole TCP/IP : DNS, les protocoles de courrier (SMTP, POP3 et IMAP4), les protocoles d'accès distant (Telnet et utilitaires Berkeley r*), les protocoles de transfert de fichiers (FTP et TFTP), les protocoles d'accès à des fichiers (NFS et Web NFS) et les protocoles d'accès au Web (HTTP/HTML, Gopher).

Les services d'application implémentent les fonctionnalités des couches 5, 6 et 7 du modèle OSI. Ces services d'application concernent le formatage, l'envoi et la réception de données d'application. Ils ne portent pas sur les interactions avec l'utilisateur, gérées par des applications spécifiques.

Testez vos connaissances

1. Quelle est la fonction de DNS ?
2. Qu'est-ce qu'un résolveur de noms ?
3. Pourquoi DNS utilise-t-il UDP au lieu de TCP ?
4. Dessinez un diagramme montrant la nature client/serveur de DNS, le service DNS s'exécutant sur le serveur DNS et les numéros des ports utilisés par DNS.
5. Quelle est l'implémentation de DNS la plus courante ?
6. En règle générale, à quoi correspondent les désignations à deux lettres des noms de domaines de premier niveau ?
7. Faut-il prêter attention aux majuscules et minuscules lorsqu'on tape un nom de domaine ? Y a-t-il une limitation à la longueur d'un nom de domaine ?
8. Donnez quelques domaines de premier niveau DNS.
9. Pourquoi affecter plus d'un serveur DNS à un domaine de premier niveau ?
10. L'entreprise SITA SARL dispose d'un hôte dont le nom de domaine est ramu.eng.sita.com. En supposant que cette entreprise dispose de serveurs DNS pour les domaines eng.sita.com et sita.com, discutez de l'ordre des



Tableau 13.17 : Valeurs du champ sélecteur

Valeur	Type d'élément
0	Fichier
1	Répertoire
2	service CSO Phone Book
3	Erreur
4	Fichier Macintosh BinHex
5	archive DOS binaire
6	fichier Unix uuencodé
7	Serveur de recherche indexée
8	Session Telnet en mode texte
9	Fichier binaire ; le client doit lire jusqu'à la fermeture de la connexion TCP
+	Serveur redondant
T	Session tn3270 en mode texte
g	Fichier au format GIF
I	fichier image ; le client décide du mode d'affichage.

requêtes adressées aux serveurs DNS. Votre réponse devra mentionner ce qui se passe lorsque les serveurs sont incapables de résoudre la requête.

11. SMTP peut-il transmettre des données binaires ? Si oui, comment ?
12. Décrivez le format d'une adresse 822.
13. Comment MIME gère-t-il l'affichage de messages multimédia sur des terminaux texte ?
14. Lorsqu'un document est envoyé à un grand nombre d'adresses e-mail en utilisant une liste de diffusion, comment peut-on, grâce à MIME, minimiser la quantité d'informations envoyées ?
15. Décrivez les différences entre un UA mail et un MTA mail.
16. Un utilisateur d'adresse e-mail **kss@hoopy.frood.com** envoie un message de trois lignes de texte à un utilisateur d'adresse **veena@scs.sing.org**. Décrivez la séquence de commandes et de réponses SMTP nécessaires pour transférer ce message.
17. Un utilisateur d'adresse e-mail **kss@hoopy.frood.com** envoie un message de trois lignes de texte aux utilisateurs d'adresse **veena@scs.sing.org** **anita**

@scs.sing.org. et linda@scs.sing.org. Décrivez la séquence de commandes et de réponses SMTP nécessaires pour transférer ce message.

18. Pourquoi SMTP n'est-il pas adapté à la réception de courrier sur une station de travail ? Quels sont les protocoles adaptés à cette tâche ?
19. Dessinez un diagramme montrant l'architecture client/serveur de POP3 et les numéros de port utilisés par POP3.
20. Si POP3 est utilisé sur une station de travail, l'est-il pour la réception des messages, l'envoi des messages ou les deux ? Quels sont, s'il en faut, les autres protocoles requis ?
21. Un utilisateur utilise POP3 pour récupérer trois messages e-mail ayant les tailles de message suivantes :

ID Message	Taille
1	121
2	300
3	290

Les messages mail ne sont pas supprimés après avoir été téléchargés. Décrivez la séquence de commande POP3 client et de réponses du serveur qui auront lieu pendant la réalisation de cette opération.

Vous supposerez pour ce faire que le nom d'utilisateur est belinda et que le mot de passe est magic.

22. Un utilisateur utilise POP3 pour récupérer quatre messages e-mail ayant les tailles de message suivantes :

ID Message	Taille
1	345
2	443
3	4425
4	7450

Les messages sont supprimés après téléchargement. Décrivez la séquence de commande POP3 client et de réponses du serveur qui auront lieu pendant la réalisation de cette opération.



Vous supposerez pour ce faire que le nom d'utilisateur est **linda** et que le mot de passe est **saralfont**.

23. Décrivez les quatre étapes d'une session POP3 classique.
24. Quelle est la plus grande faiblesse de POP3, résolue par IMAP4 ?
25. Dessinez un diagramme montrant l'architecture client/serveur de IMAP4, ainsi que les ports TCP utilisés.
26. Décrivez les principales fonctionnalités de IMAP4.
27. Donnez quelques-unes des opérations que IMAP4 sait exécuter.
28. Lorsque IMAP4 est utilisé sur une station de travail, s'en sert-on pour envoyer du courrier, recevoir du courrier, ou les deux ? Quels sont, le cas échéant, les autres protocoles requis ?
29. Décrivez le comportement de IMAP4 en termes de diagramme d'états.
30. Dessinez un diagramme montrant l'architecture client/serveur de Telnet et les ports utilisés par Telnet.
31. Un utilisateur tape les commandes suivantes en utilisant Telnet :

```
mv file.1 file.2 cr lf
```

cr et *lf* sont les caractères retour chariot et saut de ligne, qui permettent de marquer la fin d'une commande.

Combien de paquets de données seront-ils nécessaires au client Telnet pour qu'il transmette cette commande en mode ligne ? Qu'en est-il sans l'emploi de ce mode ligne ?
32. Qu'est-ce que le format NVT ? A quoi sert-il ?
33. Comment envoie-t-on une commande Telnet en utilisant NVT ?
34. Si chaque caractère saisi par un utilisateur d'une session Telnet est envoyé dans un segment TCP séparé, quelle est la perte de performance due à l'envoi d'un caractère ? Prenez en compte les protocoles en partant de la couche liaison de données, qu'on supposera être Ethernet.
35. Si chaque caractère saisi par un utilisateur d'une session Telnet est envoyé dans un segment TCP séparé, quelle est la perte de performance due à l'envoi d'un caractère ? Prenez en compte les protocoles en partant de la couche Physique, qu'on supposera être Ethernet, tout comme la couche liaison de données.
36. Quelle est la commande NVT envoyée lorsqu'un utilisateur fait une faute de frappe au cours d'une session Telnet et appuie sur BackSpace (Retour arrière) pour la corriger ?
37. Quel est le but de la négociation d'options de Telnet ? Comment est-elle réalisée ?
38. Donnez quelques-unes des commandes Berkeley r* et décrivez-les brièvement.

39. Afin de fournir un accès transparent en utilisant les commandes Berkeley r*, quels sont les fichiers utilisés ?
40. Dessinez un diagramme montrant la pile de protocoles et les numéros de port utilisés par rlogin.
41. La commande suivante doit être exécutée sur la machine d'adresse IP 142.44.45.44, et les résultats doivent être renvoyés dans un fichier local appelé local.cpy.
cat /etc/passwd
Ecrivez la commande rsh permettant d'exécuter cette tâche.
Quelle est le but de la fonction rsh ?
42. Comment peut-on exécuter la fonction de la question 41 en utilisant la commande Berkeley rcp ?
43. Quels sont les problèmes de sécurité liés à l'utilisation des commandes Berkeley r* ?
44. Donnez la liste des connexions TCP utilisées dans une session FTP et décrivez leur fonction. Ces connexions restent-elles ouvertes pendant toute la durée de la session FTP ?
45. Dessinez un diagramme montrant la pile de protocoles et les numéros de port utilisés lors d'une session FTP.
46. Comment un serveur FTP fait-il pour savoir à quel port du client FTP se connecter pour la connexion de données ?
47. Listez les types de messages utilisés par TFTP.
48. Dessinez un diagramme montrant la pile de protocoles et les numéros de port utilisés lors d'une session TFTP.
49. Combien de paquets TFTP faut-il pour envoyer un fichier de 10 000 octets ?
50. Combien de paquets TFTP faut-il pour envoyer un fichier de 8 192 octets ?
51. Quelle est la relation entre les numéros de version NFS et le protocole de transport utilisé ?
52. Quels sont les protocoles des couches Session et Présentation de NFS ? Décrivez-les brièvement.
53. Quels sont les protocoles de la couche application de NFS ? Décrivez-les brièvement.



54. Comment peut-on utiliser NFS_PROC, la procédure 0, à des fins de diagnostic ?
55. Qu'est-ce que Web NFS ?
56. Qu'est-ce qu'un descripteur NFS ? Pourquoi utilise-t-on un descripteur public pour Web NFS ?
57. Que sont HTTP et HTML ?
58. Dessinez un diagramme montrant les interactions client/serveur entre un client et un serveur Web. Notez tous les numéros de port utilisés.
59. Qu'est-ce qu'une balise HTML ? Qu'est-ce qu'une balise d'ancrage ? Donnez-en un exemple.
60. Qu'est-ce qu'une URL ? Décrivez-en la syntaxe et montrez comment l'utiliser pour spécifier que le document /usr/public.doc doit être téléchargé depuis l'hôte **krishna.vaikunth.heaven.org**.
61. A quoi sert CGI sur un serveur Web ?
62. Dessinez un diagramme montrant une interaction HTTP entre un navigateur Web et un serveur Web.
63. Que sont Gopher et Veronica ?
64. Dessinez un diagramme montrant les interactions client/serveur entre un client Gopher et un serveur Gopher. Notez les ports mis en œuvre.
65. Décrivez les interactions entre un client Gopher et un serveur Gopher en termes de messages échangés.

14

Gestion de réseau TCP/IP

Les réseaux TCP/IP utilisent un protocole standard de gestion appelé SMTP (*Simple Network Management Protocol*). Le protocole SNMP est très répandu dans l'industrie. Développé au départ pour la gestion des réseaux TCP/IP, il a ensuite été étendu à d'autres protocoles de transport, notamment IPX, AppleTalk et OSI.

Introduction à la gestion de réseau TCP/IP

Nous présenterons dans cette section les concepts fondamentaux de SNMP. Ces concepts s'appliquent à toutes les implémentations de SNMP, quelle que soit la plate-forme. Nous commencerons par présenter un modèle de gestion de réseau avant d'aborder SNMP. Ce modèle décrit les enjeux de la gestion de réseau. Il est adapté à tous les protocoles de gestion de réseau, y compris SNMP.

• Le modèle de gestion de réseau

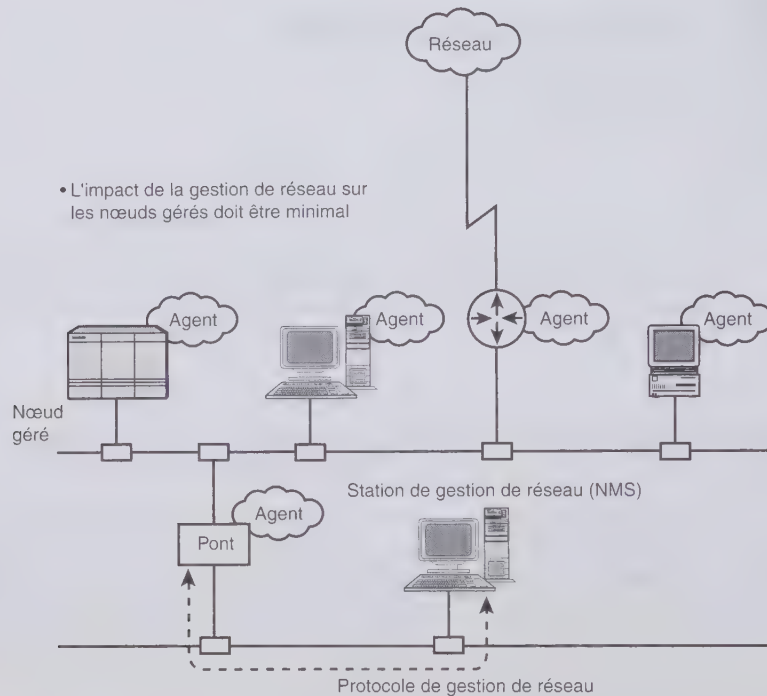
La Figure 14.1 montre un modèle de gestion de réseau. Dans ce modèle, le réseau est constitué de plusieurs périphériques, chacun d'entre eux exécutant un *agent de gestion*. Cet agent connaît les paramètres du périphérique sur lequel il s'exécute. Certains de ces paramètres sont spécifiques au périphérique : un périphérique de routage disposera par exemple de sa table de routage. Tous les périphériques ont des paramètres en commun, par exemple un nom, une durée de fonctionnement sans incident, etc.

La Figure 14.1 montre que les agents peuvent être gérés par un périphérique particulier appelé station de gestion de réseau (ou NMS, *Network Management Station*). La station NMS peut envoyer des requêtes à un périphérique afin d'obtenir des informations sur son paramétrage. L'agent du périphérique reçoit la requête et renvoie les informations demandées. Lorsqu'elle reçoit cette réponse, la station NMS peut utiliser les informations de configuration du périphérique afin de déterminer les opérations à entreprendre en fonction de son état.



Figure 14.1

Modèle de gestion de réseau
(source : Learning Tree).

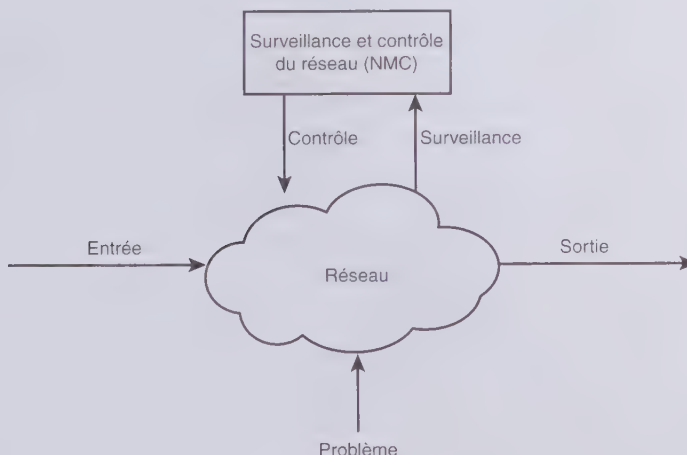


Il est aussi important d'empêcher une station NMS non autorisée d'avoir accès aux informations de paramétrage des périphériques composant le réseau. Il faut donc disposer d'un mécanisme d'authentification afin d'empêcher les accès non autorisés.

La Figure 14.2 donne les enjeux de la gestion de réseau. Le réseau est une boîte noire disposant d'entrées et de sorties. En entrée, les données partagées et l'activité générée par les utilisateurs du réseau. En sortie, une augmentation d'efficacité liée au partage des informations. Le réseau peut subir des dysfonctionnements liés à un ordinateur, un périphérique ou un brin du réseau. Le but de la gestion de réseau est d'effectuer un suivi de l'état du réseau et d'utiliser des mécanismes de contrôle afin d'assurer l'opération optimale du réseau en dépit d'éventuels dysfonctionnements.

Les mécanismes permettant de suivre et de contrôler le réseau doivent avoir le plus petit impact possible sur le réseau. En d'autres termes, les protocoles utilisés pour la collecte d'informations ne doivent pas amoindrir les performances du réseau et des périphériques qui le composent. Si le mécanisme de gestion de réseau utilise trop de bande passante, il pénalisera les utilisateurs. De la même façon, les agents s'exécutant sur les différents périphériques ne doivent pas consommer trop de ressources système, sans quoi les périphériques ne pourront pas effectuer correctement leurs tâches.

Figure 14.2
Les objectifs
de la gestion de réseau.



• SNMP et nœuds gérés

Le périphérique géré par la station NMS est appelé le nœud géré (*managed object*). Ce nœud géré (voir Figure 14.3) dispose de paramètres que la station NMS peut interroger. Le rôle du *protocole de gestion* est d'établir les communications entre la station NMS et le nœud géré, d'envoyer des requêtes et de recevoir des réponses. SNMP est un protocole de gestion.

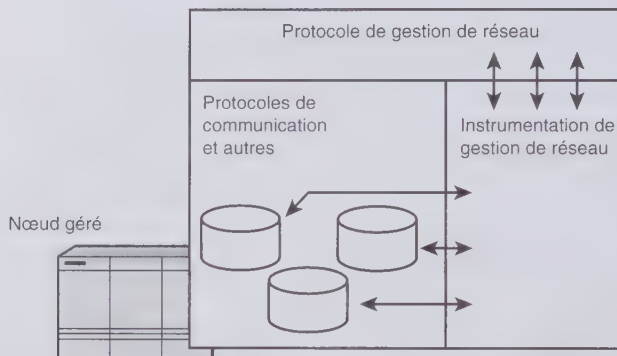
Le protocole de gestion s'interface avec la fonction NMI (*network management instrumentation*, ou "instrumentation de gestion de réseau") dans le nœud géré. L'instrumentation de gestion dispose d'informations sur les paramètres et les emplacements mémoire du nœud géré. Lorsqu'une requête est transmise via le protocole de gestion, par exemple SNMP, l'instrumentation de gestion du réseau reçoit la requête et accède aux paramètres du nœud géré. Les résultats sont alors renvoyés à la station NMS par le biais de l'instrumentation de gestion en utilisant le protocole de gestion de réseau.

SNMP appelle le nœud géré *périphérique géré*. Ces termes seront utilisés indifféremment tout au long de ce chapitre.

SNMP et MIB (*Management Information Base*) ont subi d'énormes transformations ces dernières années. Voici une liste non exhaustive des RFC traitant de SNMP :

- RFC 2128 PS G. Roeck, "Dial Control Management Information Base using SMIV2," 31/03/1997. (Pages = 34) (Format = .txt)



Figure 14.3*Le nœud géré**(source : Learning Tree).*

- Le protocole de gestion de réseau permet de surveiller et de contrôler le nœud géré.
- L'instrumentation de gestion de réseau permet d'accéder aux structures de données du nœud géré.

- RFC 2127 PS G. Roeck, "ISDN Management Information Base," 31/03/1997. (Pages = 49) (Format = .txt)
- RFC 2096 PS F. Baker, "IP Forwarding Table MIB," 30/01/1997. (Pages = 21) (Format = .txt) (Remplace la RFC1354)
- RFC 2037 PS K. McCloghrie, A. Bierman, "Entity MIB," 30/10/1996. (Pages = 35) (Format = .txt)
- RFC 2021 PS S. Waldbusser, "Remote Network Monitoring Management Information Base Version 2 using SMIV2," 16/01/1997. (Pages = 130) (Format = .txt)
- RFC 2020 PS J. Flick, "Definitions of Managed Objects for IEEE 802.12 Interfaces," 17/10/1996. (Pages = 31) (Format = .txt)
- RFC 2013 PS K. McCloghrie, "SNMPv2 Management Information Base for the User Datagram Protocol using SMIV2," 12/11/1996. (Pages = 6) (Format = .txt) (Met à jour la RFC1213)
- RFC 2012 PS K. McCloghrie, "SNMPv2 Management Information Base for the Transmission Control Protocol," 12/11/1996. (Pages = 10) (Format = .txt) (Met à jour la RFC1213)
- RFC 2011 PS K. McCloghrie, "SNMPv2 Management Information Base for the Internet Protocol using SMIV2," 12/11/1996. (Pages = 18) (Format = .txt) (Met à jour la RFC1213)

- RFC 1910 E G. Waters, "User-based Security Model for SNMPv2," 28/02/1996. (Pages = 44) (Format = .txt)
- RFC 1909 E K. McCloghrie, "An Administrative Infrastructure for SNMPv2," 28/02/1996. (Pages = 19) (Format = .txt)
- RFC 1908 DS J. Case, K. McCloghrie, M. Rose, S. Waldbusser, "Coexistence between Version 1 and Version 2 of the Internet-standard Network Management Framework," 22/01/1996. (Pages = 10) (Format = .txt) (Remplace la RFC1452)
- RFC 1907 DS J. Case, K. McCloghrie, M. Rose, S. Waldbusser, "Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)," 22/01/1996. (Pages = 20) (Format = .txt) (Remplace la RFC1450).
- RFC 1906 DS J. Case, K. McCloghrie, M. Rose, S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)," 22/01/1996. (Pages = 13) (Format = .txt) (Remplace la RFC1449)
- RFC 1904 DS J. Case, K. McCloghrie, M. Rose, S. Waldbusser, "Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)," 22/01/1996. (Pages = 24) (Format = .txt) (Remplace la RFC1444).
- RFC 1903 DS J. Case, K. McCloghrie, M. Rose, S. Waldbusser, "Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)," 22/01/1996. (Pages = 23) (Format = .txt) (Remplace la RFC1443).
- RFC 1902 DS J. Case, K. McCloghrie, M. Rose, S. Waldbusser, "Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)," 22/01/1996. (Pages = 40) (Format = .txt) (Remplace la RFC1442).
- RFC 1901 E J. Case, K. McCloghrie, M. Rose, S. Waldbusser, "Introduction to Community-based SNMPv2," 22/01/1996. (Pages = 8) (Format = .txt)
- RFC 1759 PS R. Smith, F. Wright, T. Hastings, S. Zilles, J. Gyllenskog, "Printer MIB," 28/03/1995. (Pages = 113) (Format = .txt)
- RFC 1757 DS S. Waldbusser, "Remote Network Monitoring Management Information Base," 10/02/1995. (Pages = 91) (Format = .txt) (Remplace la RFC1271)
- RFC 1749 PS K. McCloghrie, F. Baker, E. Decker, "IEEE 802.5 Station Source Routing MIB using SMIV2," 29/12/1994. (Pages = 10) (Format = .txt) (Mise à jour de la RFC1748)
- RFC 1748 DS K. McCloghrie, E. Decker, "IEEE 802.5 MIB using SMIV2," 29/12/1994. (Pages = 25) (Format = .txt) (Mise à jour par la RFC1749)



- RFC 1724 DS G. Malkin, F. Baker, "RIP Version 2 MIB Extension," 15/11/1994. (Pages = 18) (Format = .txt) (Remplace la RFC1389)
- RFC 1697 PS D. Brower, R. Purvy, A. Daniel, M. Sinykin, J. Smith, "Relational Database Management System (RDBMS) Management Information Base (MIB) using SMIPv2," 23/08/1994. (Pages=38) (Format=.txt)
- RFC 1696 PS J. Barnes, L. Brown, R. Royston, S. Waldbusser, "Modem Management Information Base (MIB) using SMIPv2," 25/08/1994. (Pages = 31) (Format = .txt)
- RFC 1695 PS M. Ahmed, K. Tesink, "Definitions of Managed Objects for ATM Management Version 8.0 using SMIPv2," 25/08/1994. (Pages = 73) (Format = .txt)
- RFC 1694 DS T. Brown, K. Tesink, "Definitions of Managed Objects for SMDS Interfaces using SMIPv2," 23/08/1994. (Pages = 35) (Format = .txt) (Remplace la RFC1304)
- RFC 1666 PS Z. Kielczewski, D. Kostick, K. Shih, "Definitions of Managed Objects for SNA NAUs using SMIPv2," 11/08/1994. (Pages = 68) (Format = .txt)
- RFC 1660 DS B. Stewart, "Definitions of Managed Objects for Parallel-printer-like Hardware Devices using SMIPv2," 20/07/1994. (Pages = 10) (Format = .txt) (Remplace la RFC1318)
- RFC 1659 DS B. Stewart, "Definitions of Managed Objects for RS-232-like Hardware Devices using SMIPv2," 20/07/1994. (Pages = 21) (Format = .txt) (Remplace la RFC1317)
- RFC 1658 DS B. Stewart, "Definitions of Managed Objects for Character Stream Devices using SMIPv2," 20/07/1994. (Pages = 18) (Format = .txt) (Remplace la RFC1316)
- RFC 1657 PS S. Willis, J. Burruss, J. Chu, "Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIPv2," 21/07/1994. (Pages = 21) (Format = .txt)
- RFC 1650 PS F. Kastenholz, "Definitions of Managed Objects for the Ethernet-like Interface Types using SMIPv2," 23/08/1994. (Pages = 20) (Format = .txt)
- RFC 1643 S F. Kastenholz, "Definitions of Managed Objects for the Ethernet-like Interface Types," 13/07/1994. (Pages = 19) (Format = .txt) (Remplace la RFC1623) (STD 50)
- RFC 1905 DS J. Case, K. McCloghrie, M. Rose, S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)," 22/01/1996. (Pages = 24) (Format = .txt) (Remplace la RFC1448)
- RFC 1628 PS J. Case, "UPS Management Information Base," 19/05/1994. (Pages = 45) (Format = .txt)

- RFC 1612 PS R. Austein, J. Saperia, "DNS Resolver MIB Extensions," 17/05/1994. (Pages = 36) (Format = .txt)
- RFC 1611 PS R. Austein, J. Saperia, "DNS Server MIB Extensions," 17/05/1994. (Pages = 32) (Format = .txt)
- RFC 1604 PS T. Brown, "Definitions of Managed Objects for Frame Relay Service," 25/03/1994. (Pages = 46) (Format = .txt) (Remplace la RFC1596)
- RFC 1595 PS T. Brown, K. Tesink, "Definitions of Managed Objects for the SONET/SDH Interface Type," 11/03/1994. (Pages = 59) (Format = .txt)
- RFC 1567 PS G. Mansfield, S. Kille, "X.500 Directory Monitoring MIB," 11/01/1994. (Pages = 19) (Format = .txt)
- RFC 1566 PS N. Freed, S. Kille, "Mail Monitoring MIB," 11/01/1994. (Pages = 21) (Format = .txt)
- RFC 1565 PS N. Freed, S. Kille, "Network Services Monitoring MIB", 11/01/1994. (Pages=18) (Format=.txt)
- RFC 1525 PS E. Decker, K. McCloghrie, P. Langille, A. Rijssinghani, "Definitions of Managed Objects for Source Routing Bridges," 30/09/1993. (Pages = 18) (Format = .txt) (Remplace la RFC1286)
- RFC 1503 I K. McCloghrie, M. Rose, "Algorithms for Automating Administration in SNMPv2 Managers," 26/08/1993. (Pages = 19) (Format = .txt)
- RFC 1461 PS D. Throop, "SNMP MIB extension for MultiProtocol Interconnect over X.25," 27/05/1993. (Pages = 30) (Format = .txt)
- RFC 1451 PS J. Case, K. McCloghrie, M. Rose, S. Waldbusser, "Manager to Manager Management Information Base," 03/05/1993. (Pages = 36) (Format = .txt)
- RFC 1447 PS K. McCloghrie, J. Galvin, "Party MIB for version 2 of the Simple Network Management Protocol (SNMPv2)," 03/05/1993. (Pages = 50) (Format = .txt)
- RFC 1446 PS J. Galvin, K. McCloghrie, "Security Protocols for version 2 of the Simple Network Management Protocol (SNMPv2)," 03/05/1993. (Pages = 51) (Format = .txt)
- RFC 1445 PS J. Davin, K. McCloghie, "Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2)," 03/05/1993. (Pages = 47) (Format = .txt)



- RFC 1441 PS J. Case, K. McCloghrie, M. Rose, S. Waldbusser, "Introduction to version 2 of the Internet-standard Network Management Framework," 03/05/1993. (Pages = 13) (Format = .txt)
- RFC 1420 PS S. Bostock, "SNMP over IPX," 03/03/1993. (Pages = 4) (Format = .txt) (Remplace la RFC1298)
- RFC 1419 PS G. Minshall, M. Ritter, "SNMP over AppleTalk," 03/03/1993. (Pages = 7) (Format = .txt)
- RFC 1418 PS M. Rose, "SNMP over OSI," 03/03/1993. (Pages = 4) (Format = .txt) (Remplace la RFC1283)

• Management Information Base

Les paramètres d'un nœud géré s'appellent des *objets de gestion*. L'ensemble des paramètres d'un nœud géré s'appelle la MIB (*Management Information Base*). La MIB peut se concevoir comme une base de données. Les objets contenus dans la MIB, appelés des *variables*, ont chacun un identifiant unique. Cet identifiant numérique est souvent appelé l'*id d'objet*. Cet identifiant est formé en suivant un système de numérotation hiérarchique permettant d'ordonner les variables de la MIB. Cette notion est identique à la notion d'index dans une base de données, et permet de rechercher des objets et de les parcourir en passant d'un objet au suivant. Les messages SNMP demandent et renvoient des variables MIB.

Les variables MIB contiennent un indicateur d'état indiquant si la variable est en lecture seule ou si elle permet un accès en écriture.

Il existe un certain nombre de variables MIB standard pour les différents éléments des protocoles TCP/IP. Ces variables MIB traitent des valeurs des paramètres IP, ICMP, TCP, SNMP, EGP et les tables de conversion d'adresses.

Les interfaces de la couche liaison de données, notamment Ethernet, Token Ring, SMDS et ATM, disposent de leur propre jeu de variables MIB. Les fabricants de périphériques peuvent en outre définir leurs propres variables MIB, qu'on appelle tout naturellement des *MIB propriétaires*. Des mécanismes d'interface permettent à un gestionnaire SNMP de demander une description d'une MIB propriétaire et de la compiler afin qu'elle soit reconnue par le gestionnaire SNMP.

• Le paradigme de gestion de SNMP

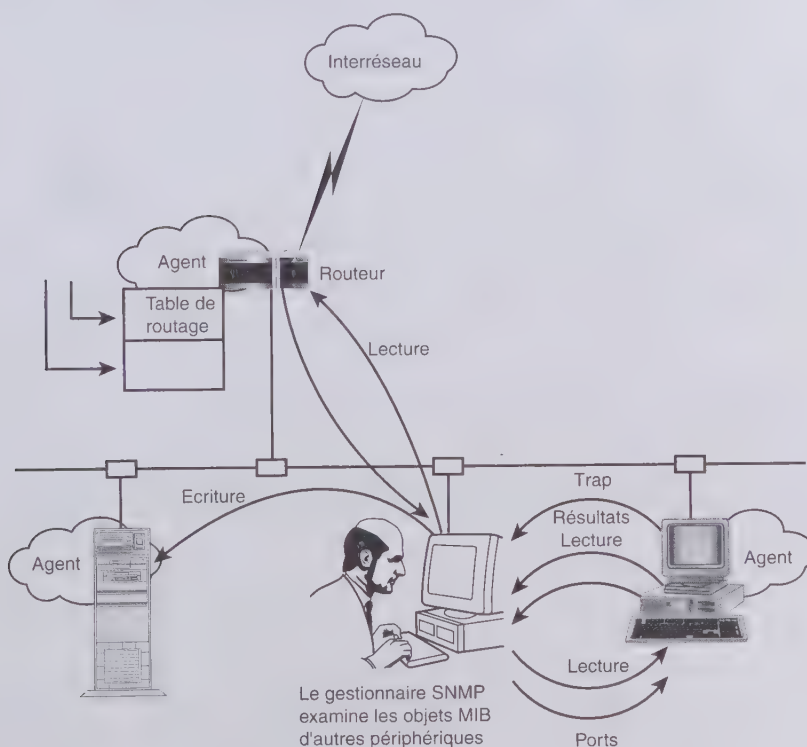
La station NMS (Station de gestion de réseau) de SNMP est souvent appelée *gestionnaire SNMP*. Ce gestionnaire utilise un modèle de gestion appelé le paradigme du débogage à distance (voir Figure 14.4). Dans ce modèle, le gestionnaire SNMP se comporte comme un développeur sur une station de travail qui déboguerait des programmes à distance. Ce développeur aurait besoin de lire les variables

du programme et de les modifier. De la même façon, le gestionnaire SNMP doit être à même de lire et de modifier les valeurs des variables MIB des périphériques qu'il gère. Le gestionnaire SNMP sait effectuer les actions suivantes :

- lire et écrire des variables MIB ;
- effectuer des scrutations dirigées par trap ;
- effectuer un simple parcours des variables du nœud géré.

Figure 14.4

Le modèle de gestion SNMP
source : Learning Tree).



Lorsqu'un événement inattendu se produit sur un des périphériques gérés, par exemple un échec de transmission ou une modification d'état, le périphérique envoie un message de *Trap SNMP* au gestionnaire SNMP. Ce message contient une indication de l'événement ayant provoqué la génération du message. Le gestionnaire SNMP doit alors prendre les mesures qui s'imposent. Il peut se contenter



d'enregistrer le message dans un fichier de suivi, ou prendre des mesures plus immédiates, par exemple demander des informations complémentaires au périphérique lui ayant envoyé le message. Ces informations supplémentaires sont obtenues grâce à des requêtes de lecture des variables MIB. Si le gestionnaire SNMP est programmé pour contrôler le périphérique, il peut lui demander de modifier la valeur de ses variables MIB.

Lorsque le gestionnaire décide de modifier l'état du périphérique, il le fait en modifiant les variables MIB du périphérique. Par exemple, le gestionnaire modifiera la variable MIB *ipPowerOff* d'un périphérique afin de l'éteindre à distance.

Comme les variables MIB sont ordonnées selon leur identificateur d'objet, le gestionnaire SNMP peut parcourir toutes les variables du périphérique en utilisant la commande SNMP *GetNext*. En résumé, voici les commandes utilisées par le gestionnaire SNMP :

- Get (Lecture d'une variable MIB) ;
- Set (Modification d'une variable MIB) ;
- GetNext (Récupération de la variable MIB suivante) ;
- Trap (Message envoyé au gestionnaire SNMP en cas de comportement anormal).

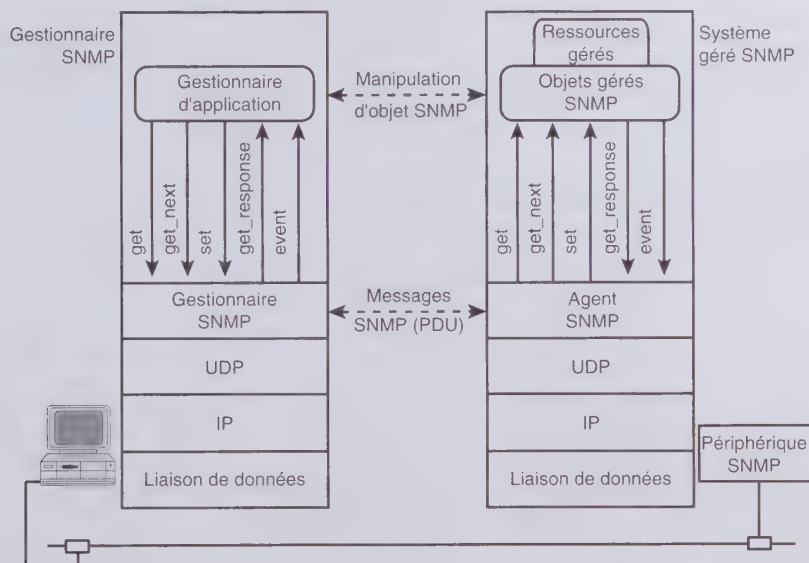
• Les commandes et protocoles SNMP

La Figure 14.5 montre les commandes SNMP ainsi que les protocoles afférents. Le gestionnaire SNMP peut envoyer l'une des commandes suivantes :

- Get ;
- GetNext ;
- Set.

Ces commandes sont envoyées aux agents SNMP en utilisant le protocole UDP/IP. L'agent SNMP peut envoyer une commande SNMP *GetNext_Response* en réponse à une requête *Get* ou à une requête *GetNext* en provenance du gestionnaire SNMP. La commande SNMP *Set* en provenance du gestionnaire SNMP n'est jamais explicitement acquittée. En d'autres termes, il n'existe pas de commande *Set_Response* envoyée par un agent SNMP au gestionnaire SNMP. Les événements *Trap* sont envoyés par l'agent SNMP au gestionnaire SNMP lorsque le périphérique géré passe dans un état inhabituel.

Figure 14.5
L'architecture SNMP
(source : Learning Tree).



• Traps SNMP

Lorsqu'un périphérique entre dans un état anormal, l'agent SNMP prévient le gestionnaire SNMP par le biais d'un Trap SNMP. La Figure 14.6 montre un petit réseau et quelques-uns des messages Trap générés par les agents SNMP. Les messages Trap peuvent être *link up* ou *link down* (lorsque l'interface devient active ou au contraire devient passive), *cold start* (démarrage à froid) ou *warm start* (démarrage à chaud) ; réinitialisation de l'agent SNMP, *authentication failure* (échec d'authentification ; lorsqu'un nom de communauté incorrect est spécifié dans une requête), *loss of EGP neighbour* (perte de voisin EGP) lorsqu'un agent ne peut communiquer avec son voisinage EGP. Mise à part la perte de voisinage EGP, tous les messages précédents peuvent survenir sur un réseau NetWare.

Le protocole EGP a été très employé sur l'Internet, et c'est pour cette raison qu'il fait encore partie des messages Trap. Le Tableau 14.1 reprend les différents messages Trap.



Figure 14.6

Traps SNMP.

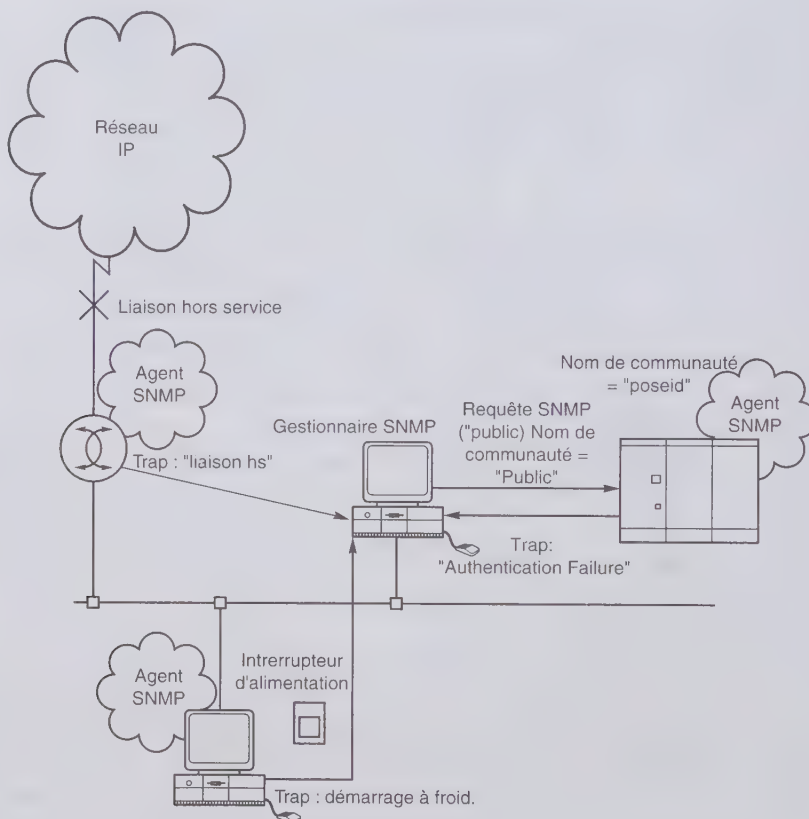


Tableau 14.1 : Messages Trap SNMP

Paramètre	Description
Link down ou Link up	Lorsqu'une carte réseau du périphérique cesse de fonctionner, un message <i>link down</i> est généré. Lorsque la carte réseau redevient active, un message <i>link up</i> est généré.
Cold ou warm	Lorsqu'un agent SNMP est initialisé, il génère un message <i>cold start</i> . Lorsque l'agent se contente de réinitialiser sa table, il envoie un message <i>warm start</i> .
Authentication	Lorsqu'un agent SNMP reçoit une requête dont le nom de communauté ne correspond pas au sien, il génère un message <i>authentication failure</i> .
Loss of EGP	Lorsqu'un agent SNMP ne peut plus communiquer avec son voisinage EGP, il génère un message <i>loss of EGP neighbor</i> .

Messages et identifiants d'objets SNMP

Les sections suivantes abordent la structure des messages SNMP. Nous ne passerons en revue que les messages les plus courants de SNMP version 1, SNMP 2 n'étant pas encore très répandu.

La plupart des messages et réponses SNMP contiennent le nom de l'objet MIB sur lequel on doit intervenir. Avant de nous pencher sur la structure d'un message SNMP, voyons comment sont nommés les objets MIB.

• Identifiants d'objets MIB

Sous SNMP, les objets MIB reçoivent un identifiant unique composé de séquences de chiffres séparés par des points. Cette séquence se lit de gauche à droite et correspond à des nœuds dans l'arborescence des noms. La Figure 14.7 reprend un arbre partiel de nommage montrant les identifiants d'objets MIB des objets sysDescr et sysLocation. Les nœuds de premier niveau de l'arbre représentent les différents comités et organisations responsables de la composition du nom. Les nœuds de premier niveau suivants ont été définis :

- **ccitt(0).** Les nœuds situés dans la branche ccitt(0) sont administrés par le CCITT.
- **iso(1).** Les nœuds situés dans la branche ccitt(0) sont administrés par l'ISO et l'IEC.
- **joint-iso-ccitt(2).** Ces nœuds sont co-administrés par les trois organismes précédents.

La gestion de réseau est définie par la branche iso(1). Dans cette branche, on trouve un certain nombre de définitions d'organisations subordonnées. La gestion de réseau entre dans le nœud org(3), comme illustré sur la Figure 14.7.

Sous le nœud dod(6) se trouvent un certain nombre de réseaux subordonnés. La gestion de réseau entre dans le nœud internet(1) pour l'Internet (voir Figure 14.7).

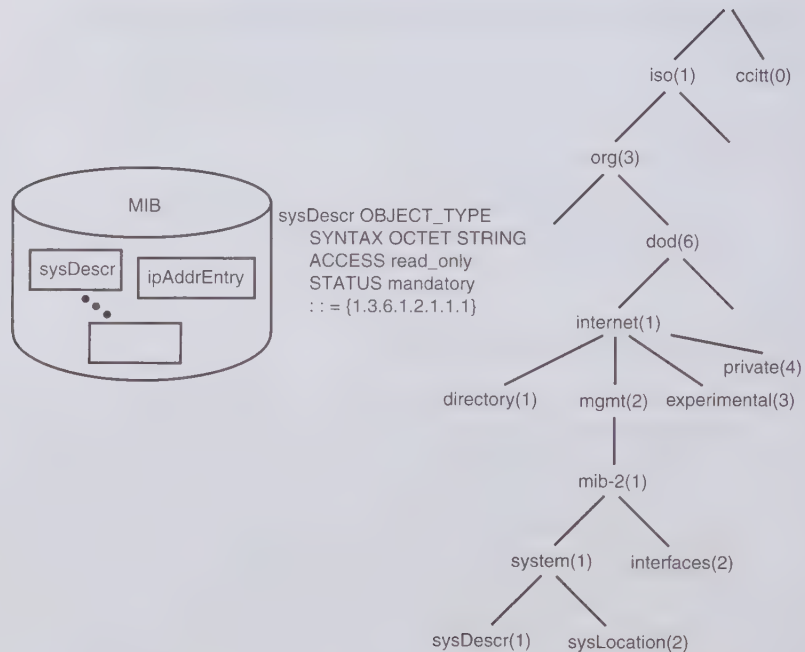
Sous le nœud internet(1) se trouvent un certain nombre de nœuds subordonnés représentant différents services et tentatives de normalisation. La gestion de réseau standardisée se trouve dans le nœud mgnt(2), (voir Figure 14.7).

Sous le nœud mgnt(2) se trouvent un certain nombre de nœuds subordonnés représentant différentes tentatives de normalisation. La gestion de réseau standardisée se trouve dans le nœud mib-2(1), comme illustré sur la Figure 14.7.



Figure 14.7

Un arbre de nommage
(source : *Learning Tree*).



Sous le nœud mib-2(1) se trouvent un certain nombre de nœuds subordonnés représentant différents groupements de variables MIB. La Figure 14.7 montre les groupements system(1) et interfaces(2).

Sous le nœud system(1), on trouve les deux variables MIB sysDescr(1) et sysLocation(2). Les identifiants d'objets de ces variables s'obtiennent en écrivant de gauche à droite les différents nœuds, séparés par des points. Les identifiants de ces variables MIB sont donc :

- sysDescr : 1.3.6.1.2.1.1.1 ;
- sysLocation : 1.3.6.1.2.1.1.2.

Remarquez qu'on observe un ordonnancement naturel des variables MIB. La variable MIB suivant sysDescr est sysLocation.

• Les messages SNMP

Le format et la longueur des messages SNMP sont variables et relativement complexes. On utilise le langage ASN.1 (*Abstract Syntax Notation*, version 1) pour décrire la structure des messages SNMP. ASN.1 est un langage relativement intuitif, mais sa définition rigoureuse sortirait du cadre de cet ouvrage.

Voici un exemple d'utilisation d'ASN.1, décrivant la structure d'une trame Ethernet :

```
Ethernet-Frame ::= SEQUENCE {
    destAddr  OCTET STRING (SIZE(6)),
    srcAddr   OCTET STRING (SIZE(6)),
    etherType INTEGER (1501..65535),
    data      ANY(SIZE(46-1500)),
    crc       OCTET STRING(SIZE(4))
}
```

Remarquez que le ::= sert à attribuer à la variable de gauche la définition du membre de droite. SEQUENCE représente une liste ordonnée d'éléments. Cette liste ordonnée contient les champs d'une trame Ethernet, notamment l'adresse de destination, l'adresse source, le type d'Ethernet, les données et le CRC.

Le type d'un champ est spécifié à la suite du nom. Par exemple, destAddr et srcAddr sont déclarés comme étant de type OCTET STRING, définissant une variable de 0 ou plusieurs octets. La valeur de chaque octet est comprise entre 0 et 255. La taille de la chaîne obtenue est placée après la déclaration de la chaîne OCTET STRING.

EtherType est défini en tant que INTEGER, à savoir une valeur entière de taille et de précision arbitraires. Le (1501..65535) situé juste après permet de définir sa plage de valeurs.

Le champ de données est de type ANY et sa taille varie de 46 à 1500 octets.

En utilisant la notation ASN.1, les messages SNMP prennent le format suivant :

```
SNMP-message ::= SEQUENCE {
    version INTEGER {version-1(1)},
    community OCTET STRING,
    data ANY}
```

Le format des messages donne le sens de chaque champ (voir Figure 14.8).

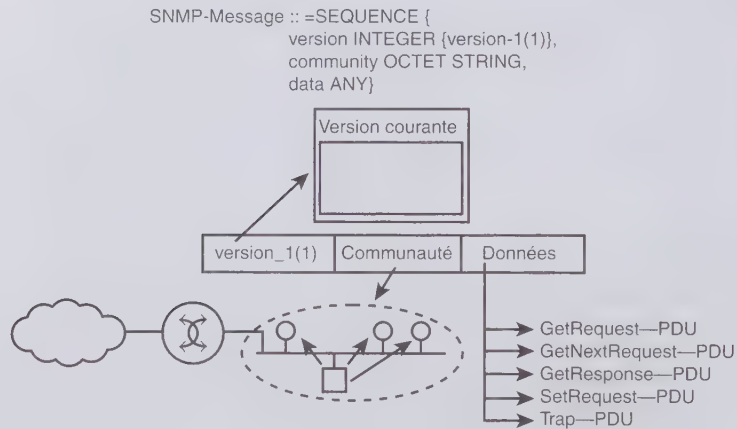
Le numéro de version est celui du format de message. Community est une valeur chaîne transmise dans chaque message SNMP. Un agent recevant un message SNMP commence par vérifier la valeur de ce champ. Si elle correspond à celle qu'il a stockée, l'opération demandée dans le message SNMP est effectuée. Sinon, l'agent SNMP renvoie un message Trap SNMP indiquant que l'authentification a échoué. La procédure d'authentification se rapproche beaucoup d'un système à



mot de passe (le nom de communauté), mais ce mot de passe n'est pas crypté et circule donc en clair sur le réseau.

Figure 14.8

Le format de message SNMP
(source : Learning Tree).



Le champ de données représente le détail des différents messages SNMP, notamment GetRequest, GetNextRequest, GetResponse, SetRequest et Trap. Le format de ces messages est décrit dans les Figure 14.9 et 14.10.

Figure 14.9

Le format SNMP PDU
(source : Learning Tree).

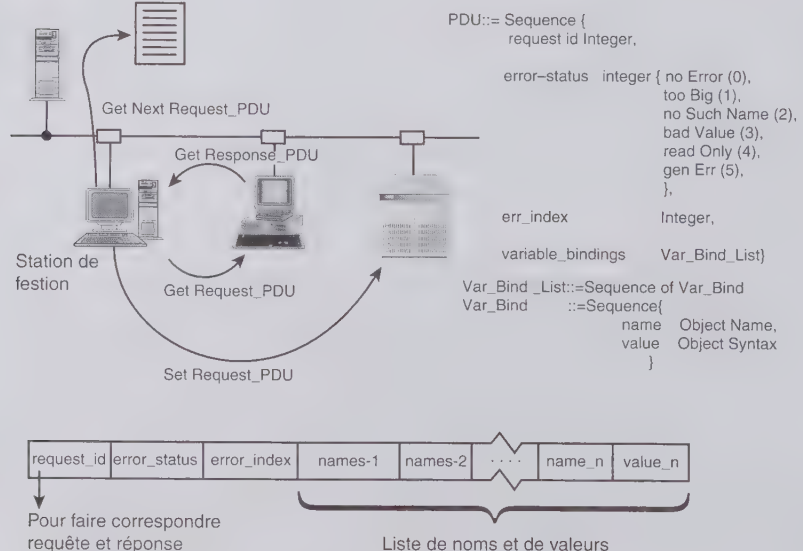
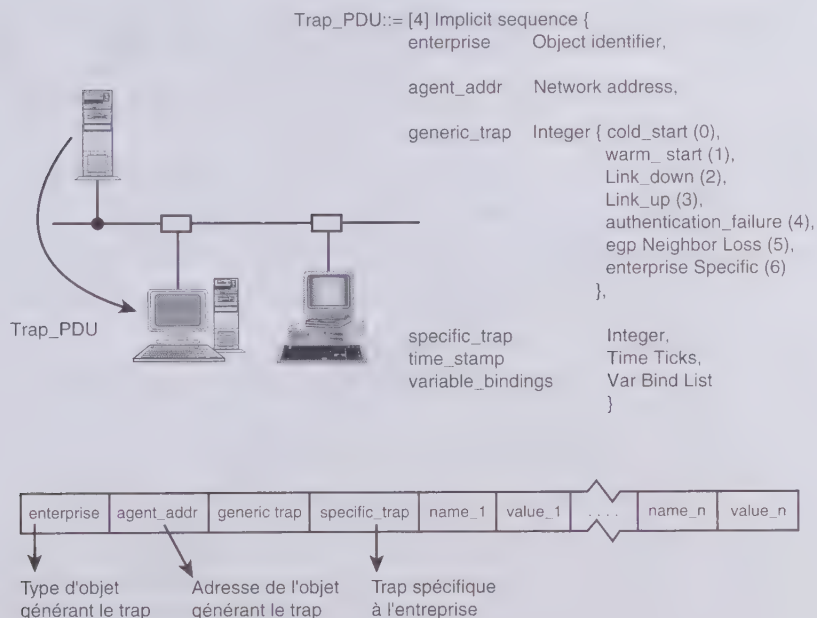


Figure 14.10

Le format Trap PDU
(source : Learning Tree).



Récapitulatif

Les réseaux TCP/IP utilisent un protocole de gestion de réseau standard appelé SNMP (*Simple Network Management Protocol*). SNMP est très largement répandu dans l'industrie. Il a été au départ développé pour la gestion des réseaux TCP/IP. SNMP n'est cependant plus limité aux réseaux TCP/IP. Il est implémenté pour d'autres protocoles de transport, notamment IPX et OSI.

Ce chapitre vous a donné quelques bases sur SNMP. Nous nous sommes limités au format des messages de SNMP 1.



Testez vos connaissances

1. SNMP se limite-t-il à TCP/IP ? Quels sont les autres protocoles de transport utilisables ?
2. Décrivez les rôles de l'agent SNMP et du gestionnaire SNMP.
3. Expliquez brièvement les enjeux de la gestion de réseau.
4. Décrivez le modèle d'un nœud géré dans le cadre de la gestion de réseau.
5. Quelle est la RFC traitant des extensions MIB des serveurs DNS ?
6. Qu'est-ce qu'un objet géré en SNMP ?
7. Quels sont les indicateurs (flags) d'état associés à une variable MIB ? Précisez leur fonction.
8. Expliquez le modèle de gestion de SNMP.
9. Comment effectue-t-on toutes les actions en SNMP ?
10. Citez les commandes SNMP version 1, ainsi que leurs fonctionnalités.
11. Dessinez un diagramme montrant l'architecture SNMP et les messages SNMP échangés entre un agent SNMP et le gestionnaire SNMP.
12. Qu'est-ce qu'un Trap SNMP ? Donnez quelques exemples de messages Trap.
13. Quel est le rôle de la chaîne de communauté des messages SNMP ?
14. Quand doit-on envoyer un message d'échec d'authentification SNMP ?
15. Pourquoi ASN.1 est-il important dans la définition de SNMP ?
16. Décrivez le format général d'un message SNMP. Utilisez ASN.1 pour décrire le format d'un message SNMP.
17. Quel est le message SNMP dont la structure est différente de celle des autres messages SNMP ?

L'AVENIR

15 *IPv6 et ATM* 643



15

IPv6 et ATM

Ce chapitre traite de deux technologies très intéressantes qui devraient jouer un rôle déterminant pour l'avenir de l'Internet et des réseaux intranets. Il s'agit d'IP Next Generation (ou IPv6) et d'ATM. Ce qui suit constitue un aperçu de ce que peut réserver l'avenir, nous n'aborderons pas le détail de ces deux sujets.

Certains aspects d'IPv6 ont été abordés aux cours des précédents chapitres. Au Chapitre 4, nous avons vu qu'IPv6 répondait à long terme aux carences des adresses d'IPv4. Dans ce chapitre, nous détaillerons certains aspects du protocole. ATM est une technologie prometteuse dans la mesure où elle permet de transporter sur le même réseau, local ou étendu, différents types de données, notamment le son et la vidéo. Dans ce chapitre, nous nous intéresserons à la façon dont fonctionne une pile TCP/IP sur un matériel ATM.

IPv6

La première raison d'être d'IPv6 était d'augmenter l'espace disponible pour les adresses afin de répondre aux besoins croissants des réseaux modernes, lesquels possèdent un grand nombre de périphériques connectés. Mais les concepteurs d'IPv6 ne se sont pas contentés d'agrandir l'espace d'adressage, ils ont tiré les enseignements de quelque quinze années d'exploitation d'IPv4.

Des diverses propositions que l'on a pu faire pour répondre aux problèmes d'adressage d'IPv4 et pour améliorer le protocole, IPv6 n'a retenu que les meilleures. Dans les sections qui suivent, nous détaillons la structure de l'en-tête IPv6.

- **Format de l'en-tête IPv6**

L'en-tête IPv6 se compose principalement d'une partie de taille fixe de 40 octets, dont 16 octets pour l'adresse d'origine et 16 octets pour l'adresse de destination. En plus de cet en-tête de base, on trouve parfois des extensions pour :

- le routage (en-tête Routing) ;
- la fragmentation (en-tête Fragment) ;



- les options de destination (en-tête Destination Options) ;
- la fin des en-têtes (en-tête No Next) ;
- les options pas à pas (en-tête Hop-by-Hop Options) ;
- l'identification (en-tête Authentication) ;
- la sécurité (en-tête Encrypted Security Payload).

Des informations supplémentaires peuvent figurer à côté de l'en-tête principal IPv6.

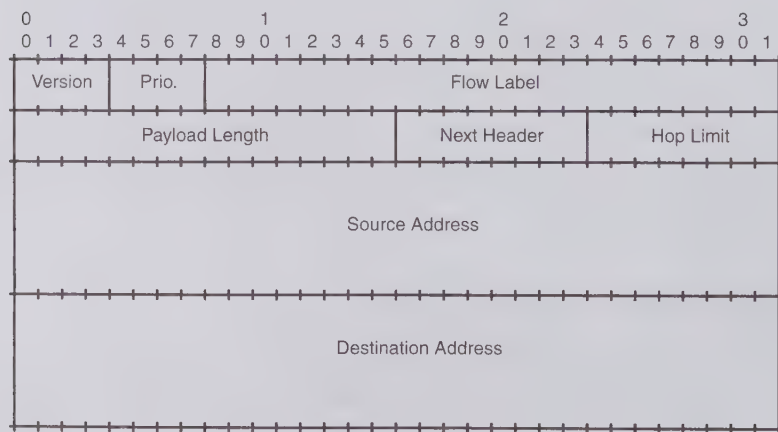
Les en-têtes d'extension fournissent des renseignements qui servent pour le traitement de IPv6. Ils décrivent des couches de sous-protocole à l'intérieur de la couche IPv6. Grâce à ces en-têtes d'extension, IPv6 peut définir des potentialités ou même permettre à l'utilisateur d'en définir.

Un datagramme IPv6 peut contenir plusieurs en-têtes d'extension (ou aucun). Les périphériques de destination se servent eux-mêmes de ces en-têtes, les périphériques intermédiaires, tels que les routeurs, se contentant alors de les transmettre sans les traiter. Certains en-têtes d'extension sont par ailleurs spécifiquement destinés aux routeurs (par exemple l'en-tête d'options pas à pas ou l'en-tête de routage).

La Figure 15.1 présente l'en-tête IPv6 de base.

Figure 15.1

L'en-tête IPv6 de base.

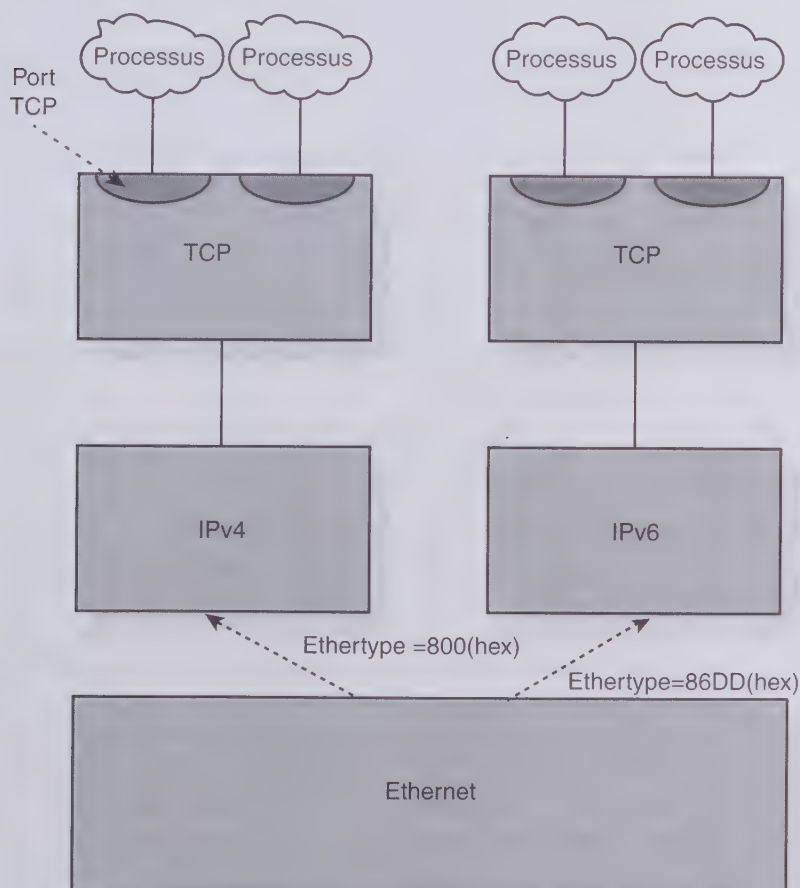


Remarquez que le premier champ est un champ de version de quatre bits. Ce champ est le même que pour IPv4, sauf que son contenu sera 6 au lieu de 4. Cette valeur permet d'indiquer que les champs qui suivent sont au format IPv6.

A l'origine, il était prévu que les deux protocoles IPv4 et IPv6 puissent fonctionner ensemble sur un même réseau en utilisant les mêmes pilotes de couche de liaison de données. C'est pourquoi on a affecté la valeur 86DD (hexadécimal) au champ Ethernet Type d'IPv6. La Figure 15.2 présente le démultiplexage d'entêtes IPv4 et IPv6 à partir du champ Ethernet Type.

Le champ Priority (priorité) occupe quatre bits et peut prendre 16 valeurs différentes. Grâce à ce champ, l'expéditeur peut attribuer un niveau de priorité aux

Figure 15.2
*Le démultiplexage
d'entêtes IPv4 et IPv6.*



datagrammes qu'il transmet. C'est l'équivalent du champ Precedence (préséance) qui existait dans IPv4. On peut distinguer deux intervalles dans les 16 valeurs possibles :

- **Valeurs 0 à 7.** Spécifient une priorité de trafic pour laquelle la source fournit un contrôle de congestion. Le trafic, par exemple le trafic TCP, se replie (*backing off*) en cas de congestion.
- **Valeurs 8 à 15.** Spécifient une priorité de trafic qui ne se replie pas en cas de congestion. Il s'agit par exemple de paquets en temps réel envoyés à débit constant.

Le Tableau 15.1 définit les valeurs de priorité pour un trafic avec contrôle de congestion.

Tableau 15.1 : Valeurs du champ Priority de l'en-tête IPv6

Valeur	Signification
0	Trafic sans autre caractéristique
1	"Filler" (remplissage) netnews par exemple
2	Transfert de données sans intervention (e-mail par exemple)
3	(Réservé)
4	Transfert en masse avec intervention (FTP ou NFS par exemple)
5	(Réservé)
6	Trafic interactif (Telnet ou X par exemple)
7	Contrôle de trafic Internet (protocoles de routage ou SNMP par exemple)

En ce qui concerne les trafics sans contrôle de congestion, la valeur de priorité la plus basse, c'est-à-dire 8, correspond en général à des paquets que l'expéditeur souhaite voir écartés en premier en cas de congestion. La valeur de priorité la plus haute est attribuée aux paquets qui doivent être écartés en dernier. Le niveau de priorité réel sera fixé par les applications et leurs options de configuration.

Le champ Flow Label occupe 24 bits et permet d'identifier les datagrammes transmis et que l'on souhaite voir traités d'une certaine façon. Les datagrammes dont le champ Flow Label contiennent la même valeur seront considérés comme faisant partie du même "flux". Considérons des datagrammes transportant tantôt des données audio, tantôt des données vidéo entre un serveur et une station multimédia. Ces datagrammes pourront présenter des valeurs différentes pour le champ Flow Label selon les cas, afin de regrouper leur traitement sur tout le réseau.

Le champ Payload Length est un champ de 16 bits contenant une valeur non signée qui représente la taille des données qui suivent l'en-tête. Ce nombre ne tient pas compte de l'en-tête IPv6 de base. Le champ Payload Length d'IPv6 équivaut au champ Total Length de IPv4. Notez cependant que dans IPv4, le champ Total Length incluait l'en-tête IP. La taille du champ Payload Length ne peut pas excéder 64 Ko. Pour le transport de datagrammes plus importants, IPv6 se sert de l'option jumbogram.

Le champ Next Header occupe huit bits et identifie le type d'en-tête qui fait immédiatement suite à l'en-tête IPv6 de base. Il peut s'agir d'une valeur identifiant le type de protocole, par exemple TCP (6) ou UDP (17). Une valeur identifiant un en-tête d'extension Ipv6 peut également être attribuée. Sur la Figure 15.3 on se sert du champ Next Header pour spécifier le message TCP qui suit. La valeur 59 correspond à l'absence d'en-tête supplémentaire : l'en-tête de base n'est suivi d'aucun protocole ni d'aucune donnée. Le Tableau 15.2 présente la signification de quelques valeurs pour le champ Next Header.

Tableau 15.2 : Les valeurs du champ Next Header pour quelques protocoles courants

Valeur décimale	Abréviation	Signification
0		Réservé (IPv4).
0	HBH	Option Hop-by-Hop (IPv6).
1	ICMP	Internet Control Message Protocol (IPv4).
2	IGMP	Internet Group Management Protocol (IPv4).
2	ICMP	Internet Control Message Protocol (IPv4).
3	GGP	Protocole de passerelle à passerelle.
4	IP	IP dans IP (encapsulation d'IPv4).
5	ST	Flux continu.
6	TCP	Transmission Control Protocol.
17	UDP	User Datagram Protocol.
29	ISO-TP4	ISO Transport Protocol Class 4.
37	DDP	Datagram Delivery Protocol.
43	RH	En-tête Routing (IPv6).

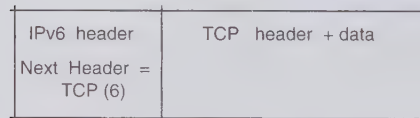


Tableau 15.2 : Les valeurs du champ Next Header pour quelques protocoles courants (*suite*)

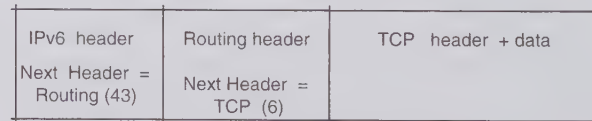
Valeur décimale	Abréviation	Signification
44	FH	En-tête Fragment (IPv6).
45	IDRP	Inter-Domain Routing Protocol.
51	AH	En-tête Authentication (IPv6).
52	ESP	En-tête Encrypted Security Payload (IPv6).
59	Null	Aucun en-tête ne suit (IPv6).
80	ISO-IP	ISO Internet Connectionless Protocol (CLNP).
88	IGRP	IGRP.
89	OSPF	Open Shortest Path First.
255		Réservé.

Figure 15.3

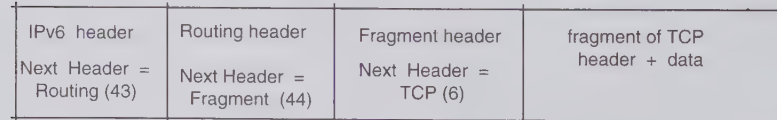
Utilisation du champ Next Header.



Basic IPv6 header encapsulating TCP segment



Basic IPv6 header, and routing header encapsulating TCP segment



Basic IPv6 header, routing header, and fragmentation header encapsulating TCP segment

Le champ Hop Limit occupe huit bits et contient une valeur entière non signée. Cette valeur décroît d'une unité chaque fois que le routeur transmet le datagramme IPv6. Lorsque la valeur atteint 0, le périphérique qui gère le datagramme l'écarte. Ainsi, les datagrammes trop anciens n'encombrent pas inutilement le réseau, ce qui peut arriver lorsque l'on crée une boucle de routage. La valeur initiale est déterminée par l'expéditeur. Le champ Hop Limit est l'équivalent

du champ TTL d'IPv4. Mais la valeur du champ TTL correspondait le plus souvent à des secondes alors que la valeur du champ Hop Limit correspond à un nombre de pas. En fait, même avec TTL, le décompte pas à pas est pratique : les routeurs diminuent de 1 sa valeur à chaque pas, le temps de traitement du datagramme n'entrant plus en ligne de compte. Dans les faits, les protocoles tels que TCP emploient de volumineux champs Numéro de séquence (voir Chapitre 11) pour éviter le retour d'anciens paquets.

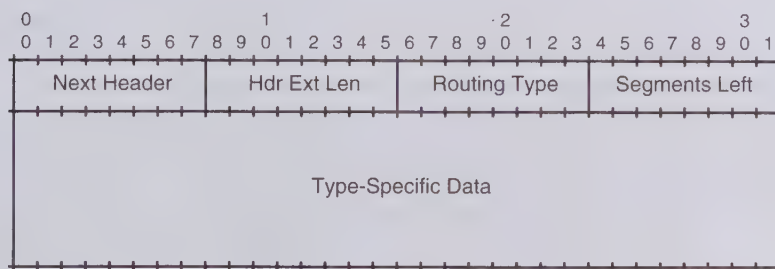
Les champs Adresse source et Adresse de destination sont des champs de 16 octets qui contiennent les adresses au nouveau format IPv6 (pour les adresses, reportez-vous Chapitre 4).

Les sections qui suivent détaillent les divers en-têtes d'extension d'IPv6.

• En-tête de routage (Routing)

Cet en-tête d'extension (voir Figure 15.4) sert à spécifier un certain nombre de routeurs par lesquels le datagramme devra impérativement passer avant d'atteindre sa destination. Cela répond à l'option de routage par la source que l'on trouvait déjà dans IPv4. On identifie l'en-tête de routage par la valeur 43 dans le champ Next Header de l'en-tête IPv6 de base.

Figure 15.4
L'en-tête d'extension Routing d'IPv6.



Voici les champs de l'en-tête de routage et leur signification :

- **Next Header.** Sélecteur de huit bits qui identifie le type de l'en-tête qui suit immédiatement l'en-tête de routage. Il peut s'agir d'un autre en-tête d'extension IPv6 ou d'un protocole de transport de couche supérieure (ULP, *Upper Layer Protocol*).
- **Hdr Ext Len.** Entier non signé de huit bits qui indique la longueur de l'en-tête de routage, c'est-à-dire le nombre d'unités de huit octets sans compter la première.



- **Routing Type.** Identificateur de huit bits correspondant à une variante particulière de l'en-tête de routage. On se sert couramment de la valeur 0 qui désigne la spécification IPv6 (RFC 1883). Les autres valeurs pourront servir pour les formats d'en-tête de routage qui s'en écartent.
- **Segments Left.** Entier de huit bits non signé qui spécifie le nombre de segments de route restants. Ce nombre correspond aux nœuds qui figurent explicitement dans la liste des nœuds intermédiaires qu'il reste à visiter avant d'atteindre la destination.
- **Type-Specific Data.** Champ à longueur variable dont le format est déterminé par la valeur du champ Routing Type. Cette longueur est calculée de manière que la longueur totale de l'en-tête de routage soit un multiple de huit octets.

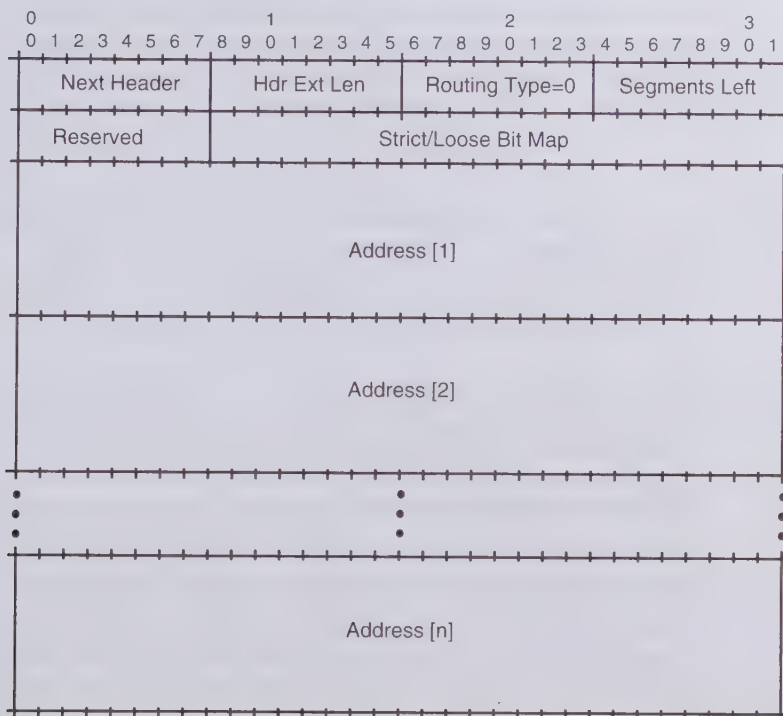
Lorsque le nœud d'un en-tête de routage dont le champ Routing Type présente une valeur inconnue, il traite le paquet en s'appuyant sur la valeur du champ Segments Left. Si cette dernière est zéro, le nœud ignore simplement l'en-tête de routage et passe immédiatement à l'en-tête suivant, identifié par le champ Next Header de l'en-tête de routage. Mais lorsque la valeur est différente de zéro, le nœud rejette le paquet et envoie à l'adresse d'origine un message ICMP Problème de paramètre, Code 0, correspondant à la valeur fautive dans Routing Type.

La Figure 15.5 présente un en-tête de routage avec Routing Type à 0. Remarquez que les 32 premiers bits de l'en-tête sont les mêmes que sur la Figure 15.4. Le champ Type-Specific Data contient la liste des adresses de routeurs qui permet au datagramme d'ignorer la route fixée par le protocole de routage et de lui préférer les spécifications faites au niveau du nœud d'origine.

Voici les champs de l'en-tête de routage IPv6 de type 0 et leur signification :

- **Next Header.** Sélecteur de huit bits qui identifie le type de l'en-tête qui suit immédiatement l'en-tête de routage.
- **Hdr Ext Len.** Entier non signé de huit bits. Ce champ contient la longueur de l'en-tête de routage, comptée en nombre d'unités de huit octets, les huit premiers octets exceptés. S'agissant d'un en-tête de routage de type 0, la longueur égale deux fois le nombre d'adresses qui figurent dans l'en-tête. C'est un nombre pair inférieur ou égal à 46.
- **Segments Left.** Entier non signé de huit bits. C'est le nombre de segments de route qui restent. Cette valeur est inférieure ou égale à 23 pour une route de type 0.
- **Reserved.** Champ réservé de huit bits qui vaut zéro à l'initialisation de la transmission et dont on ignore la valeur à la réception.
- **Strict/Loose Bit Map.** Ensemble de 24 bits numérotés de 0 à 23, de gauche à droite. A chaque bit correspond un segment de route. Le bit indique si l'adresse

Figure 15.5
 Un en-tête de routage IPv6
 avec Routing Type 0.



suivante doit forcément se trouver dans le voisinage de la précédente. Si sa valeur est 1, la route doit respecter strictement les indications du nœud d'origine : l'adresse suivante doit être au voisinage de la précédente. Si la valeur du bit est 0, la route peut s'écarter des indications de la source et l'adresse suivante ne sera par forcément au voisinage de la précédente.

- **Address[1..n].** Ensemble d'adresses sur 128 bits, numérotées de 1 à n.

• En-tête de fragmentation (Fragment)

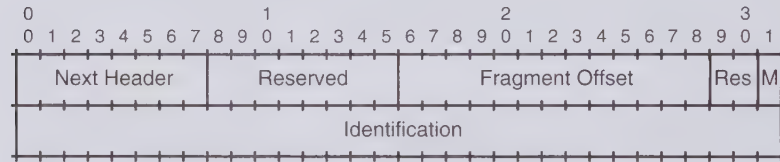
Contrairement à ce qui se passait dans IPv4, seuls les nœuds d'origine peuvent fragmenter les paquets IPv6, les routeurs ne peuvent plus le faire. Le nœud d'origine IPv6 connaît la taille de MTU. IPv6 implémente la découverte de MTU de chemin (RFC 1191 "Path MTU Discovery") pour connaître la valeur de MTU jusqu'à la destination.



Le nœud d'origine IPv6 insère un en-tête fragmentation dans les paquets dont la taille dépasse la valeur de MTU. On identifie l'en-tête de fragmentation par la valeur 44 du champ Next Header dans l'en-tête précédent. La Figure 15.6 présente le format de l'en-tête de fragmentation.

Figure 15.6

*L'en-tête d'extension IPv6
Fragment.*



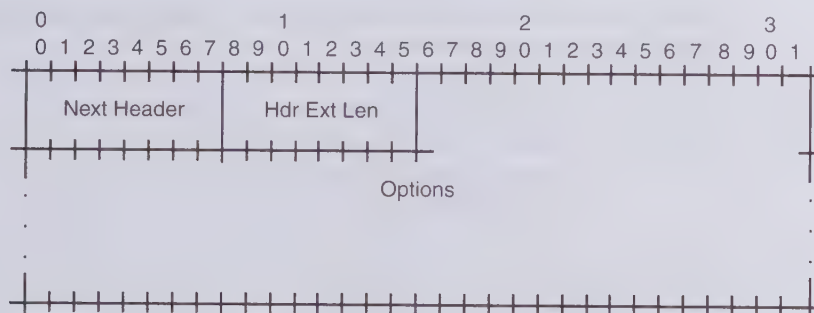
Voici les champs de l'en-tête de fragmentation et leur signification :

- **Next Header.** Sélecteur de huit bits qui identifie le type de l'en-tête qui suit immédiatement l'en-tête de fragmentation. Il peut s'agir d'un autre en-tête d'extension IPv6 ou d'un protocole de transport de couche supérieure (ULP).
- **Reserved.** Champ réservé de huit bits qui vaut zéro à l'initialisation de la transmission et dont on ignore la valeur au moment de la réception.
- **Fragment Offset.** Entier de 13 bits non signé qui indique le décalage des données qui font suite à l'en-tête (le nombre correspond à des unités de 8 octets). On mesure le décalage par rapport au début de la partie fragmentable du paquet d'origine.
- **Res.** Champ réservé de deux bits qui vaut zéro à l'initialisation de la transmission et dont on ignore la valeur au moment de la réception.
- **Flag M.** La valeur de ce champ est 1 pour tous les fragments, sauf le dernier pour lequel la valeur est 0.
- **Identification.** Champ de 32 bits. Pour chaque paquet qu'il doit fragmenter, le nœud d'origine génère un numéro d'identification. Il faut que les identificateurs permettent de distinguer les différents paquets fragmentés que ce nœud d'origine a pu envoyer avec les mêmes adresses dans les champs Adresse source et Adresse de destination.

• En-tête d'options de destination (*Destination Options*)

Cet en-tête sert à transporter des informations optionnelles qui n'intéressent que le nœud de destination. On identifie l'en-tête Destination Options avec la valeur 60 dans le champ Next Header de l'en-tête précédent (voir Figure 15.7).

Figure 15.7
L'en-tête d'extension IPv6
Destination Options.



Voici les champs que contient l'en-tête Destination Options et leur signification :

- **Next Header.** Sélecteur de huit bits qui identifie le type de l'en-tête suivant.
- **Hdr Ext Len.** Entier non signé de huit bits. Ce champ contient la longueur de l'en-tête d'options de destination exprimée en nombre d'unités de huit octets (sans compter les huit premiers).
- **Options.** Champ à longueur variable. La longueur totale de l'en-tête Destination Options est un entier multiple de huit (nombre d'octets).

Les valeurs dans Options servent pour l'en-tête Destination Options et pour l'en-tête Hop-by-Hop. Ces valeurs sont codées selon le format TLV (type, longueur, valeur). Le type occupe un octet, la longueur aussi. La longueur est celle des données qui suivent immédiatement. Les identificateurs de type sont conçus de manière que les deux bits de plus fort poids spécifient un traitement, au cas où le nœud IPv6 ne reconnaîtrait pas le type. Les codes correspondants sont présentés dans le Tableau 15.3.

Tableau 15.3 : Les deux bits de poids fort de l'octet qui spécifie le type (Option Type) dans IPv6

Bits	Signification
00	L'option sera ignorée et le traitement passera à la suite de l'en-tête.
01	Rejet du paquet.
10	Rejet du paquet. Même si l'adresse de destination est une adresse en multidiffusion, le nœud envoie à l'adresse d'origine un message ICMP Problème de paramètre, code 2, qui correspond à la valeur non reconnue.



Tableau 15.3 : Les deux bits de poids fort de l'octet qui spécifie le type (Option Type) dans IPv6 (suite)

Bits	Signification
11	Rejet du paquet. Si l'adresse de destination n'est pas en multidiffusion, le nœud envoie à l'adresse d'origine un message ICMP Problème de paramètre, code 2, qui contient la valeur non reconnue.

Le bit suivant de l'octet Option Type spécifie si les options sont susceptibles de changer en cours de route pendant l'acheminement du paquet vers l'adresse de destination :

- **Valeur 0.** Les données d'option ne changeront pas ;
- **Valeur 1.** Les données d'option sont modifiables.

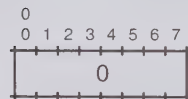
On emploie les options de remplissage Pad1 ou PadN (voir Figure 15.8) lorsqu'il faut aligner sur le même format les options qui suivent et remplir l'en-tête de façon que sa taille soit un multiple de huit octets.

L'option Pad1 introduit un seul octet de remplissage dans le champ Options de l'en-tête, PadN en introduit plusieurs.

Figure 15.8

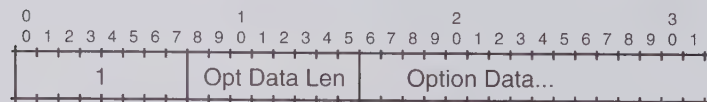
Les options de remplissage.

Option Pad1 (pas de nécessité d'alignement)



Note : le format de l'option Pad1 est spécial : pas de champs de longueur ni de valeur

Option PadN (pas de nécessité d'alignement)



• En-tête de fin d'en-tête (No Next)

Si le champ Next Header d'un en-tête IPv6 quelconque contient la valeur 59, cela signifie qu'aucun en-tête ne lui fait suite. Lorsque le champ Payload Length d'un en-tête IPv6 indique la présence d'octets après la fin de l'en-tête contenant 59 dans le champ Next Header, le nœud ignore ces octets et les transmet sans modification.

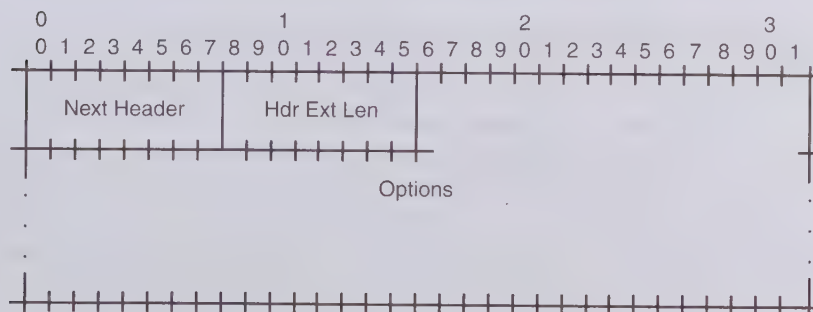
• En-tête d'options pas à pas (Hop-by-Hop Options)

Cet en-tête contient des informations qui intéressent tous les nœuds intermédiaires que rencontre le paquet pendant son acheminement. On spécifie l'en-tête Hop-by-Hop Options avec la valeur 0 dans le champ Next Header de l'en-tête IPv6 qui le précède. La Figure 15.9 présente le format de cet en-tête.

Voici les champs qu'il contient et leur signification :

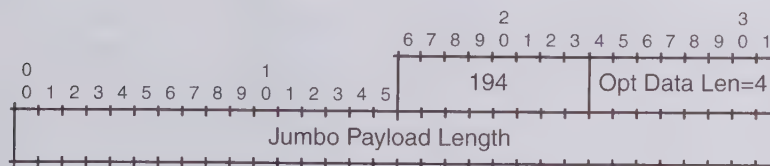
- **Next Header.** Sélecteur de huit bits qui identifie le type de l'en-tête qui suit.
- **Hdr Ext Len.** Entier non signé de huit bits. La longueur de l'en-tête est mesurée en unités de huit octets, sans prendre en compte les huit premiers.
- **Options.** Champ à longueur variable. La longueur totale du champ Hop-by-Hop Options doit être un multiple de huit octets. Le champ Options contient des valeurs codées au format TLV (type, longueur, valeur).

Figure 15.9
L'en-tête IPv6 Hop-by-Hop Options.



Outre les options Pad1 et PadN, on dispose ici de l'option Jumbo Payload (voir Figure 15.10).

Figure 15.10
L'option Jumbo Payload.



L'option Jumbo Payload s'aligne par rapport à l'en-tête sur le modèle $4n + 2$: la position de début doit pouvoir se calculer avec un multiple de quatre octets, auquel on ajoute deux.

Cette option est utile pour les paquets dont la taille excède 65 535 octets. Le champ Jumbo Payload Length occupe 32 bits et représente la longueur du paquet exprimée en octets, sans tenir compte de l'en-tête IPv6, mais en tenant compte de l'en-tête Hop-by-Hop Options. Cette longueur doit être supérieure à 65 535. Un paquet dont le champ Jumbo Payload Length contient une valeur inférieure ou égale à 65 535 entraîne l'envoi d'un message ICMP Problème de paramètre, Code 0, à l'adresse d'origine. Ce message indique l'octet de poids fort du champ fautif.

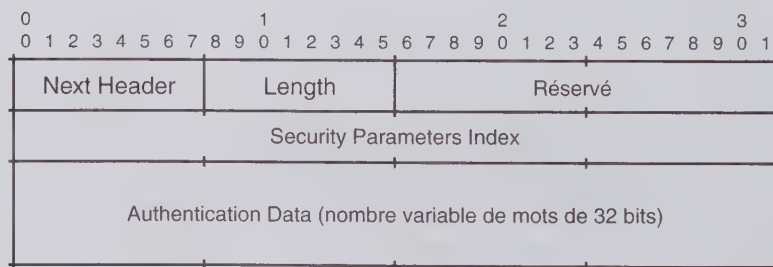
Par ailleurs, il faut affecter la valeur zéro au champ Payload Length de l'en-tête IPv6 pour tous les paquets avec option Jumbo Payload. Dans le cas contraire, le nœud renverra à l'adresse d'origine un message ICMP Problème de paramètre, Code 0. Ce message contiendra le champ Option Type.

• En-tête d'identification (Authentication)

Cet en-tête (voir Figure 15.11) permet d'assurer l'intégrité et l'identification des datagrammes IP. L'en-tête sécurise les datagrammes en y ajoutant des données d'identification. Ces données reposent sur tous les champs dont le contenu ne va pas changer au cours du transit. Les champs dont le contenu peut changer n'entrent pas dans le calcul des données d'identification (valeur zéro) : nombre de pas, TTL, etc. Pour spécifier l'en-tête d'identification, il faut affecter la valeur 51 au champ Next Header de l'en-tête qui le précède. Grâce à l'en-tête d'identification, on peut se rendre compte d'éventuelles altérations du paquet, mais les informations ne sont pas protégées par chiffrement. Pour chiffrer les paquets, il faut employer un en-tête Encrypted Security Payload (voir section suivante).

Figure 15.11

L'en-tête IPv6 Authentication.



Voici les champs de l'en-tête d'identification et leur signification :

- **Next header.** Sélecteur huit bits qui identifie le type de l'en-tête suivant.
- **Length.** Champ de huit bits qui spécifie la longueur du champ Authentication Data en nombre de mots longs (32 bits). La valeur minimum de Length est 0 et convient à des cas particuliers pour lesquels aucun algorithme d'identification n'est nécessaire.
- **Reserved.** Champ de 16 bits réservé pour un usage ultérieur. L'expéditeur lui affecte au départ des valeurs nulles. Mais la valeur finale, que le destinataire ignore toujours, sera prise en compte dans le calcul des données d'identification.
- **Security Parameters Index (SPI).** Valeur pseudo-aléatoire de 32 bits qui identifie la combinaison de sécurité du datagramme. Une valeur de 0 indique l'absence de sécurisation. Les valeurs de SPI comprise entre 1 et 255 sont réservées pour usage ultérieur par l'IANA. En général, l'IANA n'utilise pas les valeurs réservées, à moins qu'elles soient explicitement spécifiées par RFC.
- **Authentication Data.** Champ à longueur variable, mais toujours un multiple de 32 bits. Ce champ contient les données d'identification correspondant au paquet.

• En-tête de sécurité ou Encrypted Security Payload

Cet en-tête permet d'assurer l'intégrité et la confidentialité des datagrammes IP. Mais l'ESP (*Encrypted Security Payload*) n'offre aucune garantie de protection ou de non répudiation par les systèmes d'analyse de trafic. Avec certains algorithmes, l'en-tête d'identification (Authentication) permet d'assurer la non répudiation.

L'en-tête ESP ajouté à l'en-tête Authentication permet d'affiner l'identification. Lorsque la confidentialité des données n'est pas exigée, vous pouvez cependant tout à fait vous contenter de l'en-tête Authentication qui assure l'identification et protège l'intégrité des paquets.

ESP chiffre les données et les place dans la partie de l'en-tête réservée aux données. Vous pouvez vous servir de ce mécanisme aussi bien pour chiffrer un segment de couche de Transport qu'un datagramme complet.

ESP présente deux modes de fonctionnement :

- le mode tunnel ;
- le mode transport.

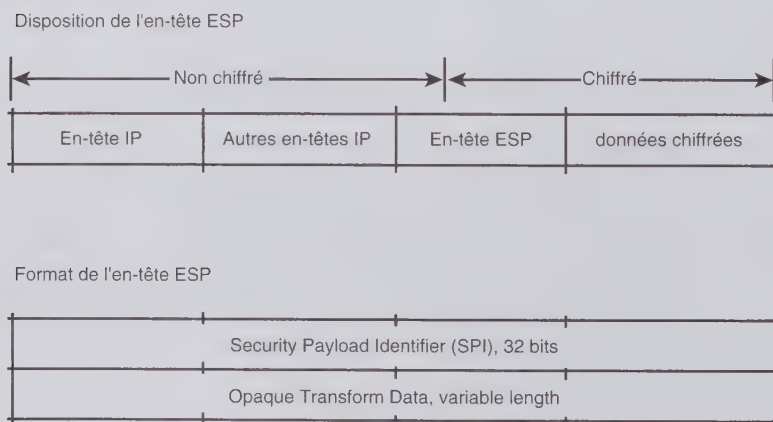


Dans le mode tunnel, on place le datagramme de départ dans la partie chiffrée de l'en-tête ESP, puis on incorpore la totalité de l'en-tête dans un nouveau datagramme dépourvu d'en-tête de sécurité. Les en-têtes IP, non chiffrés, permettent l'acheminement du datagramme chiffré jusqu'à sa destination. On peut introduire un en-tête de routage (Routing) non chiffré entre l'en-tête IP et l'en-tête ESP.

Dans le mode transport, on insère dans le datagramme IP l'en-tête ESP immédiatement avant l'en-tête Transport Layer Protocol. Ce mode ne surcharge pas la bande passante puisque l'on n'introduit ni en-têtes chiffrés ni options IP.

L'en-tête ESP peut se trouver n'importe où entre l'en-tête IP principal et l'en-tête final Transport Layer Protocol. Pour spécifier l'en-tête ESP, il faut affecter la valeur 52 au champ Next Header de l'en-tête qui le précède. L'en-tête ESP se compose d'un en-tête non chiffré suivi de données chiffrées. Les données chiffrées recouvrent les champs protégés de l'en-tête ESP et les données elles-mêmes, c'est-à-dire soit un datagramme complet, soit un protocole de couche de transport. La Figure 15.12 présente le diagramme haut niveau d'un datagramme IP sécurisé.

Figure 15.12
L'en-tête IPv6 ESP.



Les algorithmes de chiffrement ou d'identification, ainsi que le format précis des données OTD associées (*Opaque Transform Data*) sont appelés transformations. Le format ESP est conçu pour pouvoir prendre en charge les éventuelles nouvelles transformations pour s'adapter par exemple aux nouveaux algorithmes de chiffrement.

Le champ SPI (voir Figure 15.12) contient une valeur pseudo aléatoire de 32 bits qui identifie la combinaison de sécurité du datagramme. La valeur 0 dans le champ SPI indique l'absence de combinaison de sécurité. Ce champ est l'équivalent de l'identificateur SAID (*Security Association Identifier*) que l'on trouve dans

d'autre protocoles de sécurité. L'IANA se réserve les valeurs 0 à 255 pour un usage ultérieur. Le champ SPI est le seul qui soit obligatoire indépendamment de la transformation.

• Ordre des en-têtes d'extension IPv6

Lorsqu'un paquet contient plus d'un en-tête d'extension, voici l'ordre préconisé par les documents RFC :

1. IPv6.
2. Hop-by-Hop Options.
3. Destination Options.
4. Routing.
5. Fragment.
6. Authentication.
7. Encapsulating Security Payload.
8. Destination Options.
9. Upper-layer.

Chaque en-tête d'extension ne peut apparaître qu'une fois, sauf pour l'en-tête Destination Options dont on doit trouver deux exemplaires : un placé avant l'en-tête de routage (Routing) et un autre placé juste avant l'en-tête Upper-layer.

Lorsque l'en-tête de couche supérieure (Upper-layer) est lui-même un en-tête IPv6 (comme dans le cas de l'encapsulation d'un datagramme), il sera à son tour parfois suivi d'en-têtes d'extension. Ceux-ci doivent se conformer aux mêmes règles que les autres.

Même s'il est toujours préférable de suivre les recommandations des documents RFC, sachez que les nœuds acceptent de traiter les en-têtes d'extension quel que soit leur ordre, et même si une même extension apparaît plus d'une fois dans le même paquet. Toutefois, les nœuds refuseront toujours l'en-tête Hop-by-Hop Options s'il n'est pas placé comme il se doit, c'est-à-dire immédiatement après un en-tête IPv6.



IP sur ATM

La technologie ATM (*Asynchronous Transfer Mode* ou "Mode de transfert asynchrone") utilise la commutation de cellules pour fournir des débits importants et élargir la bande passante. L'élément fondamental d'un réseau ATM est le commutateur (voir Figure 15.13). Le support physique type d'un réseau ATM est constitué de fibres optiques. On groupe les fibres par paires : l'une permet de transporter les données entrant dans le commutateur, l'autre les données qui en sortent. Un commutateur ATM utilise des paquets de petite taille (53 octets), appelés cellules. Parmi ces 53 octets, 5 sont réservés à l'en-tête ATM et il en reste 48 pour les données. Ainsi, la portion efficace d'une cellule ATM est relativement importante : 90,6 %.

Les données de départ sont fragmentées en cellules et envoyées rapidement sur le réseau. La Figure 15.14 présente un réseau construit à l'aide de plusieurs commutateurs ATM. L'interface entre commutateurs s'appelle NNI (*Network to Network Interface* ou "Interface réseau/réseau"). L'interface entre la machine d'extrémité et le commutateur ATM s'appelle UNI (*User to Network Interface* ou "Interface utilisateur/réseau").

Avant d'aborder le transport des datagrammes IP sur un réseau ATM, il faut compléter vos connaissances sur ATM, en particulier ce qui concerne les circuits ATM et la couche d'adaptation. C'est l'objet des sections qui suivent.

Figure 15.13

Un commutateur ATM.

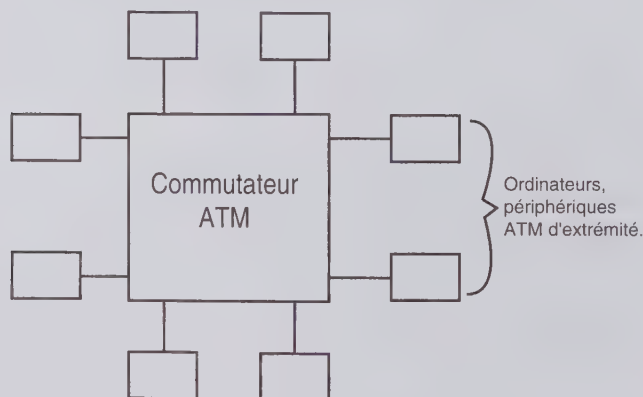
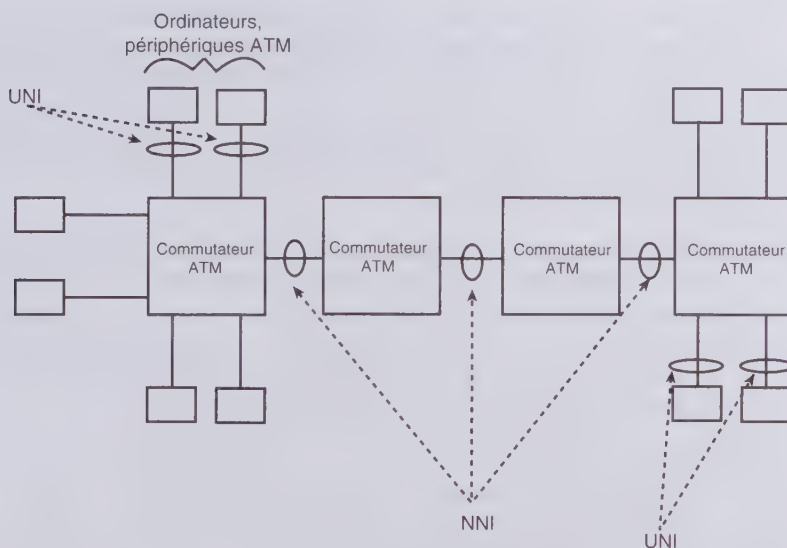


Figure 15.14
Un réseau ATM et ses interfaces.



• Les circuits ATM

Entre deux nœuds IP, ATM offre deux types d'interface orientées connexion :

- type SVC (*switched virtual circuit* ou "circuit virtuel commuté") ;
- type PVC (*permanent virtual circuit* ou "circuit virtuel permanent").

Un appel de type SVC est comparable à un appel téléphonique. Il y a d'abord une phase d'établissement de l'appel, puis une phase de maintien de connexion au cours de laquelle on transfère les données et, enfin, une phase de fin de connexion qui referme la connexion. Votre machine envoie d'abord une requête d'établissement de connexion au commutateur ATM local et spécifie l'adresse complète de la machine distante. Puis elle attend que le commutateur ait effectué l'appel. A travers plusieurs commutateurs ATM et grâce aux mécanismes de signalisation ATM, un circuit s'établit entre votre machine et la machine distante. Pendant la phase d'établissement de l'appel, chaque commutateur ATM examine la qualité de service (QoS, Quality of Service) requise par le circuit virtuel. Si le commutateur peut répondre à la demande, il transmet l'appel au commutateur suivant et devra consacrer à cet appel les ressources matérielles et logicielles nécessaires.



L'interface UNI utilise un identificateur de 24 bits pour chaque circuit virtuel accepté par le commutateur ATM local. Les cellules ne contiennent pas l'adresse d'origine et l'adresse de destination. En fait, l'expéditeur marque la cellule, à l'aide d'un identificateur, au moment de sa sortie et chaque commutateur ATM qui la reçoit la marque à nouveau. Il y a donc un identificateur ATM pour chaque pas. Par exemple, si l'expéditeur utilise 33 comme identificateur, il est possible que le destinataire utilise 108.

Mais ATM peut aussi servir à l'installation de circuits virtuels permanents (PVC) qui se comporteront comme des lignes téléphoniques spécialisées. C'est l'administrateur réseau qui établit les PVC et spécifie manuellement les adresses des points d'extrémité du circuit, les qualités de service et les identificateurs de 24 bits. Les circuits PVC sont surtout utiles lorsque l'on a besoin d'un flux de données continu (ou presque) entre deux points précis du réseau.

L'identificateur de 24 bits se compose de deux parties :

- un identificateur de chemin virtuel (VPI) de 8 bits ;
- un identificateur de circuit virtuel (VCI) de 16 bits.

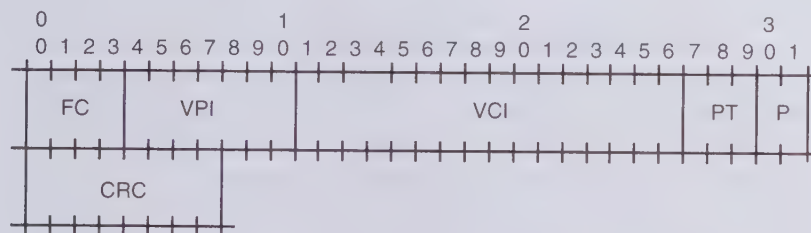
On appelle souvent l'identificateur de 24 bits le couple VPI/VCI. Le VPI caractérise plusieurs circuits virtuels qui utilisent le même chemin. Grâce au champ VPI, on peut router le trafic de données plus efficacement. Les fournisseurs commerciaux de services de communications pourront aussi se servir du champ VPI et facturer les connexions en fonction du chemin. Les utilisateurs auront toujours la possibilité d'adapter le multiplexage des circuits virtuels sur ce chemin.

Le couple VPI/VCI occupe une place prépondérante dans l'en-tête ATM qui ne comporte que cinq octets.

• La couche d'adaptation ATM (AAL)

Au plus bas niveau, ATM utilise des cellules de 53 octets. Fragmenter les datagrammes IP circulant sur un circuit ATM en des paquets aussi petits ne serait pas très efficace. C'est pourquoi la couche IP continuera à traiter les datagrammes normalement et passera le traitement à une couche d'adaptation ATM (AAL). Les fonctions de cette couche sont situées dans le firmware de la carte d'interface réseau ATM. Les pilotes logiciels qui s'exécutent sur la machine hôte interagissent aussi avec la couche d'adaptation (voir Figure 15.15).

Figure 15.15
En-tête de cellule ATM.



FC = contrôle de flux
 VPI = identificateur de chemin virtuel
 VCI = identificateur de circuit virtuel
 PT = type de payload
 P = priorité
 CRC = Cyclic Redundancy Checksum

Il existe cinq protocoles de couche d'adaptation, parmi lesquels il faut en choisir un. Ils sont numérotés d'AAL1 à AAL5. On s'intéresse aujourd'hui principalement à AAL1 et AAL5. AAL1 est utilisé pour les flux de données à débit constant, notamment pour le transport de l'audio et de la vidéo. AAL5 est utilisé pour les communications conventionnelles, telles que TCP/IP.

Le protocole AAL5 offre une interface IP dont la valeur de MTU vaut 65 535 octets. La Figure 15.16 présente le format AAL5. Remarquez que ce protocole n'utilise pas d'en-tête, mais plutôt un en-queue (*trailer*) de huit octets.

Le champ UU sert à transférer de manière transparente les informations qui doivent aller d'utilisateur à utilisateur. Ce champ n'ayant pas d'autre fonction dans le protocole, on peut lui affecter n'importe quelle valeur. Le champ CPI (*Common Part Identifier*) permet de remplir l'en-queue pour atteindre 64 bits. On affecte pour cela au champ CPI la valeur 0.

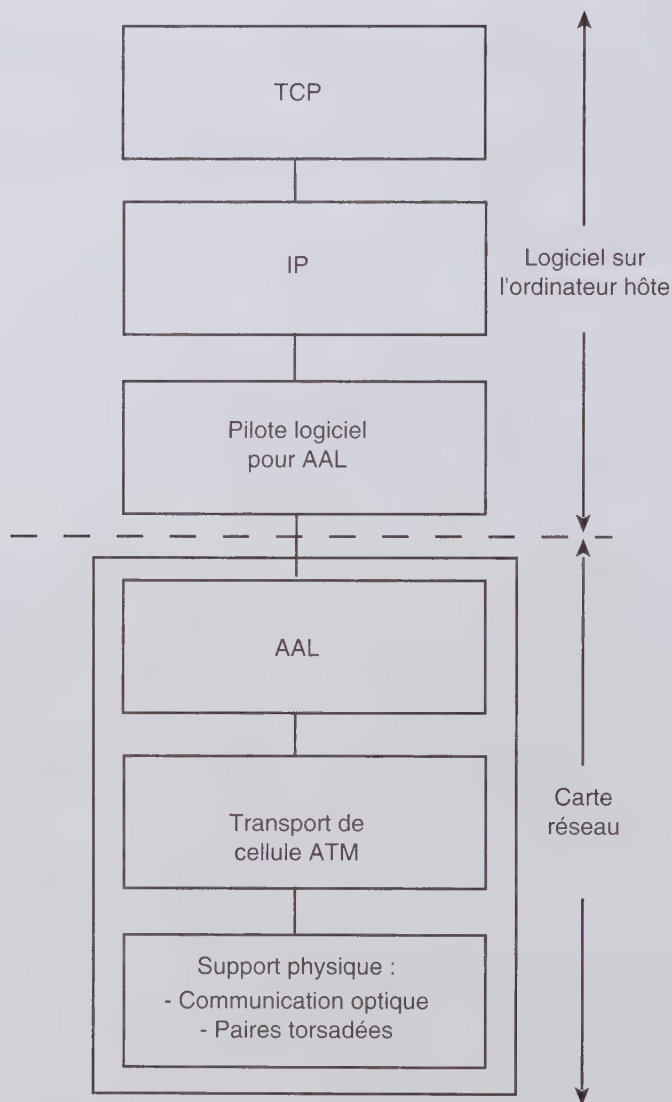
Le champ Length occupe 16 bits et contient le longueur du paquet AAL5.

Le champ CRC contient une valeur de 32 bits, utilisée par le protocole AAL5 pour détecter les erreurs dans le paquet qu'il reçoit.

Lorsque la couche AAL5 reçoit un datagramme IP, il l'ajoute devant l'en-queue de 8 octets et fragmente le tout en unités de 48 octets. On appelle ce processus la segmentation. C'est la carte d'interface réseau ATM qui s'en charge. La segmentation est donc un processus très rapide, puisqu'il a lieu au niveau matériel. Chaque



Figure 15.16
Une interface ATM.



unité de 48 octets sera ensuite envoyée avec un en-tête ATM de 5 octets, formant ainsi une cellule ATM complète de 53 octets. A la réception, AAL5 débarrasse les cellules de leur en-tête et assemble les unités de 48 octets pour reconstituer le datagramme IP de départ. On calcule alors de nouveau la valeur de CRC et on la compare à la valeur originale. Cela permet de mettre en évidence d'éventuelles erreurs de transmission. On appelle réassemblage l'opération qui consiste à

combiner les unités de 48 octets pour reconstituer le datagramme. Cette opération est, elle aussi, effectuée par la carte d'interface réseau ATM.

Lorsque la taille du datagramme plus les 8 octets de l'en-queue aboutissent à un multiple de 48 octets, toutes les cellules produites seront pleines. Dans le cas contraire, AAL5 remplit la dernière cellule avec des zéros de bourrage (entre 0 et 40) et ajoute bien sûr les 8 octets de l'en-queue (il convient de noter que l'avant-dernière cellule peut contenir de 0 à 7 octets de bourrage). AAL5 place toujours l'en-queue ATM à la fin de la dernière cellule ce qui permet à la couche AAL5 du destinataire de le retrouver rapidement, sans avoir à connaître la taille exacte du paquet.

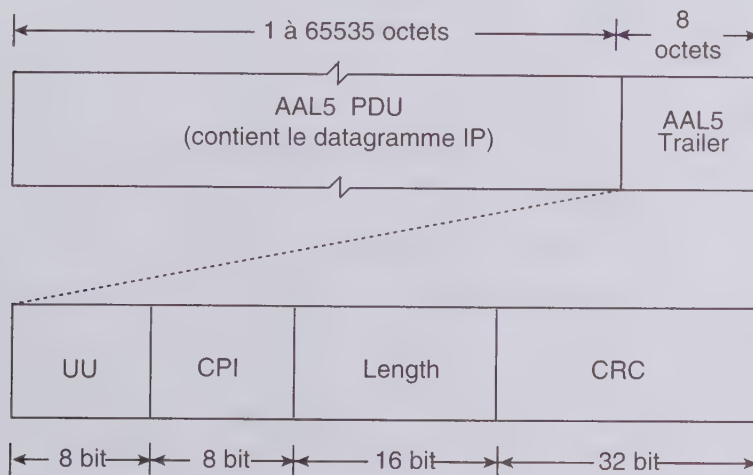
Pour reconnaître la dernière cellule, la couche AAL5 du destinataire reconnaît une valeur particulière dans le bit de poids faible du champ Payload Type de l'en-tête ATM.

• Le codage des types de données pour ATM

L'en-queue AAL5 ne dispose d'aucun champ pour spécifier le type des données transmises. Si le circuit ATM sert à transporter des données avec un seul protocole, il n'y aura aucun risque de confusion puisque l'expéditeur et le destinataire se seront mis d'accord au niveau du protocole. Mais dans un environnement multiprotocole, le besoin d'un champ spécifiant le type se fait sentir.

Dans ce cas-là, AAL5 utilise l'encapsulation SNAP (*subnetwork attachment point*, ou "point d'attachement de sous-réseau"). La Figure 15.17 présente l'encapsulation correspondant à IP.

Figure 15.17
Le format AAL5.



L'encapsulation SNAP obéit au standard IEEE 802.2. Elle se compose de l'en-tête IEEE 802.2 suivi de l'en-tête SNAP.

Voici les champs que contient l'en-tête IEEE 802.2 :

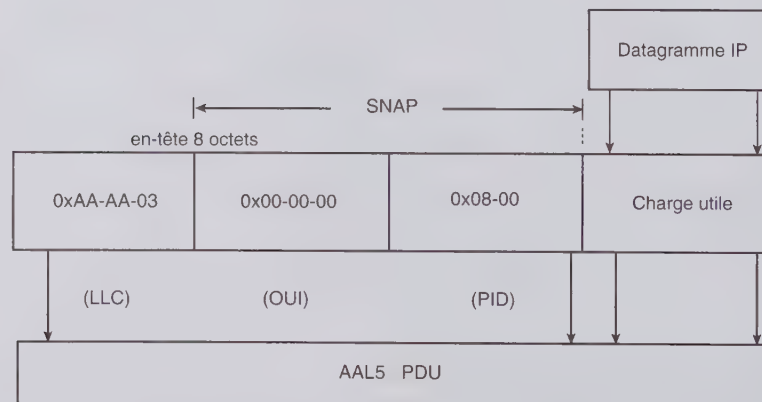
- **DSAP.** Champ de un octet qui spécifie le point d'accès du service de destination. La valeur AA (hexadécimal) caractérise l'encapsulation SNAP.
- **SSAP.** Champ de un octet qui spécifie le point d'accès du service source. La valeur AA (hexadécimal) caractérise l'encapsulation SNAP.
- **Control.** Champ de un octet qui indique le mode de communication. La valeur correspondante est 03 (hexadécimal) pour SNAP. Elle caractérise des données non numérotées (UI). AAL5 n'utilise pas de numéro pour reconnaître la séquence des paquets.

Voici maintenant les champs que contient l'en-tête SNAP qui suit immédiatement l'en-tête IEEE 802.2 :

- **OUI (Organizational Unit Identifier).** Champ de trois octets qui contient un code identifiant l'organisation ou l'autorité qui a structuré les données dans les champs qui suivent. Pour le format d'en-tête SNAP, la valeur OUI est 0.
- **Type.** Champ de deux octets qui spécifie le type de protocole encapsulé. La valeur est un type Ethernet (*Ethertype*). Pour les datagrammes IPv4 on aura 800 (hexadécimal) et pour les datagrammes IPv6 on aura 86DD (hexadécimal). Le document RFC 1700 précise les valeurs de type Ethernet pour les autres protocoles.

Figure 15.18

L'encapsulation SNAP pour utiliser IP sur un réseau ATM



Récapitulatif

Dans ce chapitre, nous vous avons présenté deux nouvelles technologies particulièrement intéressantes pour l'avenir des réseaux intranet et d'Internet IPv6 et ATM.

Nous avons vu quel était le format de l'en-tête du protocole IPv6 et quelle était la signification de ses différents champs. IPv6 ne se contente pas d'augmenter la taille du champ consacré aux adresses par rapport à IPv4 (il passe de 4 à 16 octets), mais ouvre de nouvelles possibilités grâce aux en-têtes d'extension. Ces en-têtes se trouvent après l'en-tête IPv6 de base et décrivent les couches de sous-protocole à l'intérieur de la couche IPv6. L'en-tête Destination Options permet par exemple de laisser le codage des données de son champ Data à la charge de l'expéditeur et du destinataire.

La technologie ATM fournit une interface commune pour les réseaux locaux et étendus. On sait que les réseaux locaux ou étendus utilisent traditionnellement des matériels et des technologies différentes. Grâce à ATM, on peut harmoniser ces réseaux, et pour des flux de données très variés tels que l'audio ou la vidéo. Dans ce chapitre, il a été question des bases élémentaires de cette technologie en ce qui concerne les réseaux IP.

Ces deux sujets mériteraient de plus amples développements. Mais nous ne pouvons pas faire mieux dans le cadre de cet ouvrage.



Testez vos connaissances

1. Pourquoi y a-t-il lieu de distinguer un en-tête de base des en-têtes d'extension dans un en-tête IPv6 ?
2. Quel genre d'informations figurent dans un en-tête IPv6 ? Faites le diagramme de l'en-tête IPv6 de base, avec le nom des différents champs et en expliquant brièvement à quoi ils servent.
3. Comment le démultiplexage entre les datagrammes IPv4 et IPv6 s'effectue-t-il ?
4. Pourquoi n'utilise-t-on pas le champ IP Version pour le démultiplexage entre IPv4 et IPv6 ?
5. Quel est le rôle du champ Priority dans l'en-tête IPv6 ? Décrivez les valeurs possibles.
6. Quel est le rôle du champ Next Header dans l'en-tête IPv6 de base et dans les en-têtes d'extension ?
7. Dressez la liste des en-têtes d'extension IPv6. Grâce à quel mécanisme IPv6 identifie-t-il les différents types d'en-tête d'extension ?
8. Quelle doit être la valeur de Next Header dans l'en-tête de base IPv6 pour qu'il encapsule un segment TCP ? Dessinez le diagramme de l'encapsulation.
9. Quelle doit être la valeur de Next Header dans l'en-tête de base IPv6 pour qu'il encapsule un segment UDP ? Dessinez le diagramme de l'encapsulation.
10. Quelles doivent être les valeur de Next Header dans un en-tête de base IPv6 et un en-tête de routage (Routing) pour qu'ils encapsulent un segment TCP ? Dessinez le diagramme de l'encapsulation.
11. Quelles doivent être les valeur de Next Header dans un en-tête de base IPv6, un en-tête Fragment et un en-tête de routage (Routing) pour qu'ils encapsulent un segment TCP ? Dessinez le diagramme de l'encapsulation.
12. Quel est l'ordre recommandé pour les différents en-têtes d'extension ? Que se passe-t-il si cet ordre n'est pas respecté ?
13. Pourquoi le champ Hop de l'en-tête IPv6 contient-il un nombre de pas et non plus une valeur TTL ?
14. Comparez les en-têtes IPv4 et IPv6. Quels sont les points communs et quelles sont les différences ?
15. Quel est le rôle de l'en-tête d'extension Routing ?
16. Dessinez le diagramme de l'en-tête d'extension Routing et expliquez la signification des différents champs.

17. Que se passe-t-il lorsqu'un nœud IPv6 rencontre une valeur qu'il ne reconnaît dans le champ Routing Type d'un en-tête de routage ?
18. Dessinez le diagramme de l'en-tête d'extension Routing lorsque Routing Type vaut 0. En quoi diffère-t-il du format de l'en-tête de routage utilisé dans le cas général ?
19. En quoi la fragmentation IPv6 est-elle différente de la fragmentation IPv4 ?
20. Quel est le rôle de l'en-tête d'extension Fragment ?
21. Dessinez le diagramme de l'en-tête Fragment et donnez la signification de ses différents champs.
22. Quel est le rôle de l'en-tête d'extension Destination Options ?
23. Dessinez le diagramme de l'en-tête Destination Options et donnez la signification de ses différents champs.
24. En quoi l'en-tête Destination Options peut-il étendre la fonctionnalité d'IPv6 ?
25. Quand utilise-t-on l'en-tête No Next ?
26. Quel est le rôle de l'en-tête d'extension Hop-by-Hop Options ?
27. Dessinez le diagramme de l'en-tête d'extension Hop-by-Hop Options et donnez la signification de ses différents champs.
28. Quel est le rôle de l'option Jumbo Payload dans l'en-tête Hop-by-Hop Options ? Décrivez le format de cette option.
29. Quel est le rôle de l'en-tête d'extension Authentication ?
30. Dessinez le diagramme de l'en-tête Authentication et donnez la signification de ses différents champs.
31. Quel est le rôle de l'en-tête d'extension Encrypted Security Payload ?
32. Dessinez le diagramme de l'en-tête Encrypted Security Payload et donnez la signification de ses différents champs.
33. Décrivez les deux modes de l'opération Encrypted Security Payload.
34. Quel est l'ordre des en-têtes d'extension IPv6 ?
35. Décrivez le format de base d'une cellule ATM.
36. Dans ATM, quelle est la différence entre l'interface UNI et l'interface NNI ?
37. Décrivez les deux types de circuits ATM.
38. Comment identifie-t-on un circuit ATM ? Quelles sont les deux composantes de cette identification ?



39. Dessinez le diagramme identifiant les composants de l'interface ATM dans un ordinateur où TCP/IP est en cours d'exécution.
40. Dessinez le diagramme du format d'une unité (PDU) de données pour le protocole AAL5. Expliquez les différents champs.
41. Grâce à quel mécanisme la station ATM du destinataire peut-elle identifier efficacement l'en-tête AAL5 ?
42. Comment identifie-t-on efficacement la dernière cellule d'une unité (PDU) AAL5 ?
43. Quels sont les mécanismes employés pour transmettre IP ou d'autres protocoles sur un réseau ATM ? Décrivez ce mécanisme pour des datagrammes IPv4 et IPv6.

ANNEXES

Protocoles standards 673





Protocoles standards

Cette annexe dresse la liste des protocoles standards, des ébauches de protocoles et des protocoles proposés, tels que définis dans la RFC 2200. Cette liste est présentée sous forme de tableaux afin que vous puissiez vous y référer facilement.

Le Tableau A.1 présente la liste des protocoles standards mentionnés par la RFC 2200.

Tableau A.1 : Protocoles standards

Protocole	Nom	Statut	RFC	STD
IP	Internet Official Protocol Standards	Req	2200	1
	Assigned Numbers	Req	1700	2
	Host Requirements – Communications	Req	1122	3
	Host Requirements – Applications	Req	1123	3
	Internet Protocol amendé par :	Req	791	5
	IP Subnet Extension	Req	950	5
	IP Broadcast Datagrams	Req	919	5
	IP Broadcast Datagrams with Subnets	Req	922	5
ICMP	Internet Control Message Protocol	Req	792	5
IGMP	Internet Group Multicast Protocol	Rec	1112	5
UDP	User Datagram Protocol	Rec	768	6
TCP	Transmission Control Protocol	Rec	793	7
TELNET	Telnet Protocol	Rec	854,855	8
FTP	File Transfer Protocol	Rec	959	9
SMTP	Simple Mail Transfer Protocol	Rec	821	10
SMTP-SIZE	SMTP Service Ext for Message Size	Rec	1870	10
SMTP-EXT	SMTP Service Extensions	Rec	1869	10
MAIL	Format des messages de courrier électronique	Rec	822	11



Tableau A.1 : Protocoles standards (*suite*)

Protocole	Nom	Statut	RFC	STD
CONTENT	Champ d'en-tête Type de contenu	Rec	1049	11
NTPV2	Network Time Protocol (Version 2)	Rec	1119	12
DOMAIN	Domain Name System	Rec	1034,1035	13
DNS-MX	Mail Routing and the Domain System	Rec	974	14
SNMP	Simple Network Management Protocol	Rec	1157	15
SMI	Structure of Management Information	Rec	1155	16
Concise-MIB	Concise MIB Definitions	Rec	1212	16
MIB-II	Management Information Base-II	Rec	1213	17
NETBIOS	NetBIOS Service Protocols	Ele	1001,1002	19
ECHO	Echo Protocol	Rec	862	20
DISCARD	Discard Protocol	Ele	863	21
CHARGEN	Character Generator Protocol	Ele	864	22
QUOTE	Quote of the Day Protocol	Ele	865	23
USERS	Active Users Protocol	Ele	866	24
DAYTIME	Daytime Protocol	Ele	867	25
TIME	Time Server Protocol	Ele	868	26
TFTP	Trivial File Transfer Protocol	Ele	1350	33
TP-TCP	ISO Transport Service on top of the TCP	Ele	1006	35
ETHER-MIB	Ethernet MIB	Ele	1643	50
PPP	Point-to-Point Protocol	Ele	1661	51
PPP-HDLC	PPP in HDLC Framing	Ele	1662	51
IP-SMDS	IP Datagrams over the SMDS Service	Ele	1209	52
POP3	Post Office Protocol (Version 3)	Ele	1939	53

Déclarations d'application : la documentation RFC 2200 contient quelques remarques concernant l'application d'un certain nombre de protocoles du Tableau A.

- IGMP ;
- SMI, MIB-II SNMP ;
- RIP ;
- TP-TCP.

Le Tableau A.2 dresse la liste des protocoles standards spécifiques à un réseau. Le statut de tous ces protocoles est électif.

Tableau A.2 : Protocoles standards spécifiques

Protocole	Nom	Statut	RFC	STD
IP-ATM	Classical IP and ARP over ATM	Prop	1577	
IP-FR	Multiprotocol over Frame Relay	Draft	1490	
ATM-ENCAP	Multiprotocol Encapsulation over ATM	Prop	1483	
IP-TR-MC	IP Multicast over	Prop Token-Ring LANs	1469	
IP-FDDI	Transmission of IP and ARP over FDDI Net	Std	1390	36
IP-X.25	X.25 and ISDN in the Packet Mode	Draft	1356	
ARP	Address Resolution Protocol	Std	826	37
RARP	A Reverse Address Resolution Protocol	Std	903	38
IP-ARPA	Internet Protocol	Std on ARPANET	BBN1822	39
IP-WB	Internet Protocol	Std on Wideband Network	907	40
IP-E	Internet Protocol	Std on Ethernet Networks	894	41
IP-EE	Internet Protocol on Exp. Ethernet Nets	Std	895	42
IP-IEEE	Internet Protocol	Std on IEEE 802	1042	43
IP-DC	Internet Protocol	Std on DC Networks	891	44
IP-HC	Internet Protocol	Std on Hyperchannel	1044	45
IP-ARC	Transmitting IP Traffic over ARCNET Nets	Std	1201	46
IP-SLIP	Transmission of IP over Serial Lines	Std	1055	47



Tableau A.2 : Protocoles standards spécifiques (*suite*)

Protocole	Nom	Statut	RFC	STD
IP-NETBIOS	Transmission of IP over NETBIOS	Std	1088	48
IP-IPX	Transmission of 802.2 over IPX Networks	Std	1132	49
IP-HIPPI	IP over HIPPI	Draft	2067	

Déclarations d'application : la documentation RFC 2200 contient quelques remarques à propos de certains des protocoles du Tableau A.2.

On peut s'attendre à ce qu'un réseau prenne en charge au moins un réseau physique, avec, chaque fois, les protocoles appropriés qui figurent dans le Tableau A.2. On peut en effet opter pour tel ou tel réseau physique (*elective*). Mais une fois le choix effectué, il devient obligatoire (*required*) de choisir les protocoles correspondants tels qu'ils figurent dans la liste. Consultez les documents RFC 1122 et 1123 (*Host and Gateway Requirements*) pour obtenir des renseignements plus précis concernant les protocoles spécifiques d'un réseau ("couche Liaison de données").

Le Tableau A.3 dresse la liste des ébauches de protocoles standards.

Tableau A.3 : Ebauches de protocoles standards

Protocole	Nom	Statut	RFC
BOOTP	DHCP Options and BOOTP extensions	Recommandé	2132
DHCP	Dynamic Host Configuration Protocol	Elective	2131
	Clarifications and Extensions BOOTP	Elective	1542
DHCP-BOOTP	Interoperation Between DHCP and BOOTP	Elective	1534
MIME-CONF	MIME Conformance Criteria	Elective	2049
MIME-MSG	MIME Msg Header Ext for Non-ASCII	Elective	2047
MIME-MEDIA	MIME Media Types	Elective	2046
MIME	Multipurpose Internet Mail Extensions	Elective	2045
PPP-CHAP	PPP Challenge Handshake Authentication	Elective	1994
PPP-MP	PPP Multilink Protocol	Elective	1990
PPP-LINK	PPP Link Quality Monitoring	Elective	1989
COEX-MIB	Coexistence between SNMPV1 & SNMPV2	Elective	1908

Tableau A.3 : Ebauches de protocoles standards (suite)

Protocole	Nom	Statut	RFC
SNMPv2-MIB	MIB for SNMPv2	Elective	1907
TRANS-MIB	Transport Mappings for SNMPv2	Elective	1906
OPS-MIB	Protocol Operations for SNMPv2	Elective	1905
CONF-MIB	Conformance Statements for SNMPv2	Elective	1904
CONV-MIB	Textual Conventions for SNMPv2	Elective	1903
SMIV2	SMI for SNMPv2	Elective	1902
CON-MD5	Content-MD5 Header Field	Elective	1864
OSPF-MIB	OSPF Version 2 MIB	Elective	1850
STR-REP	String Representation...	Elective	1779
X.500syn	X.500 String Representation...	Elective	1778
X.500lite	X.500 Lightweight...	Elective	1777
BGP-4-APP	Application of BGP-4	Elective	1772
BGP-4	Border Gateway Protocol 4	Elective	1771
PPP-DNCP	PPP DECnet Phase IV Control Protocol	Elective	1762
RMON-MIB	Remote Network Monitoring MIB	Elective	1757
802.5-MIB	IEEE 802.5 Token Ring MIB	Elective	1748
BGP-4-MIB	BGP-4 MIB	Elective	1657
RIP2-MIB	RIP Version 2 MIB Extension	Elective	1724
RIP2	RIP Version 2-Carrying Additional Info	Elective	1723
RIP2-APP	RIP Version 2 Protocol App. Statement	Elective	1722
SIP-MIB	SIP Interface Type MIB	Elective	1694
	Def Man Objs Parallel-printer-like	Elective	1660
	Def Man Objs RS-232-like	Elective	1659
	Def Man Objs Character Stream	Elective	1658
SMTP-8BIT	SMTP Service Ext or 8-bit MIME transport	Elective	1652
OSI-NSAP	Guidelines for OSI NSAP Allocation	Elective	1629



Tableau A.3 : Ebauches de protocoles standards (*suite*)

Protocole	Nom	Statut	RFC
OSPF2	Open Shortest Path First Routing v2	Elective	1583
ISO-TS-ECHO	Echo for ISO-8473	Elective	1575
DECNET-MIB	DECNET MIB	Elective	1559
BRIDGE-MIB	BRIDGE-MIB	Elective	1493
NTPV3	Network Time Protocol (version 3)	Elective	1305
IP-MTU	Path MTU Discovery	Elective	1191
FINGER	Finger Protocol	Elective	1288
NICNAME	Whois Protocol	Elective	954

Déclarations d'application : le document RFC 2200 contient quelques remarques concernant certains protocoles du Tableau A.3.

Le protocole PPP est une méthode possible pour envoyer des paquets IP sur des lignes série, donc un réseau physique. Il est probable que PPP finira par devenir un protocole standard spécifique pour ce type de réseau.

Le Tableau A.4 dresse la liste des protocoles standards proposés.

Tableau A.4 : Protocoles standards proposés

Protocole	Nom	Statut	RFC
IPv6-Jumbo	TCP and UDP over IPv6 Jumbograms	Elective	2147
MAIL-SERV	Mailbox Names for Common Services	Elective	2142
URN-SYNTAX	URN Syntax	Elective	2141
RADIUS	Remote Authentication Dial In Service	Elective	2138
SDNSDU	Secure Domain Name System Dynamic Update	Elective	2137
DNS-UPDATE	Dynamic Updates in the DNS	Elective	2136
DC-MIB	Dial Control MIB using SMIv2	Elective	2128
ISDN-MIB	ISDN MIB using SMIv2	Elective	2127
ITOT	ISO Transport Service on top of TCP	Elective	2126
BAP-BACP	PPP-BAP, PPP-BACP	Elective	2125
VEMMI-URL	VEMMI URL Specification	Elective	2122
ROUT-ALERT	IP Router Alert Option	Elective	2113

Tableau A.4 : Protocoles standards proposés (suite)

Protocole	Nom	Statut	RFC
MIME-RELAT	MIME Multipart/Related Content type	Elective	2112
CIDMID-URL	Content-ID and Message-ID URLs	Elective	2111
MHTML	MIME E-mail Encapsulation	Elective	2110
HTTP-STATE	HTTP State Management Mechanism	Elective	2109
802.3-MIB	802.3 Repeater MIB using SMIv2	Elective	2108
PPP-NBFCP	PPP NetBIOS Frames Control Protocol	Elective	2097
TABLE-MIB	IP Forwarding Table MIB	Elective	2096
IMAPPOPAU	IMAP/POP AUTHorize Extension	Elective	2095
RIP-TRIG	Trigger RIP	Elective	2091
IMAP4-LIT	IMAP4 non-synchronizing literals	Elective	2088
IMAP4-QUO	IMAP4 QUOTA extension	Elective	2087
IMAP4-ACL	IMAP4 ACL Extension	Elective	2086
HMAC-MD5	HMAC-MD5 IP Auth. with Replay Prevention	Elective	2085
RIP2-MD5	RIP-2 MD5 Authentication	Elective	2082
RIPNG-IPV6	RIPng for IPv6	Elective	2080
URI-ATT	URI Attribute Type and Object Class	Elective	2079
GSSAP	Generic Security Service Application	Elective	2078
MIME-MODEL	Model Primary MIME Types	Elective	2077
RMON-MIB	Remote Network Monitoring MIB	Elective	2074
IPV6-UNI	IPv6 Provider-Based Unicast Address	Elective	2073
HTML-INT	HTML Internationalization	Elective	2070
DAA	Digest Access Authentication	Elective	2069
HTTP-1.1	Hypertext Transfer Protocol HTTP/1.1	Elective	2068
DNS-SEC	Domain Name System Security Extensions	Elective	2065
IMAPV4	Internet Message Access Protocol v4rev1	Elective	2060
URLZ39.50	Uniform Resource Locators for Z39.50	Elective	2056



Tableau A.4 : Protocoles standards proposés (*suite*)

Protocole	Nom	Statut	RFC
SNANAU-APP	SNANAU APPC MIB using SMIv2	Elective	2051
PPP-SNACP	PPP SNA Control Protocol	Elective	2043
RTP-MPEG	RTP Payload Format for MPEG1/MPEG2	Elective	2038
ENTITY-MIB	Entity MIB using SMIv2	Elective	2037
RTP-JPEG	RTP Payload Format for JPEG-compressed	Elective	2035
SMTP-ENH	SMTP Enhanced Error Codes	Elective	2034
RTP-H.261	RTP Payload Format for H.261	Elective	2032
RTP-CELLB	RTP Payload Format of Sun's CellB	Elective	2029
SPKM	Simple Public-Key GSS-API Mechanism	Elective	2025
DLSW-MIB	DLSw MIB using SMIv2	Elective	2024
IPV6-PPP	IP Version 6 over PPP	Elective	2023
MULTI-UNI	Multicast over UNI 3.0/3.1 based ATM	Elective	2022
RMON-MIB	RMON MIB using SMIv2	Elective	2021
802.12-MIB	IEEE 802.12 Interface MIB	Elective	2020
IPV6-FDDI	Transmission of IPv6 Packets Over FDDI	Elective	2019
TCP-ACK	TCP Selective Acknowledgement Options	Elective	2018
URL-ACC	URL Access-Type	Elective	2017
MIME-PGP	MIME Security with PGP	Elective	2015
MIB-UDP	SNMPv2 MIB for UDP	Elective	2013
MIB-TCP	SNMPv2 MIB for TCP	Elective	2012
MIB-IP	SNMPv2 MIB for IP	Elective	2011
MOBILEIPMIB	Mobile IP MIB Definition using SMIv2	Elective	2006
MOBILEIPAPP	Applicability Statement for IP Mobility	Elective	2005
MINI-IP	Minimal Encapsulation within IP	Elective	2004
IPENCAPIP	IP Encapsulation within IP	Elective	2003
MOBILEIPSUP	IP Mobility Support	Elective	2002
TCP SLOWSTART	TCP Slow Start, Congestion Avoidance	Elective	2001
BGP-COMM	BGP Communities Attribute	Elective	1997

Tableau A.4 : Protocoles standards proposés (suite)

Protocole	Nom	Statut	RFC
DNS-NOTIFY	Mech. for Notification of Zone Changes	Elective	1996
DNS-IZT	Incremental Zone Transfer in DNS	Elective	1995
SMTP-ETRN	SMTP Service Extension ETRN	Elective	1985
SNA	Serial Number Arithmetic	Elective	1982
MTU-IPV6	Path MTU Discovery for IP (Version 6)	Elective	1981
PPP-FRAME	PPP in Frame Relay	Elective	1973
IPV6-ETHER	Transmission IPv6 Packets Over Ethernet	Elective	1972
IPV6-AUTO	IPv6 Stateless Address Autoconfiguration	Elective	1971
IPV6-ND	Neighbor Discovery for IP (Version 6)	Elective	1970
PPP-ECP	PPP Encryption Control Protocol	Elective	1968
GSSAPI-KER	Kerberos Version 5 GSS-API Mechanism	Elective	1964
PPP-CCP	PPP Compression Control Protocol	Elective	1962
GSSAPI-SOC	GSS-API Auth for SOCKS (Version 5)	Elective	1961
LDAP-STR	String Rep. of LDAP Search Filters	Elective	1960
LDAP-URL	LDAP URL Format	Elective	1959
ONE-PASS	One-Time Password System	Elective	1938
TRANS-IPV6	Transition Mechanisms IPv6 Hosts/Routers	Elective	1933
AUTH-SOCKS	Username Authentication for SOCKS V5	Elective	1929
SOCKSV5	SOCKS Protocol Version 5	Elective	1928
WHOIS++M	How to Interact with a Whois++ Mesh	Elective	1914
WHOIS++A	Architecture of Whois++ Index Service	Elective	1913
DSN	Delivery Status Notifications	Elective	1894
EMS-CODE	Enhanced Mail System Status Codes	Elective	1893
MIME-RPT	Multipart/Report	Elective	1892
SMTP-DSN	SMTP Delivery Status Notifications	Elective	1891
RTP-AV	RTP Audio/Video Profile	Elective	1890



Tableau A.4 : Protocoles standards proposés (suite)

Protocole	Nom	Statut	RFC
RTP	Transport Protocol for Real-Time Apps	Elective	1889
DNS-IPV6	DNS Extensions to support IPv6	Elective	1886
ICMPv6	ICMPv6 for IPv6	Elective	1885
IPV6-Addr	IPv6 Addressing Architecture	Elective	1884
IPV6	IPv6 Specification	Elective	1883
HTML	Hypertext Markup Language-2.0	Elective	1866
SMTP-Pipe	SMTP Serv. Ext. for Command Pipelining	Elective	1854
MIME-Sec	MIME: Object Security Services	Elective	1848
MIME-Encyp	MIME: Signed and Encrypted	Elective	1847
WHOIS++	Architecture of the WHOIS++ service	Elective	1835
	Binding Protocols for ONC RPC Version 2	Elective	1833
XDR	External Data Representation Standard	Elective	1832
RPC	Remote Procedure Call Protocol V. 2	Elective	1831
	ESP DES-CBC Transform	Ele/Req	1829
	IP Authentication using Keyed MD5	Ele/Req	1828
ESP	IP Encapsulating Security Payload	Ele/Req	1827
IPV6-AH	IP Authentication Header	Ele/Req	1826
	Security Architecture for IP	Ele/Req	1825
RREQ	Requirements for IP Version 4 Routers	Elective	1812
URL	Relative Uniform Resource Locators	Elective	1808
CLDAP	Connection-less LDAP	Elective	1798
OSPF-DC	Ext. OSPF to Support Demand Circuits	Elective	1793
TMUX	Transport Multiplexing Protocol	Elective	1692
TFTP-Opt	TFTP Options	Elective	1784
TFTP-Blk	TFTP Blocksize Option	Elective	1783
TFTP-Ext	TFTP Option Extension	Elective	1782
OSI-Dir	OSI User Friendly Naming...	Elective	1781
MIME-EDI	MIME Encapsulation of EDI Objects	Elective	1767

Tableau A.4 : Protocoles standards proposés (*suite*)

Protocole	Nom	Statut	RFC
Lang-Tag	Tags for Identification of Languages	Elective	1766
XNSCP	PPP XNS IDP Control Protocol	Elective	1764
BVCP	PPP Banyan Vines Control Protocol	Elective	1763
Print-MIB	Printer MIB	Elective	1759
ATM-SIG	ATM Signaling Support for IP over ATM	Elective	1755
IPNG	Recommendation for IP Next Generation	Elective	1752
802.5-SSR	802.5 SSR MIB using SMIv2	Elective	1749
SDLC-SMIv2	SNADLC SDLC MIB using SMIv2	Elective	1747
BGP4/IDRP	BGP4/IDRP for IP/OSPF Interaction	Elective	1745
AT-MIB	Appletalk MIB	Elective	1742
MacMIME	MIME Encapsulation of Macintosh files	Elective	1740
URL	Uniform Resource Locators	Elective	1738
POP3-AUTH	POP3 AUTHentication command	Elective	1734
IMAP4-AUTH	IMAP4 Authentication Mechanisms	Elective	1731
RDBMS-MIB	RDMS MIB—using SMIv2	Elective	1697
MODEM-MIB	Modem MIB—using SMIv2	Elective	1696
ATM-MIB	ATM Management Version 8.0 using SMIv2	Elective	1695
SNANAU-MIB	SNA NAUs MIB using SMIv2	Elective	1666
PPP-TRANS	PPP Reliable Transmission	Elective	1663
	Postmaster Convention X.400 Operations	Elective	1648
TN3270-En	TN3270 Enhancements	Elective	1647
PPP-BCP	PPP Bridging Control Protocol	Elective	1638
UPS-MIB	UPS Management Information Base	Elective	1628
AAL5-MTU	Default IP MTU for use over ATM AAL5	Elective	1626
PPP-SONET	PPP over SONET/SDH	Elective	1619
PPP-ISDN	PPP over ISDN	Elective	1618



Tableau A.4 : Protocoles standards proposés (suite)

Protocole	Nom	Statut	RFC
DNS-R-MIB	DNS Resolver MIB Extensions	Elective	1612
DNS-S-MIB	DNS Server MIB Extensions	Elective	1611
FR-MIB	Frame Relay Service MIB	Elective	1604
PPP-X25	PPP in X.25	Elective	1598
OSPF-NSSA	The OSPF NSSA Option	Elective	1587
OSPF-Multi	Multicast Extensions to OSPF	Elective	1584
SONET-MIB	MIB SONET/SDH Interface Type	Elective	1595
RIP-DC	Extensions to RIP to Support Demand Cir.	Elective	1582
	Evolution of the Interfaces Group of MIB-II	Elective	1573
PPP-LCP	PPP LCP Extensions	Elective	1570
X500-MIB	X.500 Directory Monitoring MIB	Elective	1567
MAIL-MIB	Mail Monitoring MIB	Elective	1566
NSM-MIB	Network Services Monitoring MIB	Elective	1565
CIPX	Compressing IPX Headers Over WAM Media	Elective	1553
IPXCP	PPP Internetworking Packet Exchange Control	Elective	1552
SRB-MIB	Source Routing Bridge MIB	Elective	1525
CIDR-STRA	CIDR Address Assignment...	Elective	1519
CIDR-ARCH	CIDR Architecture...	Elective	1518
CIDR-APP	CIDR Applicability Statement	Elective	1517
	802.3 MAU MIB	Elective	1515
HOST-MIB	Host Resources MIB	Elective	1514
	Token Ring Extensions to RMON MIB	Elective	1513
FDDI-MIB	FDDI Management Information Base	Elective	1512
KERBEROS	Kerberos Network Authentication Ser (V5)	Elective	1510
GSSAPI	Generic Security Service API: C-bindings	Elective	1509
DASS	Distributed Authentication Security...	Elective	1507
	X.400 Use of Extended Character Sets	Elective	1502
HARPOON	Rules for Downgrading Messages...	Elective	149

Tableau A.4 : Protocoles standards proposés (*suite*)

Protocole	Nom	Statut	RFC
Mapping	MHS/RFC-822 Message Body Mapping	Elective	1495
Equiv	X.400/MIME Body Equivalences	Elective	1494
IDPR	Inter-Domain Policy Routing Protocol	Elective	1479
IDPR-ARCH	Architecture for IDPR	Elective	1478
PPP/Bridge	MIB Bridge PPP MIB	Elective	1474
PPP/IP MIB	IP Network Control Protocol of PPP MIB	Elective	1473
PPP/SEC MIB	Security Protocols of PPP MIB	Elective	1472
PPP/LCP MIB	Link Control Protocol of PPP MIB	Elective	1471
X25-MIB	Multiprotocol Interconnect on X.25 MIB	Elective	1461
SNMPv2	Introduction to SNMPv2	Elective	1441
PEM-KEY	PEM-Key Certification	Elective	1424
PEM-ALG	PEM-Algorithms, Modes, and Identifiers	Elective	1423
PEM-CKM	PEM-Certificate-Based Key Management	Elective	1422
PEM-ENC	PEM-Message Encryption and Auth	Elective	1421
SNMP-IPX	SNMP over IPX	Elective	1420
SNMP-AT	SNMP over AppleTalk	Elective	1419
SNMP-OSI	SNMP over OSI	Elective	1418
FTP-FTAM	FTP-FTAM Gateway Specification	Elective	1415
IDENT-MIB	Identification MIB	Elective	1414
IDENT	Identification Protocol	Elective	1413
DS3/E3-MIB	DS3/E3 Interface Type	Elective	1407
DS1/E1-MIB	DS1/E1 Interface Type	Elective	1406
BGP-OSPF	BGP OSPF Interaction	Elective	1403
	Route Advertisement In BGP2 and BGP3	Elective	1397
SNMP-X.25	SNMP MIB Extension for X.25 Packet Layer	Elective	1382
SNMP-LAPB	SNMP MIB Extension for X.25 LAPB	Elective	1381



Tableau A.4 : Protocoles standards proposés (*suite*)

Protocole	Nom	Statut	RFC
PPP-ATCP	PPP AppleTalk Control Protocol	Elective	1378
PPP-OSINLCP	PPP OSI Network Layer Control Protocol	Elective	1377
SNMP-PARTY-MIB	Administration of SNMP	Elective	1353
SNMP-SEC	SNMP Security Protocols	Elective	1352
SNMP-ADMIN	SNMP Administrative Model	Elective	1351
TOS	Type of Service in the Internet	Elective	1349
PPP-IPCP	PPP Control Protocol	Elective	1332
	X.400 1988 to 1984 downgrading	Elective	1328
	Mapping between X.400 (1988)	Elective	1327
TCP-EXT	TCP Extensions for High Performance	Elective	132
FRAME-MIB	Management Information Base for Frame	Elective	1315
NETFAX	File Format for the Exchange of Images	Elective	1314
IARP	Inverse Address Resolution Protocol	Elective	1293
FDDI-MIB	FDDI-MIB	Elective	1285
	Encoding Network Addresses	Elective	1277
	Replication and Distribute Operations d	Elective	1276
	COSINE and Internet X.500 Schema	Elective	1274
BGP-MIB	Border Gateway Protocol MIB (Version 3)	Elective	1269
ICMP-ROUT	ICMP Router Discovery Messages	Elective	1256
OSI-UDP	OSI TS on UDP	Elective	1240
STD-MIBs	Reassignment of Exp MIBs to Std MIBs	Elective	1239
IPX-IP	Tunneling IPX Traffic through IP Nets	Elective	1234
IS-IS	OSI IS-IS for TCP/IP Dual Environments	Elective	1195
IP-CMPRS	Compressing TCP/IP Headers	Elective	1144
NNTP	Network News Transfer Protocol	Elective	977

Enfin, nous avons pensé qu'il serait pratique de rassembler les options Telnet dans un tableau, avec leur état et leur statut (Tableau A.5).

Tableau A.5 : Options Telnet

Protocole	Nom	n°	Etat	Statut	RFC	STD
TOPT-BIN	Binary Transmission	0	Std	Rec	856	27
TOPT-ECHO	Echo	1	Std	Rec	857	28
TOPT-RECN	Reconnexion	2	Prop	Ele		
TOPT-SUPP	Suppress Go Ahead	3	Std	Rec	858	29
TOPT-APRX	Approx Message Size Negotiation	4	Prop	Ele		
Protocol	Name	#	State	Status	RFC	STD
TOPT-STAT	Status	5	Std	Rec	859	30
TOPT-TIM	Timing Mark	6	Std	Rec	860	31
TOPT-REM	Remote Controlled Trans and Echo	7	Prop	Ele	726	
TOPT-OLW	Output Line Width	8	Prop	Ele		
TOPT-OPS	Output Page Size	9	Prop	Ele		
TOPT-OCRD	Output Carriage-Return Disposition	10	Prop	Ele	652	
TOPT-OHT	Output Horizontal Tabstops	11	Prop	Ele	653	
TOPT-OHTD	Output Horizontal Tab Disposition	12	Prop	Ele	654	
TOPT-OFD	Output Formfeed Disposition	13	Prop	Ele	655	
TOPT-OVT	Output Vertical Tabstops	14	Prop	Ele	656	
TOPT-OVTD	Output Vertical Tab Disposition	15	Prop	Ele	657	
TOPT-OLD	Output Linefeed Disposition	16	Prop	Ele	658	
TOPT-EXT	Extended ASCII	17	Prop	Ele	698	
TOPT-LOGO	Logout	18	Prop	Ele	727	
TOPT-BYTE	Byte Macro	19	Prop	Ele	735	



Tableau A.5 : Options Telnet (*suite*)

Protocole	Nom	n°	Etat	Statut	RFC	STD
TOPT-DATA	Data Entry Terminal	20	Prop	Ele	1043	
TOPT-SUP	SUPDUP	21	Prop	Ele	736	
TOPT-SUPO	SUPDUP Output	22	Prop	Ele	749	
TOPT-SNDL	Send Location	23	Prop	Ele	779	
TOPT-TERM	Terminal Type	24	Prop	Ele	1091	
TOPT-EOR	End of Record	25	Prop	Ele	885	
TOPT-	TACACS User Identification	26	Prop	Ele	927	
TOPT-OM	Output Marking	27	Prop	Ele	933	
TOPT-TLN	Terminal Location Number	28	Prop	Ele	946	
TOPT-3270	Telnet 3270 Regime	29	Prop	Ele	1041	
TOPT-X.3	X.3 PAD	30	Prop	Ele	1053	
TOPT-NAWS	Negotiate About Window Size	31	Prop	Ele	1073	
TOPT-TS	Terminal Speed	32	Prop	Ele	1079	
TOPT-RFC	Remote Flow Control	33	Prop	Ele	1372	
TOPT-LINE	Linemode	34	Draft	Ele	1184	
TOPT-XDL	X Display Location	35	Prop	Ele	1096	
TOPT-ENVIR	Telnet Environment Option	36	Hist	Not	1408	
TOPT-AUTH	Telnet Authentication Option	37	Exp	Ele	1416	
TOPT-ENVIR	Telnet Environment Option	39	Prop	Ele	1572	
TOPT-TN3270E	TN3270 Enhancements	40	Prop	Ele	1647	
TOPT-AUTH	Telnet XAUTH	41	Exp			
TOPT-CHARSET	Telnet CHARSET	42	Exp		2066	
TOPT-EXTOP	Extended-Options-List	255	Std	Rec	861	32

Symboles

&&, opérateur	382
.rhosts, fichier de configuration	576

Numériques

100BASE-T (câblage)	94
100BASE-TF (Fast Ethernet)	110
100BASE-TX (Fast Ethernet)	109
100BASE-VG (Fast Ethernet)	106, 112
100BASE-VG (IEEE 802 12)	113
100VG-AnyLAN (IEEE 802 12)	113
10BASE2 (câblage)	88
10BASE5 (câblage)	83
10BASE-FB (réseaux)	101
10BASE-FL (réseaux)	101
10BASE-T (câblage)	90
10BASE-T (hub)	90
802 (IEEE)	71
802.3 (IEEE)	71



AAL, couche d'adaptation ATM	662
ABOR, commande FTP	582
Abstraction IP	241
ACCT, commande FTP	581
ACK	
cumulés TCP	468
flag TCP	459, 462
Acknowledgment Number, champ TCP	458
Acquittement d'état de liaison, paquet OSPF	434
Active Monitor Present (Token Ring)	125
Adresse matérielle	
champ ARP	198
mappage en adresses IP	156
Adresses IP	
affectation	179
à zéro	170
BSD Unix	174

classe C	182
classes d'adresses	159, 164, 358
configuration	183
de bouclage	172
de ce réseau	172
de diffusion dirigée	168
de diffusion limitée	170
DNS	193
doublons et ARP	212
exceptions	174
hostid	158
mappage des adresses matérielles	156
multidiffusion	175
NAT	180
netid	158
nombre d'adresses par classe	161
notation	166
notation décimale à points	161
pour le réseau	167
résolution d'adresses	191-239
source et destination, champs IP	266
spéciales	166
structure	157
Adresses IPv6	185
Adresses NSAP	155
Affectation des adresses IP	179
Agent de gestion	623
Agent utilisateur	551
Algorithmes	
ARP	204
de Bellman-Ford	408
de Ford-Fulkerson	408
de Jacobson	473
de Karn (TCP)	472
de retransmission adaptative (TCP)	471
de routage	379
de routage complet	382
de routage IP	318
SPF	425
ALLO, commande FTP	582
Analyseur de protocole	220
Anneau (câblage)	79
Annnonce de routeur, message ICMP	352
API (couche Application)	45
APOP, commande POP3	560



APPE, commande FTP	582
Archie	549
Area ID, champ OSPF	428
ARP	194-224
Adresse IP, champ	199
Adresse matérielle, champ	198
algorithme	204
démons de table de cache	210
diffusion initiale	221
DoD	49
doublons d'adresses	212
encapsulation	202
format du paquet	196
HLen, champ	198
Opération, champ	198
PLen, champ	198
principes de fonctionnement	201
problèmes potentiels	212
réponse	199, 224
requête ARP	195
requêtes	223
réseaux à ponts	211
structure	207
surveillance du réseau	207
table de cache	194, 202, 206, 235, 514
timeouts de la table de cache	208
timestamps	210
trace	220
trames ARP	194
trames Ethernet	222
Type de matériel, champ	197
Type de protocole, champ	198
ARPAnet	11
Associations	452
ATM	660-667
AAL	662
codage des types de données	665
commutateur	660
CPI, champ	663
CRC, champ	663
en-tête	662
NNI	660
PVC	661
SVC	661
UNI	660
UU, champ	663
Automates à états finis	478
AuType, champ OSPF	429



Balises HTML	602
Bande de base, réseaux	94
Base 2	365
BCN (Token Ring)	125
Bellman-Ford, algorithme de	408
BGP4	439
Binaire	161, 365
conversion	365
BIND	547
Bit moniteur (Token Ring)	124
BITNET	24
Bloc CIDR	390
BOOTP	511
adresse de réponse	512
adresse de requête	512
bootstrap	518
champs	516
déroulement de la procédure	517
format de message	515
IP	514
perte de messages	514
tags du champ Vendor Specific Area	519
timeout	515
UDP	514
Vendor Specific Area, champ	519
Bootstrap	255, 518, 591
Bouclage, adresse	172
Broadcast Voir Diffusion	
BSD Unix	174, 408, 575
Bus	78



Câblage	
100BASE-T	94
10BASE2	88
10BASE5	83
10BASE-T	90
anneau	79

bus	78	• Codes	
capacitance	94	• champ ICMP	335
coaxial	79	• de réponse FTP	583
connecteurs BNC	88	• de sécurité IP	271
étoile	77	• Collision detect (Ethernet)	82
fibre optique	80	• Communication avec le réseau/hôte	
inductance	94	• de destination interdit par l'administrateur,	
MDI	90	• message ICMP	345
Media Dependent Interface	90	• Commutateur ATM	660
paires torsadées	79	• Comptage en miroir	375
topologie logique	79	• Configuration	
topologie physique	79	• automatique	509-544
UTP	94	• routeurs	297
Capacitance	94	• Congestion	
Carrier Sense with Multiple Access		• d'un routeur	347
(Ethernet)	82	• TCP	472
CD (Ethernet)	82	• Connecteurs BNC	88
CDDI (FDDI)	136	• Connexion	
CDUP, commande FTP	581	• de contrôle (FTP)	580
CGI	613	• de données (FTP)	580
Champs		• physique, couches de protocoles	34
• d'adresses (OSI)	38	• TCP	452
• de contrôle (OSI)	38	• Control, champ SNAP	666
• de données (OSI)	38	• Convergence RIP	412
• FCS (trame)	103	• Couches	
Checksum, champ	266, 447	• couches DoD	
CIDR	357, 389	• Accès réseau	49
• bloc d'adresses	390	• Application	50
• masque	389	• Hôte à hôte	49
• notation	389	• Internet	49
Circuit virtuel, IEEE	74	• de protocoles, connexions physique	34
CL_TK (Token Ring)	125	• OSI	
Classes d'adresses	159, 164	• Application	45
• A	160	• API	45
• B	160	• Liaison	38
• bits significatifs	164	• trailer field	47
• C	160, 182, 378, 388, 390	• Physique	37
• D	160	• Présentation	45
• nombre d'hôtes	161	• Réseau	40
• selon premier nombre décimal	166	• Session	42
Classes de réseaux	358	• Transport	41
Clients		• TCP/IP	41
• BOOTP	511	• TCP/IP	33
• DHCP	521	• Coupe-feu Voir firewalls	
• FTP	585		
• réseaux	11		



- Courrier électronique 551
 - IMAP4 562-565
 - mail 553
 - POP3 558-562
 - SMTP 551-557
 - Coût des routes 310
 - CPI, champ ATM 663
 - CRC (OSI) 39
 - CRC, champ ATM 663
 - CREN 24
 - CSMA (Ethernet) 82
 - CSMA/CD (Ethernet) 83
 - CSNET 24
 - CWD, commande FTP 581
- 
- DARPA 10
 - Data Offset, champ TCP 459, 467
 - DATA, commande SMTP 556
 - Datagrammes IP
 - Adresses IP source et destination,
 - champs 266
 - Checksum, champ 266
 - DoD 52
 - format 248
 - Fragment Offset, champ 255
 - fragmentation 247
 - Identification, champ 254
 - IEEE 73
 - Internet Header Length, champ 251
 - options IP 266
 - préséance 251
 - Protocol, champ 261
 - Qualité de service, champ 248
 - roulage 295-324
 - taille 244
 - Total Length, champ 254
 - trace 279
 - TTL, champ 260
 - Type of Service, champ 251
 - Version, champ 249
 - DECnet 14
 - DELE, commande FTP 582
 - DELE, commande POP3 560
 - Démons (table de cache ARP) 210
 - Démultiplexage 53
 - Descripteurs NFS 599
 - Description de BDD OSPF (paquet) 430
 - Destination
 - Address, champ IPv6 649
 - inaccessible, message ICMP 340
 - Port, champ UDP 498
 - Détection
 - d'erreurs par ICMP 327
 - de collisions (Ethernet) 82
 - DHCP 521-540
 - allocation dynamique 522
 - allocation manuelle 522
 - champs 527
 - format de message 526
 - gestion des adresses 522
 - processus d'acquisition d'adresses 523
 - trace 530
 - valeur du champ Type 529
 - Diffusion 174, 392
 - adresse de diffusion dirigée 168
 - dans les sous-réseaux 393
 - diffusion ARP 195
 - dirigée 168, 393
 - limitée 170, 512
 - RARP 227
 - requête ARP 302
 - table de routage RIP 405
 - tempête de diffusion 416
 - DIX (Ethernet) 81
 - DNS 193, 545
 - BIND 547
 - domaines de premier niveau 547
 - résolution 547
 - résolveurs de noms 546
 - serveurs de noms 547
 - serveurs du domaine racine 550
 - TCP/IP 21
 - DoD
 - ARP 49
 - couches
 - Accès réseau 49
 - Application 50
 - Hôte à hôte 49
 - Internet 49
 - datagrammes 52
 - full-duplex 50
 - message 52
 - RARP 49
 - segment 52
 - TCP 49

trame	52
UDP	49
Domaine	
en erreur (Token Ring)	130
de premier niveau	547
Doublons d'adresses IP et ARP	212
DSAP (IEEE)	73
champ SNAP	666
Durée de vie	
d'un segment	462
maximale d'un segment, MSL	478
<i>Voir TTL</i>	
DVA <i>Voir Bellman-Ford</i>	



EARN, réseaux	25
Echo	
commande Telnet	574
ICMP	337
EGP, famille de protocoles	404
End Systems (OSI)	46
En-queue AAL5	665
En-têtes 882	551
d'extension IPv6	644
ESP	657
ET bit à bit	382
Etats de liaison	424
down	425
up	425
Ethernet	168
Carrier Sense with Multiple Access	82
CD	82
Collision detect	82
CSMA	82
CSMA/CD	83
détection de collisions	82
DIX	81
Ethernet II	102
Ethernet	55
IEEE	71
switch Ethernet	116
trames ARP	222

Ethertype	55
trame	103
Etoile, câblage en	77
Extrémité TCP	450



Fast Ethernet	
100BASE-TF	110
100BASE-TX	109
100BASE-VG	106, 112
FDDI	
CDDI	136
jeton	134
taille de MTU	246
TRT	136
TTRT	136
Fenêtre glissante TCP	449
Fichiers de configuration	
.rhosts	576
hosts	193
hosts.equiv	576
masque de sous-réseau	364
FIN, flag TCP	459, 462
Firewalls (message ICMP)	346
Flow Label, champ IPv6	646
Flux de données TCP	446
Ford-Fulkerson, algorithme de	408
Fragment Offset, champ IP	255, 256
Fragmentation	
Fragment Offset, champ	256
inconvenients	260
IP	247
problèmes	343
réassemblage	260
Frame Status (Token Ring)	122
Frontière de masque de sous-réseau	374
FTP	579-590
ABOR	582
accès anonyme	586
client	585
codes de réponse	583



FTP

commandes	581
ACCT	581
ALLO	582
APPE	582
CDUP	581
CWD	581
DELE	582
HELP	582
LIST	582
MKD	582
MODE	581
NLST	582
NOOP	582
PASS	581
PASV	581
PORT	581
PWD	582
QUIT	581
REIN	581
REST	582
RETR	581
RMD	582
RNFR	582
RNT0	582
SITE	582
SMNT	581
STAT	582
STO	581
STRU	581
SYST	582
TYPE	581
USER	581
connexion	
de contrôle	580
de données	580
exemple de session	586
principes	579
RFC	580
TCP/IP	9
utilisation	584
Fuber distributed Data Interface <i>Voir</i> FDDI	
Full-duplex	
DoD	50
OSI	42



gated	409
Gestion	
d'activité (OSI)	43
de réseau	623-640
MIB	630
modèle	623
SNMP	625
GET, méthode HTTP	607
GGP	404
Gopher	614
sélecteurs	616
Veronica	614
Group Address, champ IGMP	397
Groupes de multidiffusion	177



Hello, paquet OSPF	429
HELO, commande SMTP	556
HELP, commande FTP	582
Hexadécimal	162
HLen, champ ARP	198
Hold down	415
Hop Limit, champ IPv6	648
Hops <i>Voir</i> pas	
hostid	158
hosts, fichier de configuration	193
hosts.equiv, fichier de configuration	576
Hôte	
de destination inconnu, message ICMP	344
inaccessible, message ICMP	341
multiconnecté	297
réseaux	11
routes spécifiques à un hôte	314
source isolé, message ICMP	345
HTML	601, 602
balises	602
hyperliens	602

HTTP	607
CGI	613
GET	607
TCP/IP	9
Hub	90
Hyperliens	602



IAB	27
IANA	179
ICMP	325-356
annonce de routeur	352
caractéristiques	326
code	335
code 0	341
Décalage de fragment, champ	331
destination inaccessible	340
détection d'erreurs	327
échec source route	344
echo	337
encapsulation	329
en-tête	331
flags d'en-tête	331
hôte de destination inconnu	344
hôte inaccessible	341
hôte source isolé	345
Identification, champ	331
implémentation	326
Longueur d'en-tête, champ	331
Longueur totale, champ	331
paramètres facultatifs	337
Ping	330, 337, 374
port inaccessible	342
présentation	325
problèmes de fragmentation	343
problèmes non traités	327
protocole inaccessible	342
Protocole, champ	331
redirection	349
relations avec IP	330
remise	332

réseau	
de destination inconnu	344
inaccessible	341
RFC	327
sélection de routeur	352
services	332
source quench	347
TOS, champ	331
Total de contrôle, champ	331, 336
TTL, champ	331
type 0 et 8	337
type 3	340
type 3 code 1	341
type 3 code 13	346
type 3 code 14	346
type 3 code 15	346
type 3 code 2	342
type 3 code 3	342
type 3 code 4	343
type 3 code 5	344
type 3 code 6	344
type 3 code 7	344
type 3 code 8	345
type 3 codes 11-12	346
type 3 codes 9-10	345
type 4	347
type 5	349
type 9	352
Type, champ	334
Version, champ	331
violation de préséance d'hôte	346
Identification, champ IP	254
IEEE	
802	71
802.12	
100BASE-VG	113
100VG-AnyLAN	113
root repeater	113
802.3	71
circuit virtuel	74
datagramme	73
DSAP	73
Ethernet	71
LLC	73
LSAP	73
MAC	72



- services de type 1 73
- services de type 2 74
- services de type 3 75
- Token Ring 71
- IESG, Internet 25
- IETF, Internet 25
- IGMP 177, 396
 - Group Address, champ 397
 - groupes de multidiffusion 178
 - requête 397
 - Version, champ 397
- IGP 404
- IMAP4 562-565
 - diagramme d'états 564
- IMAP4 (RFC) 565
- Inductance, câble UTP 94
- Internet 10
 - IAB 27
 - IESG 25
 - IETF 25
 - IRSG 26
 - IRTF 25
- Internet Protocol, IP 8
- InternetHeader Length, champ IP 251
- Intranet 10
- IP 191
 - abstraction 241
 - Adresses IP source et destination, champs 248, 266
 - algorithme de routage 318
 - Checksum, champ 266
 - diffusion limitée 512
 - en-tête 249
 - format du datagramme 248
 - fragment offset, champ 255, 256
 - fragmentation 247
 - identification, champ 254
 - Internet Header Length, champ 251
 - Internet Protocol 8
 - IPv4 et IPv6 153
 - messages BOOTP 514
 - MTU 244
 - préséance 251
 - Protocol, champ 261
 - qualité de service, champ 248
 - relations avec ICMP 329
 - remise au mieux 244
 - séquençement 243
 - taille du datagramme 244
 - Total Length, champ 254
 - TTL, champ 260
 - Type of Service, champ 251
 - version, champ 249
- IPv4 646
 - Precedence, champ 646
 - TTL, champ 649
- IPv6 643-659
 - adresses 185
 - champs 649
 - champs de l'en-tête de routage, type 0 650
 - Destination Address, champ 649
 - en-tête 644
 - Authentification 656
 - de base 643
 - Destination Options 652
 - ESP 657
 - Fragment 651
 - Hop-by-Hop 655
 - No Next 654
 - Routing 649
 - Flow Label, champ 646
 - format d'en-tête 643
 - fragmentation 651
 - Hop Limit, champ 648
 - Jumbo Payload Length, option 656
 - Nest Header, champ 654
 - Next Header, champ 647
 - options de destination 652
 - ordre des en-têtes d'extension 659
 - Payload Length, champ 647
 - Priority, champ 645
 - remplissage 654
 - routage 649
 - Routing Type, champ 650
 - sécurité 657
 - Source Address, champ 649
 - valeurs du champ
 - Ethernet Type 645
 - Next Header 647
 - Priority 646
- IRS 462
- IRSG, Internet 26
- IRTF, Internet 25
- ISN 462
- ISS 462

J

JANET, réseaux	25
Jeton	
FDDI	134
OSI	42
Token Ring	120
Jumbo Payload Length, option IPv6	656

K

Kernel, mode	570
--------------------	-----

L

Length, champ TCP	467
Length, champ UDP	498
Liaisons point à point	373
Linemode, commande Telnet	575
LIST, commande FTP	582
LIST, commande POP3	559
LLC (IEEE)	73
Logout, commande Telnet	569
Loopback <i>Voir</i> Bouclage	
LSAP (IEEE)	73

M

MAC	193
MAC (IEEE)	72
MAIL FROM, commande SMTP	556
Masque CIDR	389
Masques de sous-réseau	362
aligné à l'octet	368
à longueur variable <i>Voir</i> VLSM	
comptage en miroir	375
conversions binaire-décimal	365
emplacement de stockage	364
frontière	374
non alignés à l'octet	369, 371
notation abrégée	365
principes	362
problèmes d'optimisation	372, 374
structure	362
VLSM	376
RFC	365
table de routage	379
MAU (Token Ring)	127
MDI	90
Messages	
DoD	52
ICMP	325-356
MIB	625, 630
identifiants	635
MIME	553, 608
MKD, commande FTP	582
MODE, commande FTP	581
Modèle	
de gestion de réseau	623
DoD	48
Moniteur actif (Token Ring)	124
Mount	597
MSL	462, 478
MSS	468
MTU	244
Multicast <i>Voir</i> Multidiffusion	



Multidiffusion	174, 392, 395, 426
adresses standards	175
groupes	177, 395
IGMP	396
niveaux	395
Multiplexage	
OSI	41
TCP	450
TCP/IP	53
Multistation Access Unit (Token Ring)	127



NAT	180
NAUN (Token Ring)	130
NAWS, commande Telnet	574
Négociation à trois temps, TCP	460
Nest Header, champ IPv6	654
Netid, adresse IP	158
Next Header, champ IPv6	647
Next hop router Voir Routeur de prochain pas	
NFS	594-600
descripteur	599
numéros de RPC	598
RPC	596
RPC NFS	597
TCP/IP	21
versions	595
Web NFS	599
XDR	596
NFSNET	24
NLST, commande FTP	582
NMI	625
NMS	623
NNI	660
Nœud	11
géré	625
NOOP, commande	
FTP	582
POP3	560
SMTP	556
Notation décimale à points	161
NSAP	155
NSFNET	405

Numéros	
de port usuel	457
de réseau	158
de séquence	449, 468
NVT	570
caractères de contrôle	572



OPEN, appel	455
Opération, champ ARP	198
Optimisation des masques de sous-réseau	372
Options IP	266
classes d'option	268
Fin de liste	269
Internet Timestamp	276
Loose Source Routing	273
pas d'opération	269
Record Source	272
Sécurité	270
Strict Source Routing	273
Ordre d'octets réseau	278
OSI	
champs	
d'adresses	38
de contrôle	38
de données	38
couches	
3	295
Application	45
Liaison	38
Physique	37
Présentation	45
Réseau	40
Session	42
Transport	41
CRC	39
End Systems	46
full-duplex	42
gestion d'activité	43
jetons	42
multiplexage	41
PCI	46
PDU	46
TCP/IP	17
trame	38
unité de dialogue	43

OSPF	296, 378, 424-438
Area ID, champ	428
AuType	429
caractéristiques	427
Checksum, champ	429
format de message	428
Packet Length, champ	428
paquets	
d'acquittement	434
d'annonce	434
de description de BDD	430
de mise à jour d'état de liaison	433
de requête d'état de liaison	432
Hello	429
Router ID, champ	428
table de routage	436
Type	428
Version, champ	428
zones OSPF	427



Packet Length, champ OSPF	428
Paquet de mise à jour d'état de liaison (OSPF) ...	433
PAR	447
Pas	309
PASS, commande FTP	581
PASS, commande POP3	560
Passerelle	142
PASV, commande FTP	581
Payload Length, champ IPv6	647
PCI (OSI)	46
PDU (OSI)	46
PING	330, 337, 374
P-IP	251
PLen, champ ARP	198
Point SNAP	666
Poison reverse	416
Ponts	
ARP	211
réseaux	139

source routing	145
spanning tree	146
transparent	145
POP3	558-562
commandes	
APOP	560
DELE	560
facultatives	560
LIST	559
NOOP	560
obligatoires	559
PASS	560
QUIT	560
RETR	560
RSET	560
STAT	559
TOP	560
UIDL	560
USER	560
états	
autorisation	561
mise à jour	562
transaction	562
réponses	561
RFC	558
PORT, commande FTP	581
Portmapper	597
Ports	
Port inaccessible, message ICMP	342
TCP	457
TCP/IP	61
Préambule (Trame)	102
Precedence, champ IPv4	646
Préséance d'un datagramme IP	251
PRG (Token Ring)	125
Priority, champ IPv6	645
Problèmes potentiels	325
Programmation dynamique	408
Protocol, champ IP	261
Protocole inaccessible, message ICMP	342
Protocoles	
à vecteur de distance	
inconvenients	407, 410
RIP	405-424
couches	34
de routage	296



états de liaison	424
historique	408
OSPF	424-438
de transfert	443-507
ID	262
TFTP	517
Proxy ARP	235, 383
inconvenients	385
réponses	385
Pseudo en-tête	
TCP	466
en-tête UDP	498
PSH, flag TCP	459, 464
Puissances de 2	366
push	447
PVC	661
PWD, commande FTP	582



QoS Voir Qualité de service	
Qualité de service	241
champ IP	248
QUIT, commande	
FTP	581
POP3	560
SMTP	556



Rapports d'erreur	325-356
RARP	225-235
DoD	49
fonctionnement	227
réponses	233
requêtes	227, 232
serveurs	230
tempête RARP	229
trace	230
rep	577
RCPT TP, commande SMTP	556

● Réassemblage des datagrammes IP	260
● Redirection, type ICMP	349
● REIN, commande FTP	581
● Remise	
au mieux	244
directe	299
ICMP	332
indirecte	299
● Répéteur	138
● Réponses	
ARP	199, 224
RARP	233
● Requêtes	
ARP	195, 223
d'état de liaison, paquet OSPF	432
RARP	227, 232
● Réseaux	
10BASE-FB	101
10BASE-FL	101
bande de base	94
BITNET	24
client	11
CREN	24
CSNET	24
DECnet	14
de destination inconnu, message ICMP	344
EARN	25
hôte	11
inaccessible, message ICMP	341
Internet	10
intranet	10
JANET	25
NFSNET	24
nœud	11
passerelle	142
pont	139
privés, inconvenients	182
répéteur	138
routeur	141
serveur	11
SNA	14
TCP/IP	8
● Réseau/hôte de destination inaccessible	
pour le TOS, message ICMP	346
● RESET, commande SMTP	556
● Résolution d'adresses	191-239
algorithme ARP	204
ARP	194
autres protocoles	235
DNS	193
doublons et ARP	212

fonctionnement de ARP	201	•	Root repeater (IEEE 802.12)	113
problème ARP	212	•	Routage	243, 273, 295-324
Proxy ARP	235	•	algorithme	318, 379
RARP	225	•	algorithme complet	382
table de cache	194	•	boucles de routage	393
Résolveurs de noms	546	•	configuration des routeurs	297
REST, commande		•	congestion d'un routeur	347
FTP	582	•	convergence RIP	412
RETR, commande		•	coût des routes	310
FTP	581	•	dynamique	296
POP3	560	•	enregistrement de route	273
RFC		•	erreurs signalées par ICMP	328
FTP	580	•	Loose Source Routing, option IP	273
ICMP	327	•	masques de sous-réseau	362, 379
IMAP4	565	•	OSPF	378, 424-438
masques de sous-réseaux	365	•	pas	309
POP3	558	•	principes	295
SMTP	551, 557	•	protocoles	295-296, 324
TCP/IP	27	•	Proxy ARP	383
RIP	296, 378, 405-424	•	remise	
convergence	412	•	directe	299
corps du paquet	419	•	indirecte	299, 308
diffusion de la table de routage	405	•	RIP	378, 405-424
distance	405	•	route par défaut	316
format de message	417	•	route spécifique à un hôte	314
gated	409	•	routeurs	
historique	408	•	de prochain pas	309
hold down	415	•	RIP	409
inconvenients	407, 410	•	transparents	386
nombre de pas infini	414	•	Source Quench	348
poison reverse	416	•	sous-réseaux	379
réponses	418	•	statique	296
requêtes	418	•	Strict Source Routing, option IP	273
RIPv2	439	•	systèmes autonomes	403
route par défaut	419	•	tables	308, 312
routed	408	•	hôte	311
routeurs passifs et actifs	409	•	OSPF	436
split horizon reverse	414	•	ToS	253
structure des entrées de table	405	•	traitement des datagrammes entrants	319
taille du message <i>vi</i>	420	•	TTL	307
trace	421	•	routed	408
rlogin	576	•	Router ID, champ OSPF	428
RMD, commande FTP	582	•	Routes par défaut	419
RNFR, commande FTP	582	•	Routeurs	141
RNTO, commande FTP	582	•	actifs	409
		•	passifs	409



Routing Type, champ IPv6	650
RPC	596
NFS	597
RSET, commande POP3	560
rsh	577
RST, flag TCP	459
RTO	471
RTT	471



Sécurité, utilitaires Berkeley	578
Segment	447
DoD	52
durée de vie	462
Sélection de routeur, message ICMP	352
SEND, appel	464
Séquence	449
Sequence Number, champ TCP	458
Séquencement	243, 446
Serveurs	11
BOOTP	511
de noms	547
DHCP	521
DNS du domaine racine	550
RARP	230
Services de type 1, IEEE	73
Services de type 2, IEEE	74
Services de type 3, IEEE	75
SGA, commande Telnet	574
SGML	602
SITE, commande FTP	582
SMNT, commande FTP	581
SMTP	551-557
commandes	
d'envoi	556
DATA	556
HELO	556
MAIL FROM	556
NOOP	556
QUIT	556
RCPT TP	556
RESET	556
en-têtes	551
messages de réception	556
RFC	551, 557

• SNA, réseaux	14
• SNAP	666
• champs	666
• Control, champ	666
• DSAP, champ	666
• SSAP, champ	666
• Trame	105
• SNMP	625
• commandes	632
• identifiants MIB	635
• messages	636
• MIB	630
• noeuds gérés	625
• principes de fonctionnement	630
• TCP/IP	18
• traps	633
• SNMP, protocole	623
• Source	
• Address, champ IPv6	649
• Port, champ UDP	498
• Quench, message ICMP	347
• route failure, message ICMP	344
• routing, pont	145
• Sous-réseaux	357-402
• algorithme de routage	382
• avantages	361
• diffusion	393
• frontière de masque	374
• liaison point à point	373
• masque de sous-réseau	362
• multidiffusion	395
• présentation	357
• problèmes d'optimisation	372, 374
• Proxy ARP	383
• routing	379
• taille	372
• Spanning tree (pont)	146
• SPF	424
• Split horizon reverse	414
• Spoofing	385
• SPREAD	404
• SRTT	471
• SSAP, champ SNAP	666
• STAT, commande	
• FTP	582
• POP3	559
• Stations sans disque	225
• Status, commande Telnet	574
• STD	27
• STO, commande FTP	581
• STOU, commande FTP	581

Stream IP	250
STRU, commande FTP	581
Structure des adresses IP	157
Surréseaux	357, 388-392
CIDR	389
Surveillance ARP	207
SVC	661
Switch Ethernet	116
SWS	450
SYN, flag TCP	459
SYST, commande FTP	582
Systèmes autonomes	403



Tables de cache ARP	194, 202, 206, 235, 514
timeouts	208
Tables de routage	308, 312
entrées RIP	405
masques de sous-réseaux	379
OSPF	436
route par défaut	316
route spécifique à un hôte	314
table d'hôte	311
Taille maximale de segment, MSS	468
Tampons d'heure <i>Voir</i> Timestamps	
TCB	477
TCP	8, 444-495
ACK cumulés	468
ACK, flag	459, 462
Acknowledgment Number, champ	458
acquitements cumulés	468
algorithme	
de Jacobson	473
de Karn	472
de retransmission adaptative	471
appels système	455
association	452
automates à états finis	478
caractéristiques	445
checksum, champ	447, 466
connexion	452

connexions	450, 452
Data Offset, champ	459, 467
décalage de données	459
DoD	49
extrémité	450
fenêtre glissante	449, 466
fermeture de connexion	455
fiabilité	447
FIN, flag	459, 462
flags	459
flux de données	446, 449
format des messages	457
initialisation de la connexion	460
Length, champ	467
multiplexage	450
négociation à trois temps	460
numéro de port	450, 457
numéro de séquence	468
OPEN, appel	455
options	467
ouverture de connexion	455
pseudo en-tête	466
PSH, flag	459, 464
RST, flag	459
rupture de connexion	477
SEND, appel	464
Sequence Number, champ	458
séquençement	446, 468
silly window	474
SYN, flag	459
timeouts adaptatifs	470
total de contrôle	466
trace	479
transfert de données de base	446
URG, flag	459, 465
Urgent Pointer, champ	465
TCP/IP	
ARPAnet	11
couche	33
couche Transport	41
DARPA	10
démultiplexage	53
DNS	21
Ethernet-II	102
FTP	9
HTTP	9



- multiplexage 53
- NFS 21
- OSI 17
- port 61
- réseaux 8
- RFC 27
- SNMP 18
- STD 27
- Telnet 9
- topologie physique 77
- trames 102
- UDP 9
- UNIX BSD 15
- Telnet 479, 566-575
 - architecture 569
 - caractères de contrôle NVT 572
 - commandes 574
 - Echo 574
 - Linemode 575
 - Logout 569
 - NAWS 574
 - SGA 574
 - Status 574
 - Terminal 574
 - TFC 575
 - Timing 574
 - Tspeed 575
 - Xdisploc 575
 - données de contrôle 571
 - négociation 570
 - des options 573
 - NVT 570
 - session 567
 - TCP/IP 9
 - telnetd 566
- Tempête
 - de diffusion 416
 - RARP 229
- Temps de réponse déterministe (Token Ring) 120
- Terminal
 - virtuel Voir NVT
 - commande Telnet 574
- TFC, commande Telnet 575
- TFTP 517, 591
 - messages TFTP 591
 - principes de fonctionnement 593
- Timeouts adaptatifs (TCP) 470
- Timestamps (Options IP) 276
- Timing, commande Telnet 574
- Token Ring
 - Active Monitor Present 125
 - BCN 125
 - bit moniteur 124
 - CL_TK 125
 - domaine en erreur 130
 - Frame Status 122
 - IEEE 71
 - jeton 120
 - MAU 127
 - moniteur actif 124
 - Multistation Access Unit 127
 - NAUN 130
 - PRG 125
 - taille de MTU 245
 - temps de réponse déterministe 120
- TOP, commande POP3 560
- Topologie
 - logique 79
 - physique 79
 - TCP/IP 77
- ToS Voir Type of service
- Total de contrôle 499
- Total Length, champ IP 254
- TP/IX 250, 251
- Trace
 - ARP 220
 - RARP 230
- Trailer field, couche Liaison 47
- Trames
 - ARP 194
 - champ FCS 103
 - DoD 52
 - Ethertype 103
 - OSI 38
 - préambule 102
 - SNAP 105
 - TCP/IP 102
- Transmission Control Protocol (TCP) 8
- Transparent, pont 145
- Traps SNMP 633
- TRT, FDDI 136
- Tspeed, commande Telnet 575
- TTL, champ IP 260
- TTL, champ IPv4 649
- TTRT (FDDI) 136
- TUBA 250

Type	
de matériel, champ ARP	197
de protocole, champ ARP	198
champ ICMP	334
champ OSPF	428
Type of Service, champ IP	251
TYPE, commande FTP	581

U

UDP	496-502
applications	497
caractéristiques	496
couches	499
Destination Port, champ	498
DoD	49
format d'en-tête	497
Length, champ	498
messages BOOTP	514
pseudo en-tête	498
Source Port, champ	498
TCP/IP	9
trace	500
UIDL, commande POP3	560
UNI	660
Unicast	174
Unité de dialogue (OSI)	43
UNIX	
BSD	15, 174
configuration des adresses IP	184
URG, flag TCP	459, 465
Urgent Pointer, champ TCP	465
USER, commande	
FTP	581
commande POP3	560
Utilitaires Berkeley	575-578
rcp	577
rlogin	576
rsh	577
sécurité	578
Utilitaires r*	575-578
UU, champ ATM	663
uuencode	552

V

Vendor Specific Area, champ BOOTP	519
Veronica	614
Version, champ IP	249
Violation de préséance d'hôte, message ICMP	346
VLSM	376, 427
RFC	377
VPI	662
VPIi	662

W

Web NFS	599
Windows NT, configuration des adresses IP	184
WWW	600
CGI	613
HTML	602
HTTP	607

X

Xdisplac, commande Telnet	575
XDR	596

Z

Zones OSPF	427
Area ID, champ OSPF	428



Dépôt légal : avril 2004
IMPRIMÉ EN FRANCE

Achevé d'imprimer le 9 avril 2004
sur les presses de l'imprimerie «La Source d'Or»
63200 Marsat
Imprimeur n° 9670

STUDENT EDITION

TCP/IP

La référence sur TCP/IP

Un ouvrage indispensable pour tout administrateur système souhaitant concevoir, installer et maintenir avec succès un réseau puissant et efficace. Vous trouverez dans *Le Student Edition TCP/IP* toutes les clés de l'administration en environnements multi-plates-formes : des conseils d'experts, une analyse approfondie des protocoles menée à partir de situations réelles, des scénarios de configuration et d'implémentation différents. Modèles en couches, adressage, routage, connectivité, sous-réseaux et surréseaux sont traités dans les moindres détails. Vous découvrirez en outre deux technologies prometteuses : IPv6, la nouvelle génération IP, et ATM.

Table des matières

- > Introduction à TCP/IP
- > Les couches de TCP/IP
- > Support réseau pour TCP/IP
- > L'adressage IP
- > Protocoles de résolution d'adresse
- > Internet Protocol
- > Le routage IP
- > Le protocole ICMP
- > Sous-réseaux et surréseaux
- > Protocoles de routage IP
- > Protocoles de transport
- > Configuration automatique
- > Services d'application
- > Gestion de réseau TCP/IP
- > IPv6 et ATM
- > Protocoles standards

Niveau : Débutant / Intermédiaire

Catégorie : Réseaux

Configuration : Multi-plates-formes

Cet ouvrage est une réédition du *Campus TCP/IP* aujourd'hui épuisé.

CampusPress

PEARSON

Education
France

CampusPress est une marque de
Pearson Education France
47 bis, rue des Vinaigriers
75010 Paris
Tél. : 01 72 74 90 00
Fax : 01 42 05 22 17
www.pearsoneducation.fr

ISBN : 2-7440-1653-5

COOP Maisonneuve

34.95\$

RÉGULIER

NON - MEMBRE

TCP/IP COLL. STUDENT EDITION

31.40\$

MEMBRE

SCO TIN ERPI DLM

06/06/2006 Ferme



636642