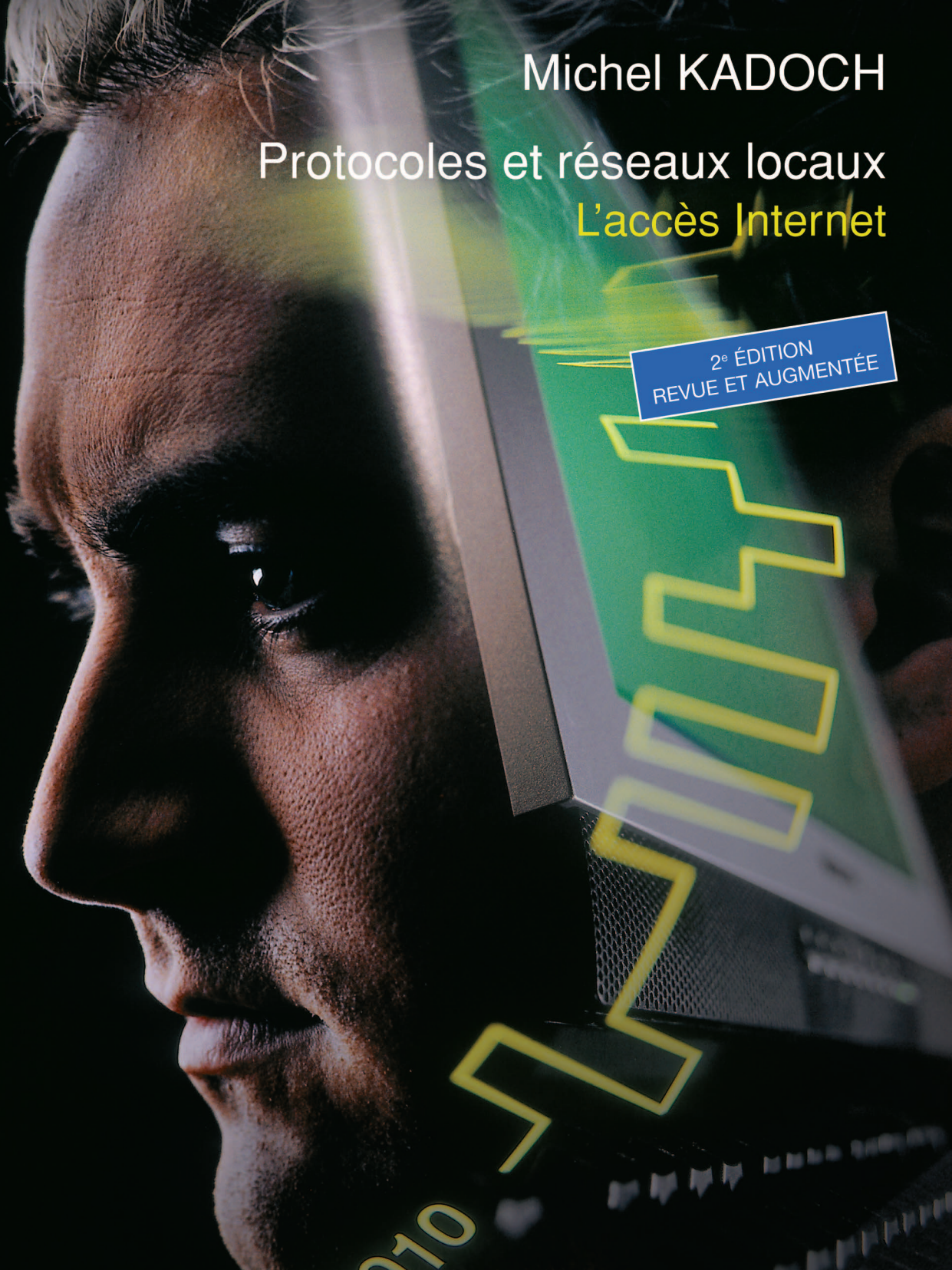
Michel KADOCH

Protocoles et réseaux locaux

L'accès Internet

2e ÉDITION
REVUE ET AUGMENTÉE



Université du Québec
École
de technologie
supérieure

Presses
de l'Université
du Québec

Protocoles
et réseaux locaux

Membre de L'ASSOCIATION NATIONALE DES ÉDITEURS DE LIVRES

Presses de l'Université du Québec
Le Delta I, 2875, boulevard Laurier, bureau 450, Québec (Québec) G1V 2M2
Téléphone : 418 657-4399 – Télécopieur : 418 657-2096
Courriel : puq@puq.ca – Internet : www.puq.ca

Diffusion/Distribution :

Canada et autres pays : Prologue inc., 1650, boulevard Lionel-Bertrand, Boisbriand (Québec) J7H 1N7 – Tél. : 450 434-0306 / 1 800 363-2864

France : Sodis, 128, av. du Maréchal de Lattre de Tassigny, 77403 Lagny, France – Tél. : 01 60 07 82 99

Afrique : Action pédagogique pour l'éducation et la formation, Angle des rues Jilali Taj Eddine et El Ghadfa, Maârif 20100, Casablanca, Maroc – Tél. : 212 (0) 22-23-12-22

Belgique : Patrimoine SPRL, 168, rue du Noyer, 1030 Bruxelles, Belgique – Tél. : 02 7366847

Suisse : Servidis SA, Chemin des Chalets, 1279 Chavannes-de-Bogis, Suisse – Tél. : 022 960.95.32

DANGER LE PHOTOCOPILLAGE TUE LE LIVRE

La *Loi sur le droit d'auteur* interdit la reproduction des œuvres sans autorisation des titulaires de droits. Or, la photocopie non autorisée – le « photocopillage » – s'est généralisée, provoquant une baisse des ventes de livres et compromettant la rédaction et la production de nouveaux ouvrages par des professionnels. L'objet du logo apparaissant ci-contre est d'alerter le lecteur sur la menace que représente pour l'avenir de l'écrit le développement massif du « photocopillage ».

Michel KADOCH

Protocoles et réseaux locaux

L'accès Internet

2e ÉDITION
Revue et augmentée

Université du Québec
École de technologie supérieure

Presses de l'Université du Québec

Catalogage avant publication de Bibliothèque
et Archives nationales du Québec et Bibliothèque et Archives Canada

Kadoch, Michel, 1944-

Protocoles et réseaux locaux

2e éd.

Comprend des réf. bibliogr. et un index.

Publ. en collab. avec : École de technologie supérieure.

ISBN 978-2-7605-3479-7 (Presses de l'Université du Québec)

ISBN 978-2-921145-79-4 (École de technologie supérieure)

1. Réseaux locaux (Informatique). 2. Protocoles de réseaux d'ordinateurs. 3. Internet. 4. Réseaux d'ordinateurs - Architectures. 5. Réseaux locaux (Informatique) - Problèmes et exercices. I. Université du Québec. École de technologie supérieure. II. Titre.

TK5105.7.K32 2012 004.6'8 C2012-940703-8

Les Presses de l'Université du Québec reconnaissent l'aide financière du gouvernement du Canada par l'entremise du Fonds du livre du Canada et du Conseil des Arts du Canada pour leurs activités d'édition.

Elles remercient également la Société de développement des entreprises culturelles (SODEC) pour son soutien financier.

Révision linguistique : CHARLES DADOUN
Révision scientifique : MARIA BENNANI
Mise en pages : DANIEL MARTEL
Couverture : YVES TOUGAS

2012-2.1 – *Tous droits de reproduction, de traduction et d'adaptation réservés*
© 2012 Presses de l'Université du Québec
© 2011 École de Technologie Supérieure

Dépôt légal – 2e trimestre 2012
Bibliothèque et Archives nationales du Québec / Bibliothèque et Archives Canada
Imprimé au Canada

dédié à

Stella
Arielle, Pascale et Valérie

Remerciements

La préparation d'un livre est un travail d'équipe. Quoique l'auteur ait la lourde tâche de coordonner l'ensemble de toutes les activités, il n'en demeure pas moins que la contribution de dessinateur et réviseurs est indispensable à la réussite du projet. Je les remercie du fond du cœur.

Michel Kadoch ing. PhD.
Montréal, Québec

Avant-propos

Il existe un très grand nombre d'ouvrages excellents en réseautique exposant le lecteur à tous les concepts fondamentaux lui permettant de maîtriser la matière. La question qui se pose est de savoir : pourquoi encore un nouveau livre? En quoi cette compilation de sujets diffère de ce que l'on trouve dans les autres volumes?
Tous les thèmes qui traitent des réseaux locaux et d'Internet ont connu une évolution fulgurante durant cette dernière décennie. Le cours « Protocoles et réseaux locaux » que j'ai eu le privilège d'enseigner pendant onze ans, traite de ces sujets. Chaque année, il y a eu des changements au contenu afin de pouvoir rester à la fine pointe de l'évolution dans ce domaine. Il a fallu compiler plusieurs notions et présenter plusieurs normes pour couvrir la matière du cours.

Ainsi, ce livre est le fruit de longues années d'enseignement à l'École de technologie supérieure. Il couvre le sujet propre à l'accès Internet dans sa complexité et dans sa variété de même qu'à Internet même. Dans de nombreux cours portant sur les réseaux, on enseigne la couche physique avec ses différentes approches, les couches liaison et réseau dans le cadre des réseaux étendus. Les réseaux locaux y sont aussi présentés. Malheureusement, il n'est pas possible de couvrir tous ces sujets par le détail dans le cadre d'une session de 45 heures de cours théorique. C'est pourquoi j'ai préféré offrir deux cours en réseaux, le premier portant sur les réseaux étendus et incluant tous les aspects théoriques et pratiques des différentes couches en rapport avec ces réseaux et le second portant sur les réseaux locaux ainsi que les protocoles directement reliés à Internet.

Comme toutes disciplines qui se respectent, le domaine de la réseautique exige de grandes connaissances et couvre un spectre de spécialités assez vaste. C'est un domaine qui passionne tant les informaticiens, que les ingénieurs en génie logiciel, les ingénieurs en télécommunications, de même que des techniciens en réseaux. Les domaines de recherche sur le sujet sont des plus variés. Il y a des travaux axés sur le développement de nouveaux services et d'autres portant sur l'amélioration des performances de réseaux.

J'ai été exposé à la réseautique en 1971 alors que j'avais commencé mon premier emploi à Communication Canada dans un département de développement de nouveaux services. La mission du groupe fut d'élaborer des concepts pour un réseau universitaire canadien semblable en tous points au nouveau réseau qui avait été conçu aux États-Unis et était connu sous le nom d'Arpanet. Notre réseau se nommait Canunet (Canadian University Network). Il nous a fallu bien évidemment saisir tous les éléments du réseau américain ainsi que tous les protocoles utilisés. En outre, il nous fallut accroître aussi nos connaissances en regard de la théorie des files d'attente. C'est alors que j'ai eu le plaisir d'assister à un cours de Leonard Kleinrock, l'un des pionniers d'Arpanet, traitant du sujet et ce, avant même la publication de son livre[1].

[1] Leonard Kleinrock, « Queuing System » vol.1, J. Wiley 1975 et vol. 2 J. Wiley 1976.

C'est à la lumière de cette expérience que les cours que j'ai enseignés durant ces années puis ce livre, ont vu le jour.

Organisation du livre

Le présent volume a été conçu de manière à offrir les connaissances fondamentales en ce qui a trait au domaine de la réseautique, et tout particulièrement, les réseaux locaux et l'accès à Internet.

Le chapitre 2 est une introduction au concept des couches et décrit le modèle de référence OSI. Ce chapitre sert de fondement à tous les autres chapitres et doit être maîtrisé avant de passer aux autres concepts.

Le chapitre 3 introduit la couche liaison pour les réseaux locaux. Il couvre précisément la sous-couche liaison LLC ainsi que l'accès point à point.

Le chapitre 4 traite des réseaux locaux basés sur les protocoles à compétition. L'accent est porté sur le protocole CSMA/CD et Ethernet.

Le chapitre 5 couvre les éléments essentiels permettant d'avoir une bonne compréhension des réseaux locaux sans fil, réseaux fondés sur la norme 802.11.

Le chapitre 6 nous expose aux réseaux locaux fonctionnant sur des topologies en anneaux. Le bus à jeton, l'anneau à jeton ainsi que le FDDI sont des réseaux locaux qui y sont traités.

Le chapitre 7 nous décrit l'interfonctionnement entre les réseaux locaux et résout les problèmes de boucle qui en résultent.

Le chapitre 8 est une brève introduction au concept du VLAN ou réseau local virtuel. Nous avons de plus en plus recours aux VLANs et il importe d'en connaître les principes.

Le chapitre 9 constitue l'un des chapitres les plus importants. Il couvre le protocole IP qui se trouve dans la couche réseau. Les concepts d'Internet et son évolution y sont décrits.

Le chapitre 10 couvre tous les éléments de soutien à IP. Nous y retrouvons une bonne représentation des concepts de : ARP, RARP, ICMP, DHCP, NAT, et le DNS ainsi que du IP mobile.

Le chapitre 11 traite du routage. Nous y retrouvons les protocoles de routage RIP et OSPF et BGP, d'algorithmes de routage ainsi que le Multicast (Multi distribution) et ses divers protocoles.

Le chapitre 12 fait état de la couche transport. Les protocoles UDP, TCP ainsi que des méthodes de contrôle de congestion y sont expliqués.

Le chapitre 13 couvre la couche application. Il nous donne quelques exemples retrouvés couramment dans un environnement Internet. Il s'agit des protocoles HTTP pour naviguer sur le Web, FTP et TFTP pour le transfert de fichier, ainsi que SMTP et POP3 pour le courrier électronique.

Enfin le chapitre 14, couvre le protocole SNMP utilisé pour la gestion de réseaux.

Lecteurs visés

Ce livre est écrit pour tous ceux qui désirent acquérir une bonne compréhension technique des concepts des réseaux locaux ainsi que d'Internet dans son ensemble. Il a été principalement conçu pour des cours universitaires et comprend assez de matière pour couvrir plus d'une session de 15 semaines.

Un cours de réseau, axé sur les réseaux locaux et Internet, devrait impérativement comprendre les chapitres 1, 2, 3, 4, 5, 7, 9, 10, 11. Le chapitre 6 sur les réseaux locaux à anneau peut être utile pour ceux qui ont à travailler avec ce genre de réseau. Le chapitre 8 est une source d'informations utiles sur les VLANs et devrait servir comme introduction au sujet. Les chapitres 12, 13 et 14 sont importants si le cours couvre aussi les couches supérieures et la gestion de réseaux.

Pour toute information supplémentaire concernant ce volume, consultez le site Web suivant :

http://www.lagrit.etsmtl.ca/livres/Protocoles_et_reseaux_locaux/index.html

Michel Kadoch ing. PhD.
Montréal, Québec

Table des matières

Remerciements i
Avant-propos iii
Table des matières vii

1 Télécommunications et réseaux 1

1.2 L'accès réseau 2

1.3 Types de réseau 7

1.3.1 Réseaux commutés 9
1.3.1.1 La commutation de circuits 9
1.3.1.2 La commutation par paquets 10
1.3.1.3 La commutation de messages 12
1.3.2 Acheminement dans les réseaux 13

1.4 Internet 14

1.5 Concepts de la transmission 16

1.5.1 Paires symétriques ou torsadées 16
1.5.2 Les câbles coaxiaux 17
1.5.3 Les fibres optiques 18
1.5.4 Propagation rayonnée 20
1.5.4.1 Les faisceaux hertziens 21
1.5.4.2 Les liaisons par satellite 21
1.5.5 La synchronisation 21
1.5.5.1 Synchronisation de bits 22
1.5.5.2 Synchronisation des octets 22
1.5.5.3 Synchronisation des trames 23
1.5.6 Transmission asynchrone 23
1.5.7 Transmission synchrone 23
1.5.8 Techniques de codage de données 24
1.5.9 Fréquence d'accès 24
1.5.10 Réseaux diffusés 24
1.5.11 Types d'informations 25

1.6 Codage des caractères et des symboles 25

1.7 Communications semi-duplex et duplex intégrale 25

1.8 Multiplexage par répartition dans le temps ou multiplexage temporel 26

1.9 Standardisation 28

1.10 Exercices 29

2 Modèle de référence OSI 31

2.1 Introduction 31

2.1.1 Optimisation des échanges 31
2.1.2 Indépendance du service 32
2.1.3 Restriction des modifications 32

2.2 Couches fonctionnelles 33

2.2.1 Homogénéité 33
2.2.2 Interactions entre couches adjacentes 33
2.2.3 Restriction du nombre de couches 34
2.2.4 Définition des couches 34
2.2.5 Couches inférieures 35
2.2.6 Couches supérieures 35
2.2.7 Les sept couches du modèle OSI 35

2.3 Communication et interaction 37

2.3.1 Concept de service 42
2.3.2 Point d'accès au service 43
2.3.3 Primitives 43
2.3.4 Protocoles 44

2.4 Les entités 44

2.5 Normalisation des couches 45

2.6 Encapsulation 46

2.7 Segmentation 48

2.8 Services et primitives 49

2.9 Appels avec ou sans connexion 51

2.9.1 Lien avec connexion 51

2.10 Classes de primitives 52

2.11 Les circuits virtuels 55

2.12 Modèle de référence Internet 56

2.13 Exercices 58

3 La couche liaison pour réseau local et point à point 63

3.1 Introduction 63

3.2 Les services du LLC 66

3.2.1 Service non confirmé 68
3.2.2 Service confirmé 68
3.2.3 Primitives de services sans connexion 69
3.2.4 Paramètres pour primitives de services sans connexion 69
3.2.5 Primitives de services permettant d'établir un appel de service avec connexion 70
3.2.6 Paramètres des primitives de services d'appel 71
3.2.7 Primitives permettant de procéder au transfert des données de service avec connexion 71
3.2.8 Paramètres des primitives de données 71
3.2.9 Primitives de déconnexion 72
3.2.10 Paramètres des primitives de déconnexion 72

3.3 LLC : caractéristiques de protocole 72

3.4 Format du LLC PDU 73

3.4.1 Absence de transparence dans les trames LLC 74
3.4.2 Le champ adresse 74
3.4.3 Le champ contrôle 75
3.4.3.1 La trame information 75
3.4.3.2 Le bit P/F 76
3.4.3.3 La trame supervision 76
3.4.3.4 Catégories de trames supervision 77
3.4.3.5 Trames non numérotées 78

3.5 Procédure du protocole LLC 79

3.5.1 Établissement de l'appel 79
3.5.2 Déconnexion de l'appel 80
3.5.3 Transmission des données 81
3.5.4 Accusé de réception et rectification des erreurs 82
3.5.4.1 La trame supervision RR 83
3.5.4.2 La trame supervision RNR 84
3.5.4.3 La trame supervision REJECT 86
3.5.4.4 Accusé de réception par la trame information 87
3.5.5 Compteurs de retransmission 88

3.6 Contrôle du flux **88**

3.7 LLC sans connexion **91**

3.8 Les protocoles de liaison SLIP et PPP **91**

3.8.1 Le protocole SLIP (Serial Line IP) **92**
3.8.2 Le protocole PPP (Point to point/Point à point) **93**
3.8.2.1 La trame HDLC dans le protocole PPP **95**
3.8.2.2 Système de fonctionnement des modules PPP **97**
3.8.2.3 Protocole LCP et établissement d'une liaison **98**
3.8.2.4 Les messages du protocole LCP **99**
3.8.2.5 La négociation du LCP **103**
3.8.2.6 L'authentification **105**
3.8.2.7 Protocole de contrôle de IP **106**

3.9 Exercices **107**

4 Protocoles à compétition **113**

4.1 Introduction **113**

4.2 Temps d'attente avant la retransmission **114**

4.3 Algorithme de temporisation de retransmission **114**

4.4 Longueur minimale de la trame **117**

4.5 Calcul de la longueur maximale du segment **118**

4.6 Fonctions de la sous-couche MAC **119**

4.6.1 Medium Access Management (MAM) **121**
4.6.2 Les trames **121**
4.6.3 Adresse MAC **124**
4.6.4 Trame de diffusion individuelle (Unicast) **124**
4.6.5 Trame de diffusion générale (Broadcast) **125**
4.6.6 Trame de multidistribution (Multicast) **125**

4.7 Taux de transmission paquets par seconde **125**

4.7.1 La couche physique **126**
4.7.2 Sous-couche de signalisation physique (PLS) **127**
4.7.3 Codage Manchester **128**
4.7.4 Fonction de l'unité de connexion au support (MAU) **129**
4.7.5 Types de conduit de transmission **130**

4.8 Fast Ethernet **130**

4.8.1 Modes duplex intégral (Full-Duplex) et semi-duplex (Half-Duplex) 131
4.8.2 La fonction pause 132

4.9 L'autonégociation 133

4.10 Configurations à 100 Mbps 138

4.10.1 Les codages du 100 Mbps 138
4.10.2 Media-Independant Interface (MII) 142

4.11 Étendue du réseau Ethernet 143

4.12 Gigabit Ethernet 145

4.12.1 Configuration à 1 Gbps 147

4.13 Exercices 148

ANNEXE 4A 151

Codes NRZ et NRZI 151

5 Réseaux locaux sans fil 153

5.1 Introduction 153

5.2 Service IEEE 802.11 156

5.3 La sous-couche MAC du IEEE 802.11 158

5.3.1 Vecteur d'allocation réseau 159
5.3.2 Les intervalles 159

5.4 Procédure d'accès DCF 160

5.4.1 Problèmes de station cachée et exposée : Utilisation de RTS et de CTS 163

5.5 Procédure d'accès PCF 167

5.6 Format des trames MAC 171

5.6.1 Le champ « contrôle de trame » 172
5.6.2 Champ « durée » 174
5.6.3 Les champs « adresse » 175
5.6.4 Le champ « contrôle de séquence » 176
5.6.5 Le champ « FCS » 176

5.7 Les trames de contrôle 177

5.8 Trames de gestion 177
5.9 Confidentialité et sécurité 179
5.10 Économie d'énergie 180
5.11 AP et cellules multiples 183
5.12 Exercices 188

6 Les réseaux à jeton 191
6.1 Introduction 191
6.2 Le bus à jeton (*Token Bus*) 191
6.2.1 Ajout d'une station 193
6.2.2 Retrait d'une station 195
6.2.3 Présence de plusieurs jetons 195
6.2.4 Initialisation de l'anneau 196
6.2.5 Format de trame du IEEE 802.4 196
6.2.6 Assignation des priorités 198
6.3 L'anneau à jeton 199
6.3.1 Ajout d'une station 205
6.3.2 Perte de jeton 205
6.3.3 La procédure à balisage (Beacon) 206
6.3.4 Priorité 207
6.3.5 Couche physique de l'anneau à jeton 207
6.4 Le réseau FDDI 208
6.4.1 Composantes du FDDI 209
6.4.2 Le protocole MAC du FDDI 210
6.4.3 Allocation de capacité 214
6.4.4 Surveillance de l'anneau 217
6.4.4.1 Réclamation du jeton 217
6.4.4.2 Initialisation 217
6.4.4.3 Vérification ou processus de balisage (Beacon Process) 218
6.4.5 Couche physique du FDDI 218
6.4.6 Reconfiguration de l'anneau après panne 219
6.4.7 Types de stations 221
6.5 Exercices 225
Bus à jeton 225
Anneau à jeton 227
FDDI 228

7 L'interfonctionnement 231
7.1 Introduction 231
7.2 Les concentrateurs 231
7.3 Les ponts (*Bridges*) 233
7.4 Tables d'acheminement 236
7.5 Fonction de pontage transparent (*Transparent Bridging*) 239
7.5.1 Acheminement des trames 241
7.6 Technique de pontage d'acheminement par la source 243
7.7 Algorithme *Spanning Tree* 246
7.7.1 Procédure de l'établissement de l'unique pont racine dans le réseau (Root Bridge) 248
7.7.2 Procédure de la définition des ports racine (Root Port) . 249
7.7.3 Procédure du choix du port désigné dans chaque pont (Designated Port) 249
7.7.4 Exemple d'application du protocole Spanning Tree 250
7.7.5 Les trames BPDU 253
7.7.6 Procédure dans le changement de topologie 255
7.8 Exercices 258

8 Les réseaux locaux virtuels (VLAN) 261
8.1 Introduction 261
8.1.1 Types de VLANs 264
8.1.2 Configuration des ports du commutateur 264
8.1.3 Adresse MAC de la station 265
8.1.4 Protocole de la couche 3 266
8.1.5 Sous-réseau IP 267
8.2 Étiquetage de VLAN selon le IEEE802.1Q 268
8.3 Exercices 272

9 Le protocole internet IP 275
9.1 Introduction 275

9.2 Modèles et protocoles **275**

9.2.1 Acheminement des datagrammes dans le réseau **279**
9.2.2 Encapsulation des messages **280**

9.3 Le protocole IPv4 ..... **283**

9.4 Unité de transfert maximum **286**

9.5 Fragmentation avec le protocole IP **287**

9.6 Adresse IPv4 ..... **290**

9.6.1 Classes d'adresses **291**
9.6.2 Assignation d'adresses **292**
9.6.3 Adresses réservées **293**
9.6.4 Adresse de sous-réseau **294**
9.6.5 Établissement de sous-réseau **296**
9.6.6 Masque à longueur variable **298**
9.6.7 L'adressage sans classe **302**
9.6.7.1 Exemple d'application CIDR **305**
9.6.7.2 Récapitulation des routes (Route Summarization) **306**

9.7 IPv6 ..... **308**

9.7.1 Extension d'en-tête IPv6 ..... **310**
9.7.2 Modèle d'adressage IPv6 (RFC 3513) **311**
9.7.3 Transition de IPv4 à IPv6 (RFC 4213) **314**

9.8 Exercices **316**

10 Support au protocole IP **321**

10.1 Introduction **321**

10.2 Protocole de résolution d'adresse ARP **322**

10.2.1 Format de paquet ARP **325**
10.2.2 Application du protocole ARP **326**
10.2.3 ARP mandataire (Proxy) **329**
10.2.4 ARP gratuit **331**

10.3 RARP **331**

10.4 ICMP (*Internet Control Message Protocol*) **332**

10.4.1 Réponse écho (Type 0) / Demande écho (Type 8)- Commande Ping **335**
10.4.2 Destination inaccessible (Type 3) **337**

10.4.3 Arrosage de la source (Source Quench) (Type 4) 339
10.4.4 Changement de route (Route Redirect) (Type 5) 339
10.4.5 Durée de vie écoulée d'un datagramme (Datagram Time Exceeded) (Type 11) 339
10.4.6 Problème de paramètre d'un datagramme (Type 12) 339
10.4.7 Demande et réponse d'estampille (Type 13) 340
10.4.8 Calcul de la durée de rotation RTT (Round Trip Time) ... 340
10.4.9 Demande et réponse d'information, (Types 15 et 16) 341
10.4.10 Demande et réponse de masque d'adresse, (Types 17 et 18)..... 341

10.5 Le protocole DHCP 341

10.5.1 Message DHCP 342
10.5.2 Options 344
10.5.3 Allocation dynamique des adresses réseau 345
10.5.4 Agent de relais DHCP 352
10.5.5 Autres messages DHCP 353

10.6 Traduction des adresses du réseau - NAT 354

10.6.1 Fonctionnement de NAT 357

10.7 Système de noms de domaine DNS 359

10.7.1 Format du message DNS 362
10.7.2 Opération du DNS 366
10.7.3 Commande nslookup 368

10.8 IP Mobile 370

10.9 Exercices 373

11 Les protocoles de routage 375

11.1 Introduction 375

11.2 Le protocole de routage RIP 377

11.2.1 Temporisateurs RIP 380
11.2.2 Format de message RIP version 1 380
11.2.3 Format du message RIP dans la version 2 382
11.2.4 Problèmes de convergence 383
11.2.5 Horizon coupé (Split Horizon) 385

11.3 Open Shortest Path First (OSPF) 386

11.3.1 Les aires de OSPF 387

11.3.2 Types de réseaux OSPF **389**
11.3.3 Les paquets de l'OSPF **390**
11.3.3.1 Les paquets Hello **392**
11.3.3.2 Les paquets Description de base de données (Database Description) **393**
11.3.3.3 Le paquet « Demande d'état de lien » (Link State Request) **395**
11.3.3.4 Le paquet « Mise à jour d'état de liens » (Link State Update) **396**
11.3.3.5 Le paquet « Accusé de réception d'état des liens » (Link State Acknowledgement) **396**
11.3.3.6 Annonce d'état de liens LSA (Link State Advertisement) **397**
11.3.4 La machine d'état des routeurs voisins **401**
11.3.5 Exemple d'établissement de contuguïté **403**
11.3.6 Fonctionnement de OSPF **405**

11.4 Border Gateway Protocol (BGP) **408**

11.5 Algorithmes de routage **411**

11.5.1 Algorithme de Dijkstra **411**
11.5.2 Algorithme de Bellman-Ford **415**

11.6 Multidistribution (*Multicast*) **419**

11.6.1 Besoin et contexte **420**
11.6.2 Le multicast sur réseau IP **420**
11.6.3 Adresse de groupe **423**
11.6.4 Distribution multicast **424**
11.6.5 Protocoles du routage multicast **427**
11.6.6 Le protocole IGMP **428**
11.6.6.1 Mécanisme de fonctionnement IGMP **429**

11.7 Résumé **430**

11.8 Exercices **432**

12 La couche transport **437**

12.1 Introduction **437**

12.2 UDP (*User Datagram Protocol*) **439**

12.3 TCP (*Transmission Control Protocol*) **441**

12.3.1 Services du protocole TCP **443**

12.3.2 Primitives de service TCP et IP **445**
12.3.3 Format d'en-tête TCP **445**
12.3.4 Établissement d'appel **448**
12.3.5 Échanges des données **450**
12.3.6 Libération de l'appel **451**
12.3.7 Machine d'état TCP **452**
12.3.8 Contrôle de flux **455**
12.3.8.1 Fenêtre coulissante **457**
12.3.9 Contrôle de congestion **461**
12.3.9.1 Démarrage lent (Slow Start) **462**
12.3.9.2 Évitement de la congestion (Congestion Avoidance) **464**
12.3.9.3 TCP Tahoe **465**
12.3.9.4 TCP Reno **466**
12.3.9.5 Retransmission rapide et récupération rapide- new Reno **466**
12.3.10 Temporisateurs, RTT (Round Time Trip) et RTO (Retransmission Timeout) **467**
12.3.10.1 Temporisateur de persistance (Persist Timer) **470**
12.3.10.2 Temporisateur de survie (Keepalive Timer) **470**

12.4 Exercices **472**

13 Les applications sur Internet **475**

13.1 Introduction **475**

13.2 Le protocole HTTP **475**

13.2.1 Session HTTP **476**
13.2.2 Connexion persistante **477**
13.2.3 Pipelining **479**
13.2.4 Les messages HTTP **480**
13.2.4.1 Le message de demande **480**
13.2.4.2 Le message de réponse **482**

13.3 Le protocole FTP **484**

13.4 Le protocole TFTP **487**

13.5 Le courrier électronique **490**

13.5.1 Le protocole SMTP **492**
13.5.2 Contenu du message **494**
13.5.3 Le protocole POP3 ..... **494**

13.6 Exercices **498**

14 La gestion de réseau SNMP **501**

14.1 Introduction **501**

14.2 La gestion de réseau TCP/IP **502**

14.2.1 Structure de gestion d'information (SMI) **504**
14.2.2 MIB-II **508**
14.2.3 Le protocole SNMP **509**
14.2.3.1 Le message GetRequest **511**
14.2.3.2 Le message GetNextRequest **513**
14.2.3.3 Le message SetRequest **515**
14.2.3.4 Le message Trap **516**
14.2.4 Agent mandataire (Proxy) **516**
14.2.5 SNMPv2 ..... **517**
14.2.5.1 Les messages SNMPv2 ..... **518**

14.3 Exercices **522**

Acronymes **525**

Bibliographie **531**

Index **535**

1 Télécommunications et réseaux

1.1 Introduction

Le domaine des télécommunications couvre l'ensemble des interactions soit interpersonnelles, soit intermachines. Ces deux formes d'interactions se recoupent attendu que le but, finalement, repose sur la transmission d'informations entre deux personnes ou plus. Toutefois, le propos de ce volume est de se limiter strictement aux communications entre ordinateurs et en ciblant, tout particulièrement, les communications qualifiées de numériques.

Les réseaux de télécommunications couvrent un vaste éventail de technologies. Ainsi, parmi les techniques de transmission, l'on peut citer les divers traitements suivants : traitement du signal, de la modulation, du multiplexage, du codage, des protocoles spécifiques au transport, du genre de conduit, etc.

En outre, l'accès au réseau est rattaché à des services d'ordre très précis en raison des protocoles qui le définissent. À titre indicatif, l'accès RNIS (*Réseaux numériques d'intégration de services*) en est un bon exemple. L'accès Internet représente un intérêt tout particulier dans le cadre de cet ouvrage.

L'infrastructure réseautique d'Internet représente le déploiement des éléments réseau. Elle inclut également des modes de transmission qui peuvent différer au cœur du réseau de ceux relatifs aux accès réseau. De plus, les éléments de réseau doivent acheminer les informations en transit en se conformant aux protocoles en vigueur; mais il faut que ces derniers soient judicieusement choisis et soigneusement mis en application afin de tenir compte des possibilités de congestion des réseaux et de l'accès prioritaire.

La topologie du réseau peut remédier, en partie, au problème de congestion. Toutefois, l'ingénierie du trafic permettra au réseau d'éviter cette congestion et de prévoir la croissance à l'expansion du trafic.

La complexité des réseaux repose sur la disposition des services offerts à l'utilisateur compte tenu des protocoles d'accès, du soutien physique et des

méthodes de transmission de même que toute l'infrastructure réseautique. Cette infrastructure permettra aux informations d'être acheminées vers la destination sans altérer pour autant les garanties offertes par le terminal d'origine de la communication.

Chaque terminal de communication est relié au réseau par un port. Il est possible, voire même tout indiqué, de définir un réseau d'accès, notamment un réseau local, en vue d'optimiser l'accès au réseau. Le réseau local se composerait de plusieurs terminaux, d'une topologie propre, de protocoles précis et d'un point d'accès réseau qui leur soit commun. La figure 1.1 illustre l'interconnexion qui prévaut entre les hôtes et le réseau.

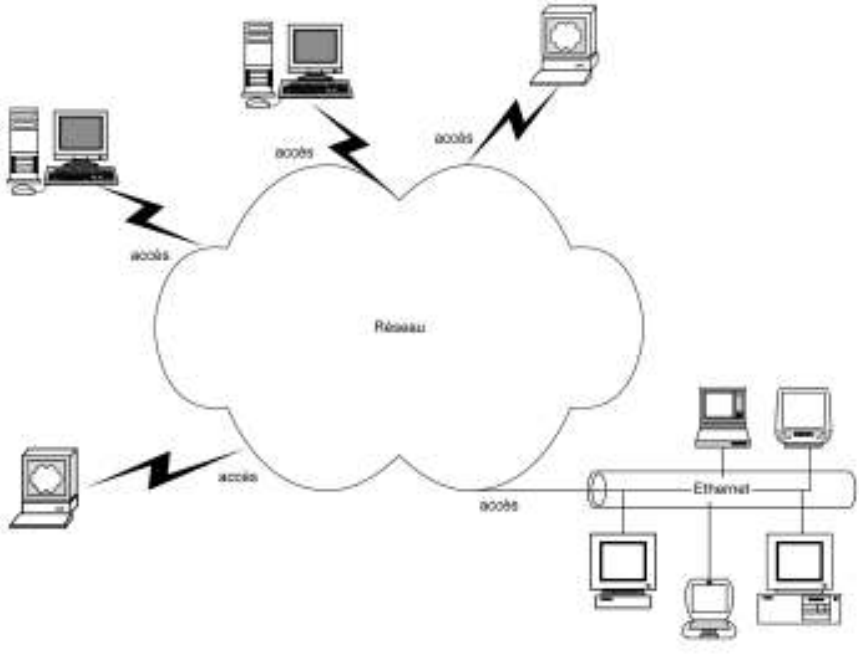


Figure 1.1 Accès réseau

1.2 L'accès réseau

Un réseau de télécommunications peut être représenté par ses accès, le cœur du réseau et le moyen de transport ou le mode de transmission. Les accès qui se trouvent en périphérie du réseau permettent aux terminaux de s'interconnecter. Le cœur du réseau permet de transmettre l'information d'un terminal source vers plusieurs terminaux de destination. Le mode de transport utilisé sert de conduit de déplacement d'un point à l'autre tout en respectant les contraintes imposées (délai, gigue, largeur de bande).

L'accès peut ne comporter qu'un simple terminal hôte ou encore être relié à un réseau local dont les rôles et fonctions sont de desservir de nombreux terminaux. Lorsqu'un utilisateur ou un travailleur autonome veut avoir accès à Internet, on parle alors et en général du branchement à l'accès réseau d'un seul terminal. Il peut s'agir d'un poste de travail doté de l'un des différents systèmes d'exploitation tels : Microsoft, Unix ou Linux, d'un portable ou d'un agenda électronique.

Ces postes de travail sont branchés au réseau par un modem. La liaison au modem peut être faite grâce à un lien sans fil. Ce cas est souhaitable lorsqu'on dispose d'un ordinateur portatif, d'un agenda électronique ou lorsque la liaison par câble pose des problèmes. Les modems dont on se sert sont fonction de la technologie d'accès prévalant dans l'industrie. Au nombre des techniques les plus retenues, citons : les réseaux locaux, le réseau téléphonique, le DSL, le câble et le satellite.

Il est bien plus économique pour la petite ou moyenne entreprise d'être intégrée à un réseau local dans le cas où celle-ci aurait un bon nombre de postes de travail qui doivent soit être interconnectés, soit avoir accès à Internet. L'avantage réside dans le fait que les utilisateurs de ces postes de travail peuvent échanger des données sans devoir pour cela passer par l'accès à Internet. En général, un routeur se charge d'établir la liaison entre le réseau local et le modem qui permet d'avoir accès à Internet. Les réseaux locaux peuvent opter pour la technologie Ethernet ou CSMA/CD, pour l'anneau à jeton ou pour le FDDI.

Les postes de travail peuvent également assurer le rôle de serveurs. Dans ce cas-ci, ils offriraient un éventail de services distincts dont, et pour n'en citer que certains : l'hébergement de sites Web, l'enregistrement de noms de domaines et le serveur de messagerie électronique.

Une grande partie des technologies qui ont cours sur le marché est dotée d'un accès partagé à Internet, ce qui représente en soi un partage des ressources. De nombreux paramètres influent sur le degré du partage de l'accès à Internet : le rendement des réseaux de télécommunications, la puissance des serveurs en place, la surcharge du réseau Internet.

L'accès commuté d'Internet est une technologie de transmission des données qui se fait par le réseau téléphonique commuté. Ainsi, lorsque cet accès est activé, le service téléphonique devient inutilisable. Pour que le service d'accès fonctionne, il faut brancher un modem.

Pour être relié à Internet, chaque utilisateur devra se connecter au réseau d'un fournisseur de services en ayant recours au modem et à la ligne téléphonique. Ce type d'accès Internet est plutôt lent. En effet, la vitesse de communication est influencée par la qualité de la ligne téléphonique. Dans ce cas, le transfert des données est qualifié d'asymétrique. Cela signifie que la vitesse du débit des données diffère selon qu'on soit en mode réception ou en mode transmission. La vitesse de ce lien est, dans les faits, établi à 56 Kbps. Toutefois, les débits en modes transmission et réception peuvent atteindre les vitesses de 33.6 Kbps et 53 Kbps respectivement.

Le Réseau numérique à intégration de services (RNIS) est un réseau numérique qui offre aussi l'accès Internet. Entièrement numérique, cette voie de communication permet la transmission simultanée des signaux voix, données et vidéo. Elle assume une très vaste gamme d'applications possibles. Avec l'avènement de cette technologie, il est possible d'éliminer le recours à des lignes téléphoniques distinctes : une pour la voix, une pour la télécopie et une pour le modem. Désormais, seule une ligne suffit!

Grâce au RNIS, le transfert des données est qualifié de symétrique, ce qui veut dire que les conduits de transmission et de réception ont la même capacité en matière de transfert des données. Selon les besoins des utilisateurs, le service RNIS offre une plus grande gamme de vitesses. Parmi les plus courantes, on retrouve les vitesses de 64 Kbps et de 128 Kbps. Cependant, certains des fournisseurs de services proposent des vitesses plus grandes : 256 Kbps ou 512 Kbps.

Cette technologie requiert un routeur et des cartes Ethernet ou des cartes dédiées RNIS pour ordinateurs. Des adaptateurs (modem numérique) sont indispensables à tout équipement non numérique de l'entreprise (téléphone, fax, …). L'accès Internet peut être commuté ou dédié. Dans le cadre du RNIS, il faut être abonné au service RNIS d'une compagnie de téléphone.

T1, ligne numérique à haut débit, est en soi un lien numérique relié à l'accès Internet. Voie de communication entièrement numérique, elle permet la transmission simultanée des signaux suivants : voix, données et vidéo. Cette ligne numérique à haut débit donne lieu à un vaste éventail d'applications.
À l'instar de RNIS, la technologie T1 élimine le recours à l'utilisation de lignes distinctes que ce soit pour la voix, pour le télécopieur ou pour le modem. Le transfert de données est qualifié de symétrique. La technologie du T1 est un moyen de transport capable de revêtir plusieurs formes toutes dépendantes du multiplexeur utilisé pour l'accès et les besoins en matière de vitesses. Parmi les plus fréquentes, celles que l'on retrouve sont programmées à 64 Kbps et

128 Kbps dans le cas où voix et données passent indistinctement. La vitesse maximale de transmission peut être de l'ordre de 1.5 Mbps. Toutefois, et pour fonctionner correctement, la technologie T1 requiert les éléments suivants : un multiplexeur, des modems numériques, des cartes Ethernet et un routeur (éventuellement). Le T1 est un lien dédié point à point dont les accès simultanés à Internet sont multiples. Chacun d'entre eux peut être soit commuté, soit dédié. Cette technologie concède une certaine souplesse en ce qui a trait aux raccordements. Elle sert surtout dans le cas de l'intégration de la téléphonie numérique et dans celui de la transmission des données. Cependant, on doit être aussi abonné à un fournisseur de services pour avoir accès à l'Internet.

La ligne numérique à paire asymétrique (LNPA) dont l'acronyme anglais est ADSL (*Asymmetric Digital Subscriber Line*) permet à tout abonné d'avoir accès à Internet par une ligne téléphonique normale. De la sorte, l'abonné peut tout à la fois avoir accès à Internet et utiliser simultanément la ligne téléphonique. Le transfert des données est qualifié d'asymétrique. En outre, les vitesses de transmission et de réception sont respectivement de 16 Kbps à 640 Kbps et de 1.5 Mbps à 9 Mbps. Cependant, et pour avoir accès au service, un modem LNPA doit être installé. D'une part, il doit être raccordé à une ligne téléphonique simple et de l'autre branché à une carte Ethernet, à un réseau Ethernet ou encore à un port USB (*Universal Serial Bus*). Les abonnés doivent se trouver à une distance maximale type de tout accès au LE (*Local Exchange*/Central téléphonique local).

La ligne d'abonné numérique à très haut débit, dont l'acronyme anglais est VDSL (*Very-high-data-rate Digital Subscriber Line*), est une technologie qui permet à tout abonné d'avoir accès à Internet, sur courte distance, par une ligne téléphonique normale. Néanmoins, le débit est nettement plus élevé. L'abonné peut simultanément utiliser la ligne téléphonique. Dans ce cas, le transfert des données est qualifié d'asymétrique. Par ailleurs, les vitesses de transmission et de réception sont respectivement de 1.544 Mbps à 2.3 Mbps et de 13 Mbps à 51 Mbps.

Le modem câble est une technologie de transmission qui permet d'avoir accès à Internet par le réseau de télédistribution. Dans ce cas-ci, le service téléphonique de l'abonné n'est ni utilisé ni perturbé. Avec cette technologie, le transfert des données est qualifié d'asymétrique. Par ailleurs, les vitesses de transmission et de réception sont respectivement de 1 Mbps à 10 Mbps et de 10 Mbps à 30 Mbps. Par contre, l'accès au service doit être supporté par un modem câble. D'une part, il doit être raccordé à un câble coaxial et, de l'autre, branché à une carte Ethernet ou à un réseau Ethernet.

L'accès Internet par fibre optique repose sur une technologie distincte de la transmission des données, car elle permet d'avoir accès à Internet grâce à un réseau de fibres optiques. Dans ce cas, le transfert des données est qualifié de symétrique. La vitesse de communication varie de 1.5 Mbps à 1 Gbps. Pour accéder au service, il est alors important de prévoir la pose d'un commutateur Ethernet qui permet de relier le réseau Ethernet du client à celui du fournisseur de services.

Le système de distribution multipoint (SDM), connu sous l'acronyme anglais MDS (*Multipoint distribution system*), repose sur une technologie qui permet d'avoir accès à Internet grâce aux liens micro-ondes point à point. Cette technologie fonctionne conformément au spectre se situant entre 2.6 et 2.7 GHz. Dans ce cas, le transfert des données est qualifié d'asymétrique. Le mode de transmission de ce système (SDM) est bidirectionnel. Par ailleurs, les vitesses de transmission et de réception sont respectivement de 200 Kbps et de 705 Kbps à 1.5 Mbps. Dans le cadre de la transmission et dans celui de la réception, on a recours à une antenne de même qu'à un modem câble, tous deux reliés à une carte Ethernet ou bien à un réseau Ethernet et il est possible d'obtenir environ cinq adresses IP.

Le système mondial de communications mobiles/Service général de radio-communication en mode paquet, connu sous le double acronyme anglais de GSM/GPRS (*Global System for Mobile Communications/General Packet Radio Service*) repose sur la technologie de transmission qui permet d'avoir accès à Internet par le biais du réseau de la téléphonie cellulaire du type GSM. Dans cet exemple, le transfert des données est qualifié d'asymétrique. Théoriquement, la vitesse de communication devrait pouvoir atteindre les 115 Kbps; toutefois, elle n'est à l'heure actuelle que de 56 Kbps. Un téléphone numérique mobile GPRS doit être branché pour avoir accès à Internet. Si on désire y accéder à partir d'un ordinateur, il faut placer une carte pour PC spécialement conçue pour ce genre d'accès.

La bande industrielle, scientifique et médicale connue sous l'acronyme anglais de ISM Band (*Industrial, Scientific and Medical Band*) repose sur une technologie sans fil et permet d'avoir accès à Internet grâce aux liens micro-ondes. Cette technologie fonctionne conformément à un spectre de 2.4 GHz et elle est exempte de tout permis radio. Dans le cadre de cette technologie, le transfert des données est qualifié de symétrique. Le mode de transmission de ce système (*ISM Band*) est bidirectionnel. Par ailleurs, sa vitesse de communication varie de 1.5 Mbps à 11 Mbps. Tant pour la transmission que pour la réception, il faut avoir recours à une antenne de même qu'à une carte pour PC spécialement conçue pour ce genre d'accès.

L'accès Internet par satellite et de type unidirectionnel se base sur une technologie d'accès à Internet uniquement en mode de réception des données. Dans ce cas-ci, on a recours aux satellites de télécommunications. Les requêtes adressées au réseau se font par un modem téléphonique traditionnel. Pendant toute la durée de l'accès à Internet, l'accès téléphonique lui étant dédié n'est pas disponible. Le transfert des données y est qualifié d'asymétrique. La vitesse de transmission correspond à celle du modem d'accès commuté à Internet et la vitesse de réception peut atteindre les 400 Kbps. En ce qui a trait à la réception, une antenne parabolique doit être installée chez l'utilisateur. Cette antenne permet ainsi de capter les signaux émis par le satellite et de les acheminer jusqu'aux ordinateurs à l'aide d'une carte de réception grâce à un câble coaxial.

L'accès Internet par satellite et de type bidirectionnel repose sur une technologie d'accès à Internet tant pour les modes de transmission que de réception. Dans le cas présent, tout comme dans le cas précédent, on a recours aux satellites de télécommunications. Pendant toute la durée de l'accès, le service téléphonique n'est ni utilisé ni perturbé. Dans cet exemple d'application de technologie d'accès Internet bidirectionnel, le transfert des données est qualifié d'asymétrique. Par ailleurs, en ce qui a trait aux vitesses de transmission et de réception, soulignons que dans le premier cas, soit en mode transmission, elle est limitée à 128 Kbps alors que dans le second cas, en mode réception, elle peut atteindre les 400 Kbps. De plus, il est impérieux de disposer d'une antenne parabolique pour procéder à la transmission et à la réception. Cette antenne parabolique doit être installée chez l'utilisateur. Il est bon de souligner que ladite antenne est reliée à deux modems externes au moyen d'un câble coaxial. Ces deux modems servent à la transmission et à la réception.

La station terrienne à très petite ouverture d'antenne est connue sous l'acronyme anglais VSAT (*Very Small Aperture Terminal*). Cette technologie d'accès à Internet requiert l'utilisation de satellites de télécommunications par une station de réception terrienne. Le transfert des données qualifié d'asymétrique ne s'y fait que point à point. La liaison spécialisée (unidirectionnelle ou bidirectionnelle) va à une vitesse qui varie de 64 Kbps à 45 Mbps.

1.3 Types de réseaux

Les réseaux de télécommunications qui font appel à la communication numérique se divisent en trois catégories principales, compte tenu de leurs étendue et fonction, à savoir :

- Les réseaux étendus (*Wide Area Network* ou WAN)
- Les réseaux métropolitains (*Metropolitain Area Network* ou MAN)

- Les réseaux locaux d'entreprises (*Local Area Network* ou LAN)

Les réseaux étendus (WAN) sont, en règle générale, publics. Cette catégorie de réseaux permet d'établir des liaisons point à point entre des commutateurs et routeurs ou éléments de réseaux. Les liens à très hauts débits et ces éléments de réseaux composent la topologie du WAN. Les clients sont facturés selon le temps d'utilisation qu'ils font du réseau. Les réseaux étendus sont, en règle générale, bien protégés et jouissent d'une bonne cote en matière de sécurité.

Du fait où les distances importent, le délai de transmission est un facteur à prendre en compte. Un autre délai ne doit pas être négligé : il s'agit du temps de transit occasionné par le relais qui survient à chacun des nœuds du réseau. Si dans le réseau le taux d'erreurs est élevé, la transmission peut également accuser du retard. Il existe plusieurs catégories de réseaux étendus dont le débit était, en règle générale, faible. Toutefois, de nos jours, nous sommes témoins d'un essor fulgurant des réseaux WAN à hauts débits. Celui des réseaux WAN sur lequel nous allons nous pencher est le réseau Internet.

Les réseaux métropolitains sont également connus sous l'appellation de réseaux urbains. Ils forment généralement la dorsale (*backbone*) permettant ainsi à plusieurs réseaux locaux de s'interconnecter. On retrouve les réseaux métropolitains dans les organisations ou campus là où la dispersion, des édifices ou des étages ne peut permettre la mise en place d'un seul réseau local. Dans un tel cas, le réseau métropolitain assume un rôle clé, car il est celui qui va établir le lien entre les réseaux locaux de chacun des départements et constituer un vaste réseau sans pour autant revêtir l'envergure d'un réseau WAN.

En règle générale, on peut dire que les réseaux métropolitains ont de hauts débits. Or, pour ce qui est du principe de facturation, il convient de signaler qu'elle est plutôt basée sur le partage du coût de fonctionnement et non pas sur le temps d'utilisation. Le réseau métropolitain sur lequel nous nous pencherons dans ce volume est le FDDI (*Fiber Distributed Data Interface*/Interface optoélectronique de données à mode réparti).
Le réseau local d'entreprise est un réseau dont la portée est limitée (quelques centaines de mètres). Le réseau local en est un d'accès par excellence. Il dessert le département d'une entreprise et peut répondre à la demande d'une centaine d'utilisateurs. L'administration (son exploitation) est centralisée et les coûts afférents se limitent seulement à ceux de l'acquisition du matériel et à ceux de la gestion du réseau.

Il existe plusieurs types de topologies et de fonctionnement inhérents aux réseaux locaux. Conformément à leurs topologies, et d'après celles-ci, il nous est

donné de pouvoir procéder au classement des réseaux locaux. Au nombre des plus importantes des topologies, mentionnons les suivantes : le réseau à segment, le réseau à arborescence, la boucle simple, l'étoile et la boucle double. Toutes ces topologies feront l'objet d'une étude dans les chapitres 4 et 6 du présent ouvrage lesquelles traitent de protocoles à arborescences et à anneaux à jeton propres aux réseaux locaux.

1.3.1 Réseaux commutés

Dans le cas des réseaux de commutation de données, nous pouvons distinguer différents types de commutation. Les réseaux font appel soit à la commutation de circuits, soit à la commutation par paquets soit, et parfois seulement, à la commutation de messages.

Dans tous les cas, les commutateurs permettent de retracer les chemins empruntés lors de chacun des appels faits. En ce qui a trait aux commutateurs de circuits, un circuit sera réservé sur le chemin choisi pour placer un appel tant et aussi longtemps que celui-ci est en cours et donc activé. Par contre, pour ce qui est des commutateurs par paquets, la largeur de bande peut être utilisée par tous ceux qui désirent transmettre avec une réserve toutefois : le droit d'utilisation des lignes, empruntant le chemin pour se rendre jusqu'au point de destination, est réservé.

1.3.1.1 La commutation de circuits

Dans un réseau de données à commutation de circuits, la communication emprunte un circuit dédié appliqué à deux ou plusieurs terminaux. Ce circuit n'est dédié que pendant le temps de durée de ladite communication. Par conséquent, il est impératif d'établir au préalable un appel ou une connexion en vue de retenir un circuit. Une fois celui-ci alloué, il est possible de procéder à l'échange des données. Lorsque ces échanges d'information prennent fin, un message de fin d'appel est envoyé. La cessation de l'appel engendre la libération du circuit (après quoi, d'autres utilisateurs peuvent le réserver). La singularité de la commutation de circuits veut que, lors de la communication, les terminaux soient les seuls à se servir du circuit, aucuns des autres terminaux ne pouvant y avoir accès. L'avantage à cela repose sur ce que, pour les terminaux, la communication se fait très rapidement. La figure 1.2 illustre le principe et le phénomène de la transmission par un circuit dédié depuis la source jusqu'au point de destination. Cependant, le désavantage repose sur le fait que tout au long de la connexion, le circuit continue d'être réservé, et ce, même pendant les pauses de silence, ce qui l'empêche de pouvoir être alloué à tout autre utilisateur.

Dans la commutation de circuits, le circuit peut être physique ou temporel. Le cas du câble téléphonique d'une résidence est un bel exemple de circuit physique. En effet, seule la communication en cours utilise de manière exclusive ce lien. En ce qui a trait au lien temporel, nous nous trouvons également devant un lien physique. Toutefois, la différence entre ces deux derniers cas est liée au fait que le lien temporel est dédié périodiquement aux mêmes utilisateurs (abonnés), et ce, pour un laps de temps prédéfini. Ce laps de temps est connu sous le nom d'intervalle de temps (*time slot*). Selon la largeur de bande du lien, ce dernier peut avoir plusieurs intervalles de temps. Le multiplexage temporel synchrone est la technique utilisée. Traditionnellement, la commutation de circuits est appliquée à la voix.

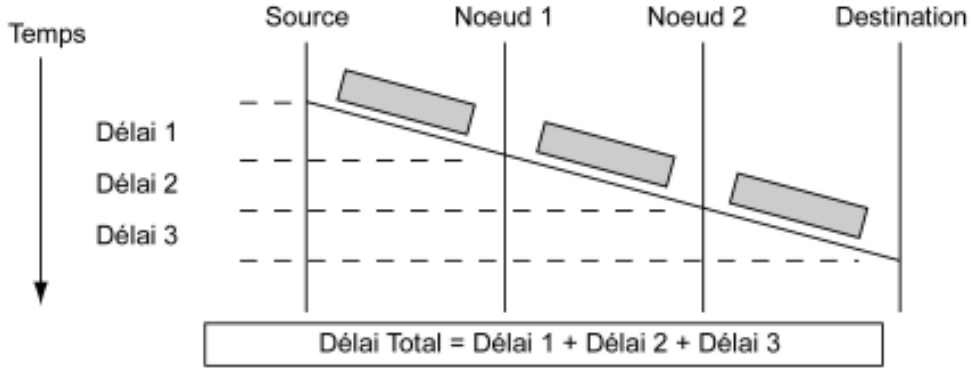


Figure 1.2 Commutation de circuits

1.3.1.2 La commutation par paquets

Le profil de la communication des données est plutôt sporadique. La transmission des données se fait en rafales et est suivie de silences. Les réseaux de commutation par paquets ne prévoient pas d'intervalles de temps périodiques. Les réseaux font plutôt appel au multiplexage temporel asynchrone. Ces intervalles de temps sont attribués de manière statistique, ce qui signifie que lorsqu'un terminal doit transmettre des données, le réseau les accepte et les envoie. Une fois la transmission terminée, le lien est assigné à un autre utilisateur pour autant que ce dernier ait des données à transmettre. Il est bon de souligner ici que la disponibilité temporelle du circuit est fonction du besoin de la transmission des données par les utilisateurs. Chacune de ces transmissions est limitée soit en regard du temps accordé, soit en regard de la quantité limite temporelle des données à transmettre.

En règle générale, lorsque la permission de transmettre lui est donnée, l'unité de transmission des données a un nombre maximal d'octets tel que défini par le réseau. Cette grandeur varie selon certains réseaux et, selon d'autres, elle demeure fixe. Ces unités de données portent différents noms d'après le genre de réseau ou encore du protocole qui s'y applique. Ainsi, on parlera de paquets, de datagrammes, de segments ou de cellules pour n'en nommer que quelques-uns.

À tout moment, le lien est partagé entre tous les transmetteurs du réseau et, de ce fait, son exploitation en devient optimale. Toutefois, chacune des unités de données doit inclure un en-tête qui vient se greffer aux données qui représentent la charge utile. Ces en-têtes donnent lieu à un surcroît du trafic dans le réseau et doivent être traités par chaque routeur. Cela représente une surcharge du réseau. La figure 1.3 illustre les délais supplémentaires de traitement par chacun des nœuds.

Par ailleurs, les réseaux limitent la longueur maximale des unités de données. Lorsqu'un terminal doit procéder à la transmission d'un fichier, il devra au préalable le fragmenter en plusieurs paquets. Chacun de ceux-ci devra comporter son propre en-tête, ce qui en soi représente une surcharge du trafic des données.

La grandeur des unités de données a des retombées sur les délais ou les retards occasionnés lors du transit des données et, par conséquent, sur le trafic au sein du réseau. Tandis que les unités des données sont petites, un plus grand nombre d'utilisateurs peuvent se partager le lien.

Par contre, l'accumulation d'en-têtes constitue une surcharge non négligeable du travail. Aussi, plus les unités de données sont grandes, plus elles contribuent à réduire cette surcharge. Cependant, en raison de leur longueur, le partage des ressources s'en trouve ralenti.

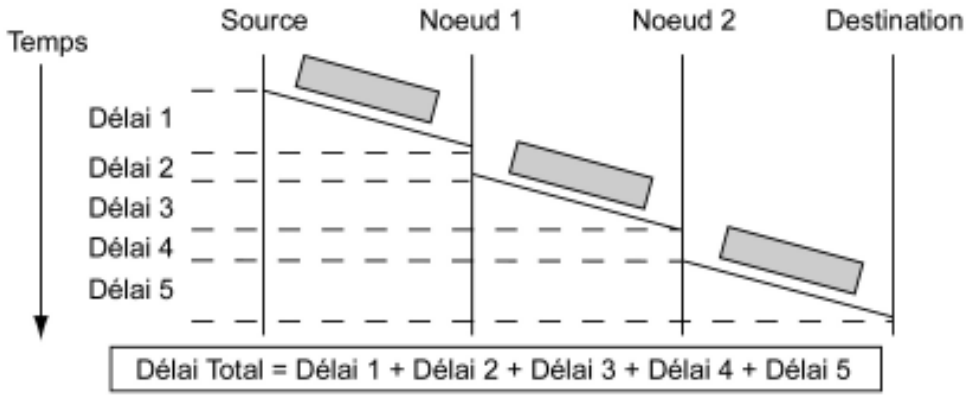


Figure 1.3 Commutation par paquets

La commutation par paquets donne des taux de transmission distincts à chacun des utilisateurs y ayant accès. Sur une période de temps donnée, la fluctuation de ces taux se traduit par la transmission d'un nombre de données plus élevé de la part d'un seul et même utilisateur et, de ce fait, le débit d'unités de données par seconde s'en trouve plus élevé.

1.3.1.3 La commutation de messages

La commutation de messages a recours aux concepts utilisés dans la commutation par paquets. Cependant, les serveurs de messages emmagasinent momentanément les données plutôt que de relayer les unités de données vers leur destination sans plus attendre. Tout comme la figure 1.4 l'illustre, ces messages sont tout d'abord traités par les serveurs de messages puis acheminés vers leur destination. Un exemple type de ce genre de réseau nous est donné dans le cas de la messagerie électronique ou du courriel. Alors que les courriers électroniques sont des messages, ils ne vont jamais directement de la source à la destination (de l'expéditeur au destinataire). En effet, les messages des courriers électroniques transitent tout d'abord par le serveur de messagerie.

Ce genre de réseau peut se montrer quelque peu plus lent. Toutefois, il s'avère être des plus efficaces en regard des services pour lesquels il a été pensé.

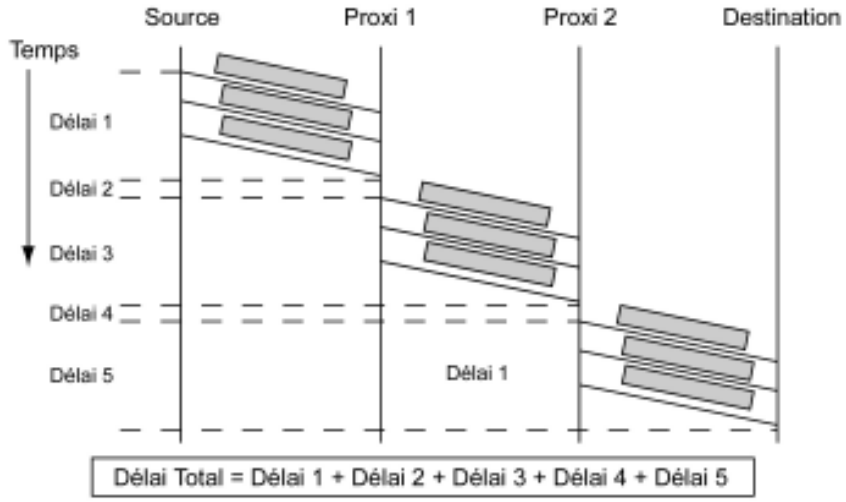


Figure 1.4 Commutation de messages

Le réseau de commutation de messages peut fort bien être appliqué au sein du réseau de commutation par paquets. En effet, l'échange de messages entre la source les serveurs et la destination, peut avoir lieu dans un réseau de commutation par paquets. Les routeurs du réseau de commutation par paquets vont acheminer les unités de données au fur et à mesure qu'ils les reçoivent. Par contre, les serveurs de messages les retiendront jusqu'à ce que leur parvienne une requête pour les transmettre.

1.3.2 Acheminement dans les réseaux

Un réseau de communication se compose de routeurs et de liens. Chacun de ces routeurs doit pouvoir établir le meilleur chemin à emprunter pour l'acheminement des unités de données reçues. Il établit ses choix à partir des informations contenues dans l'en-tête de chacune des unités de données de même qu'en se fondant sur les tables de routage mises constamment à jour.
Un protocole de routage est alors utilisé par les routeurs qui prennent les décisions nécessaires à l'acheminement des unités de données. Ils basent leurs décisions sur l'information retrouvée dans l'en-tête des unités de données et sur les tables de routage qu'ils tiennent à jour. Il existe de nombreux algorithmes de routage. Les critères peuvent se baser sur le nombre de sauts requis pour parvenir au point de destination, sur le délai qui sépare la source du point de destination ou encore sur la largeur de bande qui est disponible lors de l'acheminement des blocs de données. Les protocoles de routage se conforment à l'un ou l'autre de ces algorithmes.

Chaque routeur calcule et fixe le coût total des liens auxquels il a accès et le transmet périodiquement aux routeurs contigus. Ainsi, les tables de routage sont établies et rajustées dans chacun des routeurs sur réception des coûts des autres routeurs. Il est alors donné à chacun de ces routeurs d'établir les valeurs qu'il conservera et celles qu'il modifiera pour que ses informations demeurent toujours aussi fiables. Ces coûts sont utilisés dans la détermination du chemin le plus court.

1.4 Internet

Une des caractéristiques requises pour le développement du réseau Internet était que ce dernier devait continuer à fonctionner même si une partie de son déploiement venait à tomber en panne.

Les années 1960 avaient été celles de la guerre froide et la hantise d'une attaque nucléaire, doublée par le risque de démantèlement des moyens de communication, fut fort probablement à l'origine de l'un des critères techniques du réseau. C'est cette crainte qui a animé le département de la défense américaine et qui lui a donné la volonté nécessaire pour concevoir le réseau, et ce, malgré l'avis contraire d'un bon nombre d'experts pour qui une telle entreprise était vouée à l'échec.

L'équipe composée pour mener à bien un tel projet n'avait, en somme, pas d'expérience en la matière. D'ailleurs, il n'y avait aucun expert capable de réaliser ce projet. Par contre, les membres retenus pour former l'équipe étaient des scientifiques acharnés. L'approche prônée fut basée sur un concept nouveau : la commutation par paquets.

Ainsi naquit un réseau nommé ARPAnet. Les premiers commutateurs par paquets étaient des IMP (*Interface Message Processor*/Processeur de traitement de messages d'interface) conçus par la compagnie *Bolt Beranek and Newman* (BBN) à l'Université de Californie à Berkeley (UCB). Ces IMP étaient utilisés pour que les paquets soient relayés selon un protocole précis qui fut, dans les faits, la première version du protocole IP. ARPAnet est donc l'ancêtre du réseau Internet tel que nous le connaissons aujourd'hui.

Le réseau Internet est un réseau universel et il permet l'interconnexion entre utilisateurs de manière planétaire. L'accroissement des accès va au-delà de toute attente. De nos jours, quiconque désire établir un contact ou obtenir des informations n'a qu'à se relier à l'Internet. Quiconque souhaite faire connaître ses idées, fournir des informations, n'a qu'à se relier à l'Internet pour permettre l'accès à un site.

Quant aux services Internet, le concept de client-serveur est profondément mis en pratique. Le client, qui se trouve à une extrémité du réseau, peut établir une communication avec le serveur qui lui se trouve à l'autre extrémité du réseau. Le client adresse une demande au serveur qui répond conformément à la demande émise. Ce dernier transmet alors l'information requise ou envoie un avis de rejet relativement à la demande formulée par l'utilisateur.

À titre indicatif, soulignons que le fureteur n'est rien d'autre qu'un programme client-serveur qui permet à tout utilisateur de formuler une demande et de l'adresser. Cette dernière peut prendre la forme d'une demande d'accès au site Web propre au serveur. Le serveur transmet alors la page demandée. Tout un ensemble de demandes et de réponses peut intervenir entre utilisateurs et serveur créant une interaction et permettant d'avoir accès aux informations figurant sur le site d'un ou de plusieurs serveurs.

Comme autre exemple d'échange client-serveur, citons le cas d'une session FTP (*File Transfer Protocol*/Protocole de transfert de fichiers). Au cours de cette session, le client émet une demande de transfert d'un ou de plusieurs fichiers particuliers que le serveur lui transmet. Le courrier électronique fonctionne aussi conformément au même principe.

Le succès d'Internet est dû en grande partie à l'universalité de son protocole réseau : le protocole IP (*Internet Protocol*). Tout équipement relié à Internet doit avoir recours au protocole IP de sa couche réseau. Ce protocole est commun d'un bout à l'autre du réseau, soit entre la source et la destination. De plus, chacun des accès Internet a une adresse IP qui identifie exclusivement ledit accès. C'est donc par cette adresse IP que les routeurs peuvent acheminer les appels reçus et transmettre les paquets une fois ceux-ci parvenus à destination.

L'expansion des réseaux locaux Ethernet et l'adoption des protocoles TCP/IP, surtout par les utilisateurs de UNIX, ont contribué en très grande partie à l'évolution d'Internet. Les réseaux locaux étaient élaborés et régis par UNIX. Les stations ayant un accès au réseau eurent tout avantage à s'y conformer. Elles adoptèrent les protocoles TCP/IP afin de pouvoir communiquer au sein d'un réseau local. Les réseaux locaux avaient les tous premiers définis l'accès réseau, et ce, bien avant qu'une quelconque tendance ne prenne forme en réseautique. Avec l'avènement d'Ethernet qui s'est répandu partout sur la planète, les réseaux étendus ne purent que s'adapter, à tout le moins, dans la couche réseau, couche qui se devait d'être homogène pour qu'il y ait compatibilité de bout en bout.

1.5 Concepts de la transmission

Afin de pouvoir mieux saisir les fonctions physiques des réseaux décrits dans cet ouvrage, nous passerons en revue les éléments de la transmission dans cette section.

De prime abord, soulignons que les signaux générés par toute communication sont transmis par des méthodes distinctes. Ces signaux sont diffusés par un conduit à propagation guidée ou par un conduit à propagation rayonnée.

Les conduits à propagation guidée acheminent les signaux du terminal d'origine au terminal de destination. Le mode de transmission au travers de ces conduits dépend d'un bon nombre de facteurs dont, entre autres : le type de communication, numérique ou analogique, le mode de modulation, la vitesse de transmission, etc. Les types de conduits à propagation guidée, dont la comparaison est faite entre eux au tableau 1.1, sont les suivants :

- Paires symétriques ou paires torsadées;
- Câbles coaxiaux;
- Fibres optiques.

Tableau 1.1 Comparaison entre conduits guidés

Type de conduit	Paires	Câbles coaxiaux symétriques	Fibres optiques
Capacité	*faible*	*moyenne*	*élevée*
Bande passante	*300 - 3 kHz*	*900 MHz*	*1 GHz*
Sensibilité au brouillage	*élevée*	*moyenne*	*nulle*
Affaiblissement	*fort*	*moyen*	*faible*

1.5.1 Paires symétriques ou torsadées

Ces paires se composent de deux fils de cuivre. Elles sont utilisées isolément ou regroupées dans un câble. Chaque paire de fils est torsadée en vue de réduire tant le rayonnement électromagnétique que les brouillages.
Les paires symétriques ou paires torsadées (figure 1.5) sont parmi les plus répandues. Elles sont utilisées au sein du réseau des abonnés afin de relier les téléphones au centre de commutation local. Chacune de ces paires de fils sert à assurer le bon cheminement d'une conversation donnée.

De nos jours, l'évolution que ce genre de câble a connue va s'accentuant. Des catégories ont été établies d'après l'entrelacement des paires de fils sur une longueur d'un mètre. Plus nombreux sont les entrelacements et moins l'effet du rayonnement électromagnétique se fait ressentir sur le signal, ce qui entraîne un brouillage moindre.

Les catégories de câbles ont été définies conformément à la norme EIA 568 A. Celle-ci énonce les caractéristiques propres aux trois types de câbles UTP (*Unshielded Twisted Pairs*/Paires torsadées non protégées); il s'agit des catégories 3, 4 et 5.

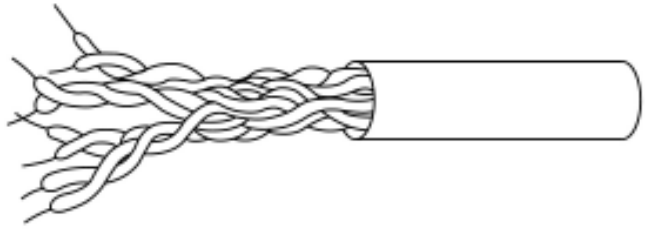


Figure 1.5 Paires torsadées

Les catégories 3 et 5 sont celles qui sont parmi les plus utilisées dans les réseaux locaux. En ce qui a trait à celle de la catégorie 5, soulignons que cette dernière a une spécificité toute particulière, car de toutes c'est la plus fiable.

Les réseaux locaux de 100 Mbps (Mégabits par seconde) préconisent ce genre de câble. Certains réseaux de 1 Gbps (Gigabit par seconde) l'utilisent sur de courtes distances. Soulignons qu'il existe aussi la catégorie 6 dont la norme a été définie et dont la mise en application sera réservée aux réseaux favorisant l'usage des gigabits par seconde.

1.5.2 Les câbles coaxiaux

Chaque câble coaxial (figure 1.6) est composé d'un conducteur central ceint d'un isolant et d'un conducteur tubulaire externe. Tout câble coaxial assume le même rôle que les paires symétriques, soit celui d'acheminer les signaux électriques.

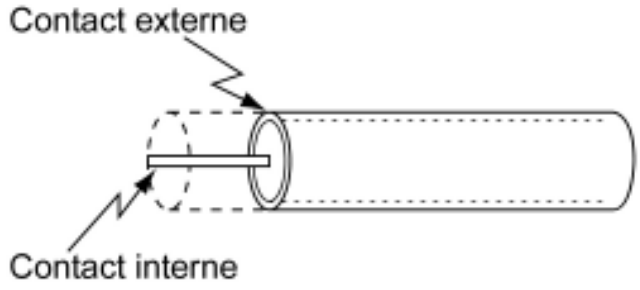


Figure 1.6 Câble coaxial

Ces câbles trouvent, entre autres situations, une application au sein des réseaux de câblodistribution de même qu'entre les centres de commutation du réseau téléphonique.

Grâce à l'isolant séparant le conducteur interne du conducteur externe, le câble coaxial est, jusque dans une certaine mesure, isolé du bruit électromagnétique. En raison de son immunisation contre le bruit, le câble coaxial peut, par exemple, transmettre à haute vitesse et assurer l'acheminement de plusieurs conversations téléphoniques. Ce câble est toujours utilisé dans les réseaux de câblodistribution, et ce, en dépit de son remplacement systématique dans plusieurs liens, sur de longues distances, par la fibre optique.

1.5.3 Les fibres optiques

On distingue divers types de fibres optiques. Peu importe la catégorie, la fibre optique permet d'assurer l'acheminement d'un signal sous la forme d'impulsions lumineuses. Le bruit électromagnétique n'a d'effet que sur le signal électrique. C'est donc dire que la transmission par fibre optique est totalement immunisée contre toute sorte d'interférence électromagnétique.

La fibre optique présente de multiples avantages si on la compare au câble métallique. En effet, elle n'est pas encombrante et elle est tout à la fois extrêmement mince, très fiable et nettement plus économique. La capacité peut atteindre plusieurs gigabits par seconde. Par ailleurs, la fibre optique n'est pas sujette aux interférences électromagnétiques, aux radiations voire à la température ou à l'humidité. Avec son utilisation, la diaphonie n'est pas à craindre attendu que les fibres ne sont pas couplées.

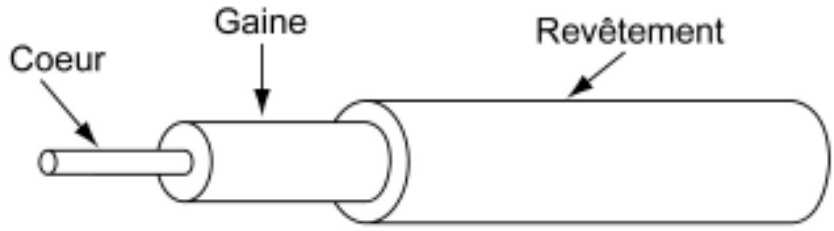


Figure 1.7 Fibre optique

La fibre optique (figure 1.7) comporte les deux éléments suivants :

- Le cœur qui se trouve à la partie centrale du câble et fait à partir d'un verre de très grande qualité.
- La gaine entourant le cœur.

Soulignons que le signal lumineux passe par le cœur (figure 1.8). Pour sa part, la gaine, faite aussi d'un verre dont l'index de réfraction n_g inférieur à celui du cœur n_c, empêche la lumière de s'échapper.

Le revêtement sert simplement à protéger ces deux éléments de la fibre optique.

La fibre optique est souvent désignée par deux nombres : le diamètre du cœur et celui de la gaine. Ces diamètres sont exprimés en micromètre, mais leur rapport n'est pas dimensionné. Par exemple, 50/125 signifie que le cœur de la fibre a un diamètre de 50µm et la gaine de 125µm.

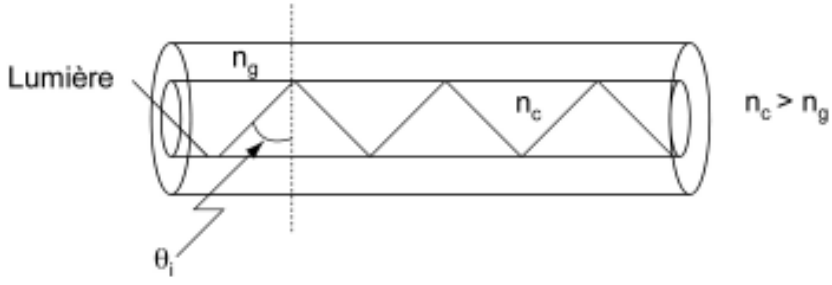


Figure 1.8 Réflexion de la lumière dans la fibre

θ_C - angle critique de la fibre est celui qui, une fois atteint par θ_i, ne réfléchit plus la lumière.

La lumière est réfléchie dans le cœur. Or, pour y rester, l'angle d'incidence θ_i de la lumière doit être plus petit que l'angle critique θ_C du matériau.

Il existe principalement deux types de fibres optiques : la fibre monomode et la fibre multimode.

Le cœur d'une fibre monomode est bien plus petit que celui d'une fibre multimode. La fibre monomode ne peut contenir qu'une seule longueur d'onde et le rapport des diamètres cœur/gaine est de 10/125. Elle n'est utilisée que sur de longues distances.

Par contraste avec la fibre monomode, la fibre multimode peut contenir plus d'une longueur d'onde et est sujette à une atténuation du signal bien plus rapidement que dans le cas de la fibre monomode.

1.5.4 Propagation rayonnée

La propagation rayonnée est définie comme étant la transmission d'ondes dans l'air. Étant donné qu'il n'y a pas de conduit, les ondes se propagent. Nous retrouvons ce mode de transmission dans les transmetteurs à faisceaux hertziens, radio, télévision, téléphonie cellulaire et satellite.

Étant donné que les transmissions d'ondes font appel au même environnement et conduit, il importe que chacun des services ait recours à une bande de fréquences qui soit distincte pour éviter de donner lieu à des interférences. Quelques attributions de fréquences sont citées à titre indicatif.

Exemples d'attributions de fréquences

Radiodiffusion AM	0,56 MHz à 1,6 MHz
Radiodiffusion FM	88 MHz à 108 MHz
Télédiffusion	54 MHz à 88 MHz et 174 MHz à 806 MHz
Services publics	Bande des 30 MHz et des 410 MHz
Téléphone cellulaire	Bande des 850 MHz
Réseau hertzien national	Bandes des 4-6 GHz et des 11-14 GHz

La transmission des ondes propagées dans les airs ne se fait pas directement de la source à la destination. En effet, celle-ci est soumise à de nombreux obstacles comme les immeubles et les véhicules en mouvement. Ces obstacles peuvent causer la réflexion des ondes.

Les effets occasionnés par le bruit sur les ondes vont varier selon la longueur d'onde du signal émis. Les bruits électromagnétiques ont un effet perturbateur bien plus important sur les signaux émis à basse fréquence. De plus, les intempéries telles que la neige, la pluie et le brouillard ont un effet néfaste sur les signaux émis à haute fréquence et, par conséquent, sur les signaux émis sur ondes courtes.

1.5.4.1 Les faisceaux hertziens

Les ondes propagées à l'aide des systèmes de transmission par faisceaux hertziens ont leurs fréquences qui oscillent entre 2 et 406 GHz. Dans ce cas, on parle de micro-ondes. La transmission de ces ondes, tout comme leur réception, se fait à partir d'antennes paraboliques montées sur des tours.

En général, ces systèmes servent de relais des signaux dans des régions où l'installation de câble est coûteuse ou difficile à implanter. Par contre, il importe que les tours adjacentes soient visibles afin de pouvoir capter les signaux. En terrain plat, les tours peuvent être espacées de 60 km. Enfin, signalons que parce qu'elles sont courtes, ces ondes sont sensibles aux intempéries.

1.5.4.2 Les liaisons par satellite

Les satellites sont des relais de signaux. Ceux qui sont géostationnaires conservent la même position par rapport à la Terre et, de ce fait, ils doivent tourner à la même vitesse pour donner cette apparence d'immobilité.

À partir de stations terriennes fixes, le satellite géostationnaire est de service en tout temps. Il faut au minimum trois satellites géostationnaires judicieusement placés pour recouvrir la surface de la Terre.

Les ondes relayées par satellite sont des ondes courtes. Leurs fréquences ascendantes et descendantes diffèrent pour éviter toute interférence.

Les signaux satellites sont transmis et reçus grâce à des antennes paraboliques. Certaines d'entre elles sont réceptrices uniquement à l'instar des récepteurs de signaux de télévision. Par contre, les antennes utilisées par les fournisseurs de services sont tout à la fois transmettrices et réceptrices.

1.5.5 La synchronisation

Afin de décoder et d'interpréter correctement la structure des bits, le récepteur et le transmetteur doivent être synchronisés, peu importe le moyen de

communication. Avant toute chose, il importe de savoir que plusieurs éléments sont visés par cette synchronisation. Le récepteur doit alors établir :

- le début (commencement) de chaque bit afin de pouvoir connaître quel en est l'état (0 ou 1), en procédant à la lecture au milieu de la cellule de bit;
- le début et la fin de chacun des octets;
- le début et la fin de chaque bloc de message au complet (trame).

Ceci représente respectivement trois genres de synchronisation :

1. de bits ou d'horloge;
2. d'octets;
3. de blocs ou de trames.

Le choix de la méthode de synchronisation dépendra de plusieurs facteurs : le conduit, la vitesse de transmission, le codage, le protocole de la couche physique et les méthodes de transmissions synchrones ou asynchrones.

1.5.5.1 Synchronisation de bits

La synchronisation de bits est l'œuvre de celle des deux horloges respectives : du récepteur et du transmetteur. Le transmetteur achemine ses bits en tenant compte des impulsions émises par l'horloge source. En ce qui concerne le récepteur, soulignons que son horloge lui permet de déterminer avec précision le moment de la lecture de chacun des bits reçus afin d'en préciser l'état. De la sorte, les deux horloges (émetteur et récepteur) doivent fonctionner à l'unisson. Par ailleurs, la méthode dite de codage des bits constitue un moyen sûr de soumettre les bits au récepteur à une cadence régulière pour qu'il puisse faire concorder son horloge avec celle du transmetteur.

Il importe que le récepteur soit d'une très grande précision lorsqu'il fera la lecture des impulsions reçues et, comme il ne peut y avoir qu'une lecture par impulsion, le facteur précision joue un rôle primordial. En effet, un décalage de l'horloge fausserait la lecture des bits. Une oscillation de ce décalage est la gigue. Le moyen de procéder à la correction de la gigue est d'extraire un bit de la trame, ce qui ajusterait l'horloge.

1.5.5.2 Synchronisation des octets

Même dans le cas où les bits sont bien synchronisés, il importe de connaître le début de chacun des octets afin de constituer l'élément de base de la communication.

Il est vrai qu'avant toute chose, il doit y avoir synchronisation des bits. Ce n'est que par la suite que celle des octets survient.

1.5.5.3 Synchronisation des trames

Tout comme nous le verrons de façon plus détaillée au chapitre 2 de ce volume, les trames forment l'unité des données de protocole. De par leur nature même, ces unités de données permettent aux couches de communication d'inclure les commandes ou les réponses échangées. Conséquemment, il importera tout autant de savoir où ces trames commencent et quand elles prennent fin. Cela est illustré dans la synchronisation des trames. Soulignons que la synchronisation des octets doit se faire avant celle de la trame.

1.5.6 Transmission asynchrone

Essentiellement, la transmission asynchrone est recommandée lorsque les données sont générées de manière aléatoire comme dans le cas de la saisie de données grâce au clavier d'un ordinateur. L'exemple type de transmission asynchrone est celui de la transmission par modem. Dans ce genre de transmission asynchrone, chaque caractère est précédé d'un bit de démarrage (*start bit*) et d'un et demi ou deux bits de parité (*stop bit*). Les polarités des bits de démarrage (0) et de parité (1) doivent être distinctes. L'état inactif (*idle*) est similaire à celui du bit de parité (*stop bit*), soit (1). La première transition de 1 à 0 est utilisée par le récepteur pour marquer le début de tout nouveau caractère. De la sorte, le récepteur peut être synchronisé dans l'attente de la réception des bits suivants. Lorsque nous envisagerons le protocole LLC (*Logical Link Control*/Protocole de liaison logique) dans le chapitre 3, nous aurons l'occasion de nous pencher sur un cas fort intéressant.

1.5.7 Transmission synchrone

On optera pour la transmission synchrone dans le cas où il y a transmission de longs blocs de données à haute vitesse. Le bloc ou la trame de données est transmis sous la forme de séries de bits contigus et évite ainsi toute interruption entre les éléments de 8 bits. En vue de permettre au récepteur une bonne synchronisation, il faut que :

- la série de bits transmise soit bien encodée pour que le récepteur puisse assumer la synchronisation des bits ;
- toutes les trames soient précédées d'un ou de plusieurs octets réservés (sync) pour être sûr que le récepteur interprète adéquatement la réception

de la série de bits émis sous la forme de caractères ou d'octets (synchronisation des caractères);

- le contenu de chacune des trames se trouve encapsulé entre des caractères réservés à la synchronisation de trames.

1.5.8 Techniques de codage de données

Les bits transmis sont encodés afin de transmettre l'information et la synchronisation. Le codage permet également la détection de certaines erreurs de bits. Il existe plus d'une technique de codage et chacune d'entre elles convient à un système donné. Une de ces méthodes, le NRZ (Non-retour à zéro), est la représentation classique des niveaux de bit, zéro et un, revêtant respectivement les formes d'absence d'impulsion et d'onde carrée. Des exemples seront donnés dans différents chapitres.

1.5.9 Fréquence d'accès

La transmission des données sur un seul et même conduit crée un problème d'accès, surtout dans le cas où un grand nombre de stations décident toutes de transmettre au même moment. On peut se servir du multiplexage par répartition en fréquence dans le cas d'une transmission simultanée. Il faudra alors allouer une fréquence distincte pour chacun des accès, ce qui évitera qu'il y ait surcharge sur un même canal. L'exemple type de ce qui précède est la transmission par câble.

Par ailleurs, il est possible de permettre l'accès à plus d'une station en n'ayant recours qu'à un seul et même conduit et en utilisant une seule fréquence de base. Pour cela, il suffit d'y définir une méthode qui permettrait à chacune des stations de transmettre de façon non simultanée. En d'autres termes, une fois la permission obtenue, chaque station devra être la seule à transmettre sur la ligne. La transmission se fait par bande de base (*Baseband*).

La méthode de la bande de base est plus simple que la large bande lorsque la communication a lieu dans les deux sens du conduit. Le réseau local en est un bel exemple. La gestion de l'attribution des fréquences n'est pas nécessaire comme cela pourrait l'être dans le cas de l'usage de la large bande (*Broadband*).

1.5.10 Réseaux diffusés

Par définition, tous les réseaux diffusés sont des réseaux où tous les terminaux assument la fonction de réception de tous les signaux transmis. Seulement, le

rôle de chacun des terminaux est d'établir et de savoir si la communication reçue lui est destinée. On identifie trois types de réseaux diffusés :

- les réseaux de radio paquet;
- les réseaux satellites;
- les réseaux locaux.

1.5.11 Types d'informations

Dans un réseau de communication, il est possible de transmettre trois types d'informations distinctes, à savoir :

- ***le son*** : à l'instar de la voie téléphonique qui est une information dont la fréquence oscille entre 300 Hz et 3,4 kHz ;
- ***les données*** : information codée (textes, programmes, tableurs, etc.) qui peut être transmise peu importe la vitesse;
- ***l'image*** : information graphique (vidéo et diffusion) qui se caractérise par sa large bande passante et par sa vitesse.

1.6 Codage des caractères et des symboles

La technique de codage, traitée précédemment dans le présent chapitre, n'a rien à voir avec le codage des caractères à proprement parler. En effet, le codage de caractères est la représentation de l'information à un palier supérieur. Il ne représente pas des bits de transmission. C'est plutôt la séquence de bits qui représente les caractères ou symboles qui constituent l'information textuelle. De plus, le codage de caractères est défini conformément aux normes de l'UIT IA5 (*International Alphabet No 5*) ou d'après les normes de-facto (ASCII, EBCDIC).

1.7 Communications semi-duplex et duplex intégrale

On parle de communication semi-duplex (*Half-Duplex)* dans le cas où ladite communication ne se fait que dans un sens à la fois : on dit qu'elle est unidirectionnelle. On parle de communication duplex intégrale (*Full-Duplex*) dans le cas où cette communication peut se faire dans les deux sens simultanément : on dit qu'elle est bidirectionnelle.

1.8 Multiplexage par répartition dans le temps ou multiplexage temporel

La technique de multiplexage par répartition dans le temps (MRT) en est une utilisée dans les réseaux numériques. Cette technique est également connue comme multiplexage temporel. Il existe de nombreux paliers distincts de MRT. La figure 1.9 donne un bref aperçu du MRT pour la transmission qui se conforme à la hiérarchie plésiochrone numérique, ou PDH (*Plesiochronous Digital Hierarchy*), à laquelle appartiennent les T1, T2 et T3.

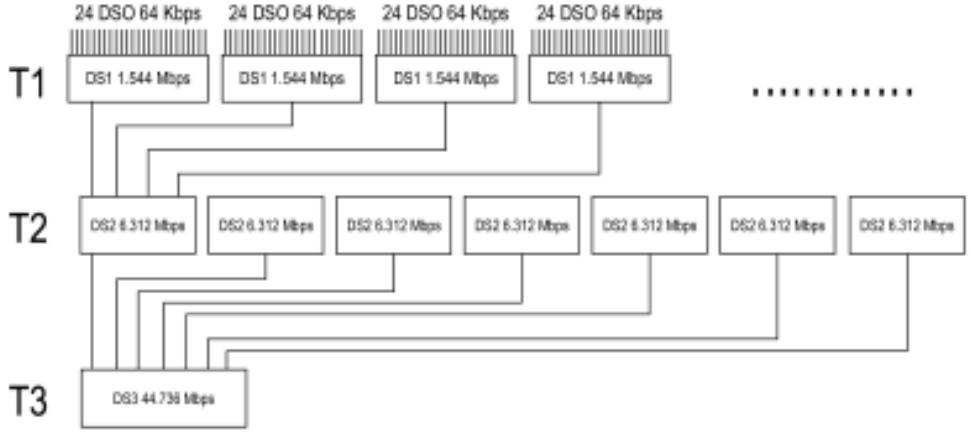


Figure 1.9 Hiérarchie plésiochrone numérique

Le T1 est un canal dont le niveau de multiplexage comporte 24 canaux, chacun d'entre eux capable d'acheminer la voix numérisée à une vitesse de 64 Kbps et, notamment, les DS-0. Le signal du T1, ou DS-1, a une vitesse de 1.544 Mbps. Ce signal est composé de 24 x 64 Kbps, soit 1.536 Mbps plus 8 Kbps réservé à l'en-tête (*overhead*).

Chacun des canaux DS-3 inclut 28 canaux DS-1 outre l'en-tête. Le tableau 1.2 nous montre les structures de chacun de ces signaux plésiochrones.

Tableau 1.2 Signaux plésiochrones

Désignation	Ordre	Mux (US)	Mux (Europe)
DS-0	0	24	30
DS-1	1	4	4
DS-2	2	7	4
DS-3	3	4	4
DS-4	4		

Tout comme le multiplexage en fréquence, le multiplexage temporel peut se faire par étapes successives conformément à une hiérarchie préétablie.

Le signal numérique de base, désigné par le sigle DS-0, correspond à une seule voie téléphonique de 64 Kbps (DS est l'acronyme pour *Digital Signal*). Dans la hiérarchie d'Amérique du Nord, le premier niveau désigné par DS-1 comporte 24 voies téléphoniques. Chacune occupe un espace-temps de 125 microsecondes dans le train d'impulsions numériques émises dans le circuit. Ce dernier a un débit de 1544 Kbps.

En Europe et dans le reste du monde, le premier niveau DS-1 comporte 32 espaces-temps de 125 microsecondes dont 30 sont des canaux de voies téléphoniques. Ce DS-1 ou E1, tel qu'il est communément connu, a un débit de 2 048 Kbps.

Sur les artères où le trafic est dense, ces signaux sont multiplexés pour former des signaux d'un ordre plus grand plutôt que d'être transmis directement. Dans le contexte Nord américain, lorsque quatre signaux DS-1 sont combinés, nous pouvons obtenir un signal du 2e ordre, soit un DS-2, qui comporte 96 voies téléphoniques.

Alors qu'on s'attendrait à ce que son débit soit de 6.176 Mbps (4 x 1.544 Mbps), ce dernier est en réalité de 6.312 Mbps, les 136 Kbps supplémentaires servant à des fins de synchronisation et d'exploitation de la liaison.

Par la suite, ce qui est le plus courant, il est possible d'envisager l'étape du multiplexage de sept signaux DS-2 pour former un signal du 3e ordre ou DS-3. Ce dernier comporte 672 voies téléphoniques et son débit est de 44.736 Mbps.

Dans le cas de certaines applications, on peut multiplexer six DS-3 et obtenir de la sorte un signal du 4e ordre soit un DS-4. Ce dernier comporte 4032 voies téléphoniques et son débit est de 274.176 Mbps.

Pour sa part, SONET (*Synchronous Optical Network*/Réseau optique synchrone) est un autre exemple de MRT plus synchronisé que le PDH.

1.9 Standardisation

La standardisation des réseaux locaux a été parrainée par l'IEEE (*Institute of Electrical and Electronic Engineers*). La normalisation des réseaux locaux eut pour but d'utiliser les couches 1 et 2 du modèle de référence de l'OSI et de les adapter aux besoins des topologies diverses. Le IEEE a mis sur pied le comité nommé groupe 802.

Trois méthodes d'accès ont été élaborées et mises au point par un comité du groupe 802 pour la sous-couche MAC (*Medium Access Control*/Contrôle d'accès au support) de la couche liaison, notamment le CSMA/CD (*Carrier Sense Multiple Access/Collision Detection*), ou réseau à accès multiples avec détection de porteuse, le bus à jeton (*Token Bus*) et l'anneau à jeton (*Token Ring*).

Une autre sous-couche de la couche liaison, commune à tous les réseaux locaux (LAN) et connue sous le nom de *Logical Link Control* (sous-couche de contrôle de liaison logique), a été conçue et élaborée pour le transfert des données par un autre comité du groupe 802.

1.10 Exercices

1.1

Chacune des commutations (circuits, paquets et messages) est plus avantageuse dans certaines situations. Décrivez ces situations pour chacune des commutations citées et les illustrer par des exemples.

1.2

Faites une recherche pour déterminer l'utilisation courante de paires torsadées, par comparaison avec la fibre optique, dans le cadre de leurs utilisations tant domestiques que commerciales. Quel est le facteur prédominant qui permettra de faire un choix entre l'un ou l'autre des conduits?

1.3

Peut-il y avoir une synchronisation de trames sans synchronisation de bits? Expliquez clairement pourquoi ceci est ou n'est pas possible.

1.4

a) Comment se fait la synchronisation lors d'une transmission asynchrone?
b) Comment les terminaux impliqués s'échangent-ils des données lors d'une communication asynchrone?

1.5

Quelle est la différence qui distingue le codage des données du codage des caractères?

1.6

Sept signaux DS-2 sont multiplexés pour former un signal DS-3 de 44.73 Mbps. Quelle est la surcharge (*overhead*) incluse dans le signal DS-3?
Trouvez la surcharge d'un signal DS-4 composé de plusieurs signaux DS-3.

1.7

Le signal T1 a un débit de 1.544 Mbps. Chaque trame est transmise toutes les 125 microsecondes.

a) Déterminez le nombre de trames transmises par seconde.
b) Déterminez le nombre de bits dans chacune de ces trames.

1.8

Le signal E1 a un débit de 2.048 Mbps. Chaque trame est transmise toutes les 125 microsecondes.

a) Déterminez le nombre de trames transmises par seconde.
b) Déterminez le nombre de bits dans chacune de ces trames.
c) Si seulement 30 espaces-temps sur 32 sont utilisés pour transporter des canaux de voies, déterminez combien de bits de chaque trame sont réservés à l'en-tête.

2 Modèle de référence OSI

2.1 Introduction

Les caractéristiques techniques de la transmission des données sont les mieux définies dans les principes d'architecture de systèmes ouverts connus sous le nom de OSI (*Open System Interconnection*/Interconnexion de systèmes ouverts*)* ou encore sous celui de modèle de référence OSI (RM-OSI) développé également par ISO (*International Standard Organization* /Organisation internationale de normalisation).

L'UIT-T, connue auparavant sous le nom de Comité consultatif international pour la téléphonie et la télégraphie (CCITT), a entrepris au cours des années 1980 de parachever ledit modèle de référence. La tâche incomba à la commission d'étude VII et un rapporteur, spécialement désigné, eut la lourde responsabilité de mener à terme la normalisation de ce modèle. Les « recommandations » (terme utilisé par l'UIT-T pour désigner les normes) alors émises ont été identifiées par la série X.200. Nous traiterons ultérieurement des recommandations propres à la série X.200.

Il a fallu définir une architecture pour la transmission des données dans un réseau de commutation par paquets. Au cours de l'élaboration du modèle de référence OSI, les contraintes suivantes ont été prises en considération :

1. **optimisation des échanges** entre terminaux ou machines normalement fournis par divers manufacturiers;
2. **indépendance du service** et de l'interconnexion au réseau;
3. **restriction des modifications** nécessaires au développement du service pour les fonctions exigées lors de la livraison de ce service.

2.1.1 Optimisation des échanges

La compatibilité entre machines constitue le premier des soucis de la communication. Il est impératif que deux machines communicantes aient le

même mécanisme d'accès pour qu'elles puissent se comprendre. Il importe tout autant que ces accès et échanges puissent se faire sans contraintes de propriété ou de brevet pour que les terminaux, de provenance distincte, puissent se parler.

Ainsi, il faut que la communication soit fondée sur la compatibilité. En d'autres termes, plutôt que de se fier aux manufacturiers, il vaut mieux se baser sur les protocoles normalisés et ouverts. De la sorte, le terminal source doit tout simplement savoir si le terminal de destination a recours au même protocole que lui pour communiquer et non se demander s'il est du même manufacturier et du même modèle que lui. Ce qui en fait un protocole ouvert et libre de tout droit.

2.1.2 Indépendance du service

Les échanges entre terminaux sont indispensables pour les services offerts ou demandés par l'un ou l'autre des terminaux. La nature de ces services et surtout leur programmation ne doit en aucun cas dépendre du genre de réseau auquel on a recours

Ainsi, le service assumé par le courrier électronique a pour finalité celui de relayer un message depuis la source jusqu'à la destination. Ce message devra pouvoir passer à travers plus d'un type de réseau sans pour autant en être conscient voire en être modifié. On peut donc dire que le courriel ne dépend pas du type de réseau.

2.1.3 Restriction des modifications

Tous les services sont soumis à des mises à jour mineures ou majeures. Celles-ci sont faites au long de leur cycle de vie. Bien évidemment, il faut donner à ces services l'occasion de procéder aux modifications sans pour autant perturber le réseau transporteur. Il serait impensable de demander des modifications au fonctionnement du réseau pour permettre que les changements soient apportés au service.

De la même façon, les mises à jour ou les modifications apportées au réseau transporteur ne doivent en aucune manière porter préjudice au bon fonctionnement des services. Il serait tout aussi impensable de devoir demander à tous les services de procéder à des changements tout simplement parce que le réseau fait sa mise à jour.

2.2 Couches fonctionnelles

Le modèle de référence OSI distribue les fonctions de communications dans un ensemble de couches afin d'avoir un meilleur contrôle sur le système qui y a recours. Par contre, les règles qui suivent ont été mises en application :

- *Homogénéité dans les fonctions de chaque couche;*
- *Définition des couches entre lesquelles les interactions sont réduites au minimum;*
- *Restriction du nombre de couches fonctionnelles, ramené à une valeur acceptable.*

2.2.1 Homogénéité

Tout équipement de communication comporte un bon nombre de fonctions lui permettant de mettre l'information en forme afin de la transmettre. Chacune de ces fonctions a un rôle et un objectif précis. Certaines d'entre elles assument les besoins de services tout comme dans le cas du courrier électronique ou bien dans celui de la préparation du transfert de fichiers. D'autres n'ont pour tout objectif que celui de coder l'information. D'autres encore assurent la fiabilité du transfert ou le routage des trames. Il y en a enfin certaines qui préparent les bits et les encodent sous la forme d'impulsions électriques, voire lumineuses.

Soulignons que toutes ces fonctions sont étagées et réparties au travers de couches fonctionnelles et que chacune de celles-ci inclut un certain nombre de fonctions. Ces dernières sont en rapport direct avec l'objectif de cette couche.

Il est primordial que chacune des fonctions soit partie intégrante d'une couche. Par définition, il ne sera pas possible qu'une fonction se trouve répartie sur deux couches adjacentes. Cette condition, qui en est une d'homogénéité des couches, est indispensable si l'on désire préserver l'autonomie des couches les unes envers les autres.

2.2.2 Interactions entre couches adjacentes

Les couches adjacentes sont autonomes entre elles. Il n'y a de communication qu'entre des couches homologues de deux systèmes communicants. La communication se fait par le protocole propre à la couche concernée. Toutefois, et pour pouvoir communiquer, les couches de même niveau de systèmes distants ont recours aux services de la couche sous-jacente.

La règle qui permet à toute couche de conserver son autonomie par rapport aux autres couches repose sur le principe d'une interaction réduite d'avec les couches adjacentes. Néanmoins, cette interaction qui doit tout de même avoir lieu revêt la

forme du système postal. À titre d'exemple, lorsque le courrier nous parvient, de qui que ce soit, il se crée un lien direct entre la personne qui nous adresse le courrier et nous-mêmes. La communication est donc bien établie et de façon directe. Nous faisons abstraction du facteur qui est l'intermédiaire. L'interaction avec le facteur est minime. Nous ne disons pas : « j'ai reçu une lettre du facteur », mais bien plutôt : « j'ai reçu une lettre d'untel ».

Il en va de même dans le modèle des couches de communications où l'interaction des couches adjacentes est tout aussi insignifiante. Chacune de ces couches a besoin de faire appel aux services de la couche adjacente inférieure. Chacune d'entre elles offre aussi ses services à la couche adjacente supérieure.

2.2.3 Restriction du nombre de couches

Le concept des couches est fort utile et permet, grâce à l'indépendance de chacune de ces couches, de limiter la complexité de la communication. Il est par contre important de définir le nombre de couches.

Considérons un nombre assez grand de couches. À cause de la restriction d'homogénéité, le nombre maximal de couches ne pourrait qu'être égal au nombre de fonctions. Dans ce cas il y aurait autant de couches que de fonctions et alors le concept de couche n'aurait aucune raison d'être.

Envisageons l'autre extrême : soit, l'existence d'une seule couche qui inclurait toutes les fonctions. Encore dans ce cas-là, le concept de couche n'aurait aucune raison d'être.

Aussi, ce qu'il nous reste à faire est d'établir un nombre de couches qui sera acceptable. Dans le modèle OSI, ce nombre est fixé à sept. Par contre, dans le modèle Internet, il n'est que de cinq.

2.2.4 Définition des couches

Afin de répartir les fonctionnalités au niveau système, les sept couches du modèle OSI ont été divisées en deux parties, soit :

- **Les couches inférieures** : les couches de 1 à 3 et qui traitent du transfert d'informations entre terminaux à travers un réseau de télécommunication donné;
- **Les couches supérieures** : les couches de 4 à 7 et qui assument les applications prises en main par les terminaux.

2.2.5 Couches inférieures

Parmi les couches inférieures, on retrouve trois couches dont les noms sont : couche physique, couche liaison et couche réseau.

Les couches inférieures assument l'acheminement des informations à travers le réseau de données. Elles en assurent tout à la fois l'accès au réseau et le transit des informations.

2.2.6 Couches supérieures

Parmi les couches supérieures, on retrouve les quatre couches suivantes : la couche transport, la couche session, la couche présentation et la couche application. Signalons que dans le modèle Internet, seules deux couches existent, soit : les couches transport et application. Nous retrouvons ces couches dans les terminaux. Elles permettent d'établir la communication de bout en bout entre deux ou plusieurs interlocuteurs.

Étant donné que chaque terminal comprend tant les couches inférieures que supérieures, celui-ci peut, d'une part, établir l'accès au réseau et, de l'autre, procéder à un échange des données de bout en bout. Les protocoles des couches supérieures sont connus comme étant des protocoles bout en bout.

2.2.7 Les sept couches du modèle OSI

Chacune des sept couches, illustrées à la figure 2.1, comporte des fonctions spécifiques homogènes. Nous donnons ci-dessous un bref aperçu de chacune d'entre elles.

Couche 1 - La couche physique : La couche physique inclut toutes les fonctions qui ont trait à la transmission des bits. Elle traite donc essentiellement du codage des bits, du multiplexage, de l'entrelacement des trames et du protocole permettant d'assurer la synchronisation. De plus, la couche physique calcule et établit la tension de référence à laquelle elle aura recours lors de la transmission, de la vitesse de transmission et par voie de conséquence du temps que les bits mettront lors de la transmission et du genre de connecteurs utilisés. Enfin, la couche physique traite de la synchronisation de transmission des bits, mais aussi de celle des octets et des trames. Elle ne se soucie guère de la signification de la charge utile. La couche physique demeure uniforme tout au long du lien point à point.

Couche 2 - La couche liaison : La couche liaison procède au transfert des informations sous forme de trames et permet de détecter et de procéder à la rectification d'erreurs. La couche liaison n'établit que le lien point à point entre deux éléments du réseau directement reliés par un lien physique (avec ou sans fil). Les fonctions de la couche liaison sont définies grâce à des protocoles précis et propres à cette couche.

Couche 3 - La couche réseau : La couche réseau a pour fonction d'assumer tant l'établissement que l'interruption des communications à travers le réseau. Elle se charge en fait du contrôle et du flux des communications dans le réseau.

Couche 4 - La couche transport : La couche transport a pour fonction d'acheminer les informations d'un bout à l'autre de la connexion et d'assumer la procédure de la connexion comme telle. Elle répond de l'accès et du service des applications dans le cas du réseau Internet.

Couche 5 - La couche session : La couche session assure l'organisation de l'échange et la structure du dialogue entre applications.

Couche 6 - La couche présentation : La couche présentation a pour rôle d'établir, d'une part, la syntaxe qui sera utilisée lors des échanges d'informations et, de l'autre, les systèmes de sécurité relativement à l'accès aux informations.

Couche 7 - La couche application : La couche application a pour rôle d'établir le système en cours, lequel peut être applicable à divers types de services. Pour avoir accès à l'OSI, l'utilisateur passe par ce service. En règle générale, les utilisateurs sont des programmes qui ont besoin de procéder à une transmission et de ce fait, d'envoyer des informations.

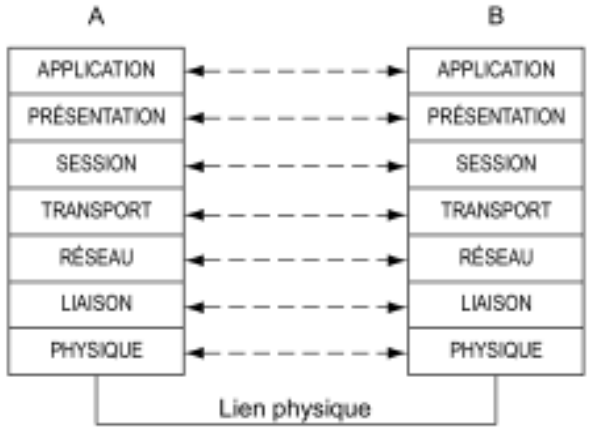


Figure 2.1 Représentation des sept couches

2.3 Communication et interaction

Les couches de même niveau entre deux terminaux distants communiquent afin d'échanger des informations. En tout premier lieu, nous allons émettre l'hypothèse suivante : soit deux terminaux qui après avoir établi une connexion dans le réseau sont prêts à s'échanger des données. Il importe de savoir que la communication ne peut avoir lieu qu'entre des couches du même niveau et sur des systèmes distants. Dans le cas de la figure 2.2, la couche transport du terminal A ne peut communiquer qu'avec son homologue du terminal B. Cette remarque s'applique à toutes les autres couches telles qu'illustrées à la figure 2.1. Toutefois, il est donné à chacune de ces couches de pouvoir interagir avec des couches directement adjacentes. À titre indicatif, soulignons que dans ce cas, la couche transport du terminal A peut interagir avec la couche réseau ou la couche session du terminal A uniquement.

Toute communication implique une certaine compréhension relative à l'échange des éléments. À supposer que la couche transport du terminal A veuille établir une communication avec celle du terminal B, elle va lui adresser, par l'en-tête, les informations qui vont permettre au récepteur B de répondre aux attentes du transmetteur A. On désigne l'information transmise par l'acronyme PDU (*Protocol Data Unit*/Unité de données de protocole). Le PDU inclut des données de même que les informations relatives au contrôle de la couche où il se trouve. De ce fait, il est le résultat d'une encapsulation qui consiste à ajouter un en-tête aux données. Les données quant à elles sont également dénommées SDU (*Service Data Unit*/Unités de données de service) et l'en-tête est connu sous PCI (*Protocol Control Information*/Informations de contrôle du protocole).

En résumé, le PDU se compose du SDU et du PCI et nous pouvons dès lors écrire : PDU = SDU + PCI.

Il est bon de noter aussi que les PDUs ne sont compris qu'entre des couches du même niveau de deux terminaux distants.

L'interaction qui prévaut entre les couches adjacentes permet de transmettre ou de recevoir les services de ces couches. Si la couche transport de A veut procéder à la transmission d'un PDU vers le terminal B, elle devra recourir aux services de la couche réseau de A qui lui est sous-jacente.

Dans l'exemple donné précédemment, la couche transport est obligée d'avoir recours à un messager pour véhiculer son PDU désigné par TPDU (transport PDU). Ainsi, la couche transport s'adresse au service de la couche réseau du terminal A pour permettre au TPDU de sortir éventuellement du terminal A et s'acheminer vers son point de destination à travers le réseau.

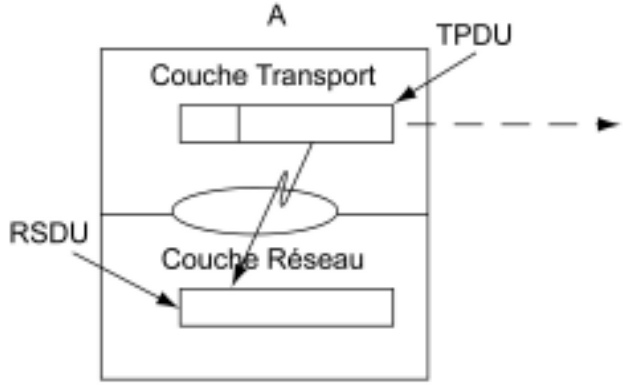


Figure 2.2 TPDU et RSDU

La couche réseau A qui reçoit alors le TPDU de la couche transport A ne perçoit ce dernier que comme un amas de données qualifié d'unité de données de service (SDU), lequel est alors seulement identifié par RSDU pour signifier qu'il s'agit du SDU de la couche réseau.

Dès lors, la couche réseau du terminal A fait ce que la couche transport du même terminal a fait, soit l'encapsulation du RSDU reçu auquel elle ajoute un RPCI afin de donner naissance à un nouveau PDU propre à la couche réseau, qualifié de RPDU.

À son tour, la couche réseau du terminal A doit procéder à la transmission du RPDU vers la couche réseau du terminal B avec laquelle elle établit une communication. Pour parvenir à ce but, la couche réseau du terminal A va requérir les services de la couche liaison A qui lui est sous-jacente. Ceci est illustré dans la figure 2.3. Tout comme nous l'avons vu dans le cas de la couche transport, la procédure va se répéter et un mécanisme s'enclenchera.

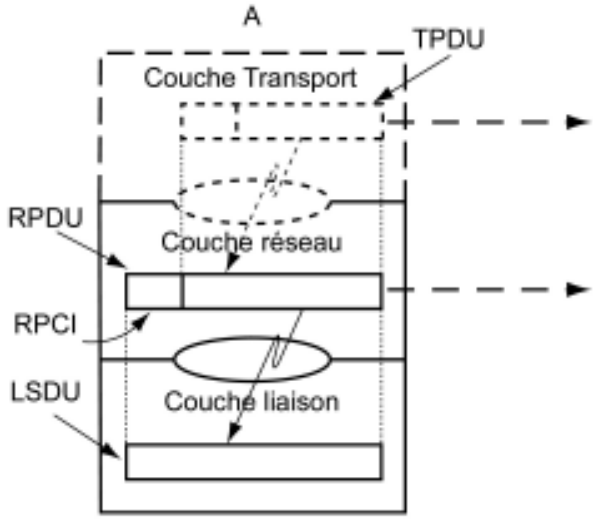


Figure 2.3 RPDU et LSDU

Tentons de saisir ce qui se produit lors de la réception des données par le terminal B tel qu'illustré dans la figure 2.4. Prenons le cas suivant : la couche réseau du terminal B a reçu le RPDU transmis par la couche réseau du terminal A. Dans un premier temps, la couche réseau du terminal B va désencapsuler le RPDU reçu et en extraire le RSDU. Une fois ce travail fait, le SDU est alors envoyé à la couche supérieure (offre de service) sans se soucier de sa teneur. Pour sa part, la couche transport du terminal B reçoit de la couche réseau du même terminal, ce qu'elle identifie comme étant un TPDU. Dans les faits, ce TPDU n'est nul autre que celui qui a été transmis par la couche transport du terminal A.

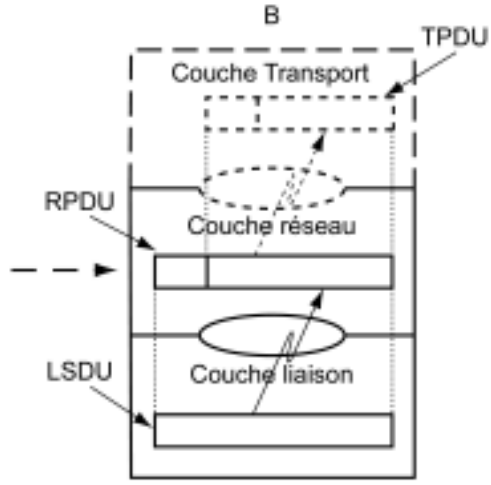


Figure 2.4 Réception du RPDU

Ce phénomène est celui qui se produit pour chacune des couches. En règle générale, nous sommes à même d'établir la tenue des mécanismes suivants :

a) La couche chargée de la transmission des données (SDU) à son homologue (couche de même niveau d'un terminal distant) procède à l'encapsulation du SDU et y ajoute un PCI (l'en-tête) pour en faire un PDU. Par la suite, il s'adresse à la couche sous-jacente requérant le service suivant : celui d'acheminer le PDU. Pour sa part, la couche sous-jacente reçoit ce qu'elle qualifie de SDU et va procéder à la même étape que ci-dessus.

b) La couche réceptrice reçoit le PDU transmis par son homologue avec le concours de la couche inférieure. Cette couche va désencapsuler le PDU et en retirer le SDU qui s'y trouve. Par ailleurs, elle va offrir ses services à la couche supérieure en lui remettant ce SDU. La couche supérieure y reconnaîtra son PDU transmis par son homologue.

Attendu que nous avons vu comment de tels mécanismes fonctionnent, nous allons, dans un deuxième volet, tenter d'en saisir l'utilité lorsqu'un terminal a accès à un réseau de commutation par paquets et qu'il veut procéder à la transmission des données vers un autre terminal branché au réseau.

Le terminal source doit tout d'abord établir un contact avec un commutateur d'accès et s'y relier. Ce lien physique doit permettre de transférer des bits encodés par le transmetteur et de les décoder par le récepteur. Autant le transmetteur que le récepteur de chacun des accès doivent posséder les mêmes

principes d'encodage et de décodage. Le terminal source tout comme le commutateur d'accès devront disposer d'encodeurs et de décodeurs identiques. De la sorte, tant l'envoi que la réception des bits se feront de manière très efficace. L'envoi et la réception des bits se font par le biais de la couche physique du terminal et du commutateur d'accès. Ces couches communiquent entre elles et ont pour fonction de procéder à un échange de bits synchronisés et sans erreur : c'est ce que l'on qualifie d'échange point à point.

Le terminal doit également établir un lien et se brancher au commutateur[1] d'accès. Pour cela, il a recours à la couche liaison qui donnera au terminal la possibilité d'établir une connexion avec la couche liaison du commutateur d'accès en lui faisant parvenir une demande de connexion LPDU conformément au protocole en cours comme les protocoles HDLC (*High-level Data Link Control*/Procédure de commande à haut niveau) ou LLC (*Logical Link Control*/Procédure LLC) dont nous traiterons ultérieurement dans le chapitre 3. Advenant le cas où la couche liaison du commutateur d'accès accepte la demande de connexion, un lien s'établit et les données peuvent être envoyées entre les couches liaison. Ces données proviennent de plusieurs SDU que la couche supérieure, notamment la couche réseau, a générée. Une fois le lien entre les couches liaison établi, le terminal source remet au commutateur d'accès les informations qui lui permettront de parvenir jusqu'au terminal de destination : ces informations comprennent l'adresse. Cette tâche incombe à la couche réseau. Le terminal source va recourir à sa couche réseau pour transmettre l'adresse à la couche réseau du commutateur d'accès. On peut aussi établir une connexion entre ces couches avant de procéder à l'envoi des données des RPDUs.

Une fois le commutateur d'accès est en possession de l'information nécessaire pour passer à l'étape du routage par le biais de la couche réseau, il convient de prendre la voie la plus rapide et la moins congestionnée pour établir un lien avec le terminal de destination. Cependant, il doit avant tout établir les liens de la couche physique, de la couche liaison et de la couche réseau avec le commutateur d'accès suivant, identifié lors de son cheminement, alors qu'il se rend vers le terminal de destination.

Tout commutateur qui se trouve le long de ce parcours doit exécuter les mêmes opérations jusqu'à parvenir au terminal de destination. Une fois ce dernier contacté, la couche transport établit une connexion de bout en bout des deux terminaux et entame la transmission des données envoyées par des couches supérieures. Les deux terminaux communiquent directement par le biais de la

[1] Commutateur est considéré ici comme un équipement qui peut faire la commutation des couches 2 et 3.

couche transport. Le réseau n'est en aucune manière concerné par les échanges de cette couche. C'est ce que la figure 2.5 illustre.

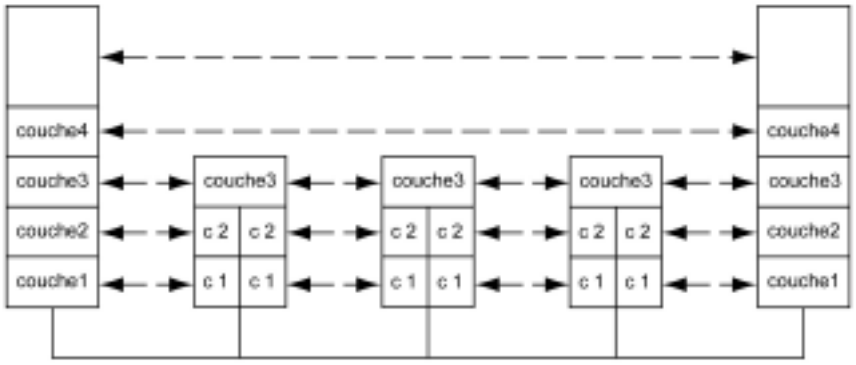


Figure 2.5 Communication bout en bout

2.3.1 Concept de service

Nous avons traité précédemment des interactions entre couches adjacentes. Si minimes sont-elles, ces interactions sont nécessaires. La ligne de démarcation ou interface des couches adjacentes est qualifiée de service. Ainsi, comme l'indique la figure 2.6, il y aura autant de services qu'il y aura de lignes entre les couches. Chacun porte le nom de la couche sous-jacente. Par exemple, la ligne qui se trouve entre la couche liaison et la couche réseau sera appelée service liaison.

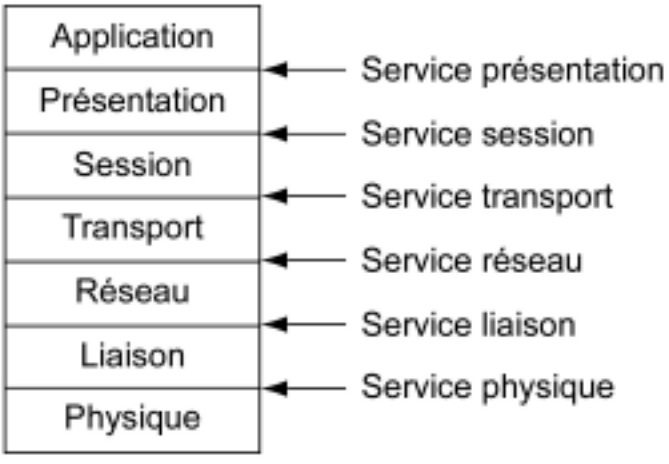


Figure 2.6 Services du modèle OSI

Ces services requièrent un point d'accès et un message pour être en mesure d'acheminer les informations entre deux couches adjacentes. Dans le modèle OSI, il s'agit de :

- SAP (*Service Access Point*/Point d'accès au service)
- Primitives

2.3.2 Point d'accès au service

Le rôle du SAP vise à identifier les interfaces des couches adjacentes d'un système donné. Chaque SAP est également identifié en fonction de la position qu'il occupe. À titre d'exemple, comme l'indique la figure 2.7, le SAP qui se trouve entre la couche 1 (physique) et la couche 2 (liaison) se dénomme : PhSAP.

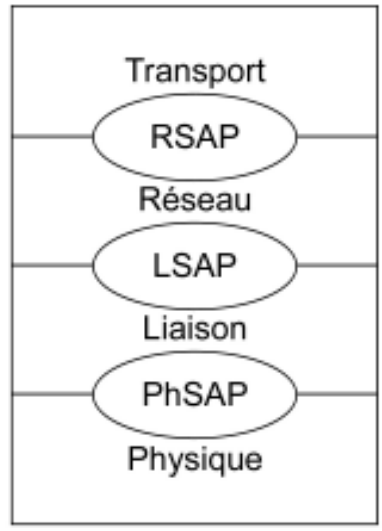


Figure 2.7 Points d'accès des services (SAP)

2.3.3 Primitives

Dans le système, les primitives, indiquées par des flèches dans la figure 2.8, sont à la source du dialogue de base établi entre couches adjacentes. On distingue quatre types de primitives : de demande (*request*), de réponse (*response*), d'indication (*indication*) et de confirmation (*confirmation*).

Les primitives de demande et de réponse sont utilisées par une couche *n* pour l'obtention d'un service donné de la couche inférieure *n–1*. Dans le cas des deux

autres primitives, soit d'indication et de confirmation, elles sont utilisées par la couche *n* pour offrir un service à la couche supérieure *n+1*.

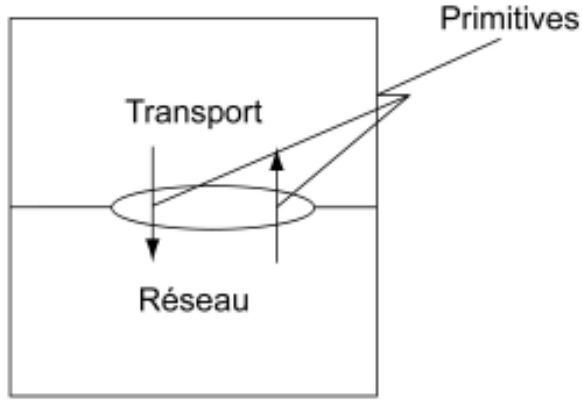


Figure 2.8 Les primitives

2.3.4 Protocoles

Les protocoles sont des règles qui permettent d'établir un dialogue entre des couches de même niveau de deux terminaux en communication. Les caractéristiques propres à ces protocoles supposent qu'il existe une répartition minutieusement établie des fonctions entre terminaux.

2.4 Les entités

Par définition, on désigne par entité le milieu où les programmes de communication de la couche dont ils relèvent sont activés. En outre, cette entité comporte un automate, ou machine de protocole. La machine de protocole réagit aux PDU reçus ou aux services échangés avec les couches adjacentes. Les fonctions de chaque entité comprennent les points suivants :

- Une entité (N) met en vigueur les fonctions et le protocole de la couche (N) pour établir la communication avec des entités (N) se trouvant dans les systèmes distants.
- Chaque entité interagit avec les entités de la couche supérieure et de la couche inférieure à la sienne à travers un SAP.
- Les services que des entités rendent à d'autres entités passent par les primitives qui spécifient les fonctions activées et transfèrent les données et l'information de contrôle.

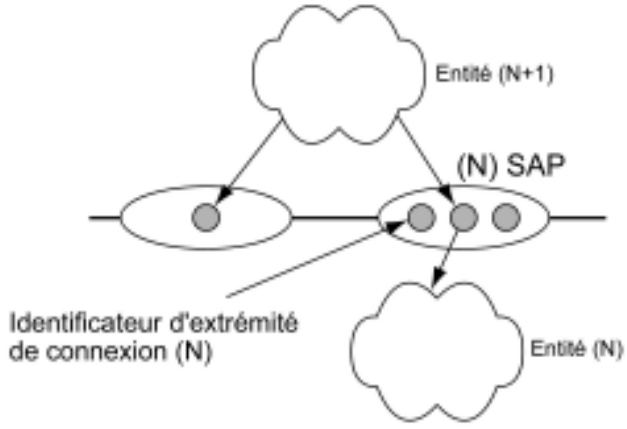


Figure 2.9 Rapports entre entités

Tel qu'illustré dans la figure 2.9, dès qu'une entité (N+1) a recours aux services d'un (N)SAP pour établir une communication avec une autre entité (N+1), l'entité (N) lui assigne, dans le (N)SAP utilisé, une extrémité de connexion. Ceci permet à l'entité (N+1) qui utilise le (N)SAP d'identifier son accès des autres accès qui utilise le même (N)SAP.

L'identificateur d'extrémité de connexion (N) comprend l'adresse du (N)SAP utilisée ainsi qu'un suffixe unique dans le (N)SAP.

2.5 Normalisation des couches

Bien que le mécanisme qui fait état du fonctionnement des couches OSI soit pratiquement le même pour toutes les couches, il importe toutefois de comprendre les fonctions réservées à chacune d'entre elles. Afin de mieux saisir les fonctions assumées par chacune d'elles, le modèle de référence OSI et sa normalisation ont permis d'établir les rôles et fonctions de chaque couche grâce à de nombreuses recommandations distinctes et diverses. Ces dernières s'inscrivent dans la série X.200 de l'UIT-T.

Un mécanisme de reconnaissance des recommandations a été appliqué dans la numérotation de la série X.200. Dans celle-ci, lorsque le chiffre des dizaines prend la valeur 1, cela signifie que la recommandation fait état d'un service; par contre, lorsqu'il prend la valeur 2, elle désigne alors un protocole. Toujours dans le cas de la série X.200, le chiffre des unités prend le numéro de la couche qu'il représente. À titre d'exemple, la recommandation X.214 représente la définition du service de la quatrième couche, ou la couche transport, conformément au

modèle de référence OSI. Ce service se trouve être entre la couche transport et la couche session. Par ailleurs, la recommandation de X.224 représente la définition du protocole de la couche transport. La figure 2.10 indique les numéros des recommandations conformément au modèle de référence de l'OSI. Notez que la recommandation X.222 n'existe pas. Le protocole HDLC est celui qui représente le protocole de la couche liaison. D'autre part les recommandations X.227, X.228 et X.229 sont spécifiques à la couche Application.

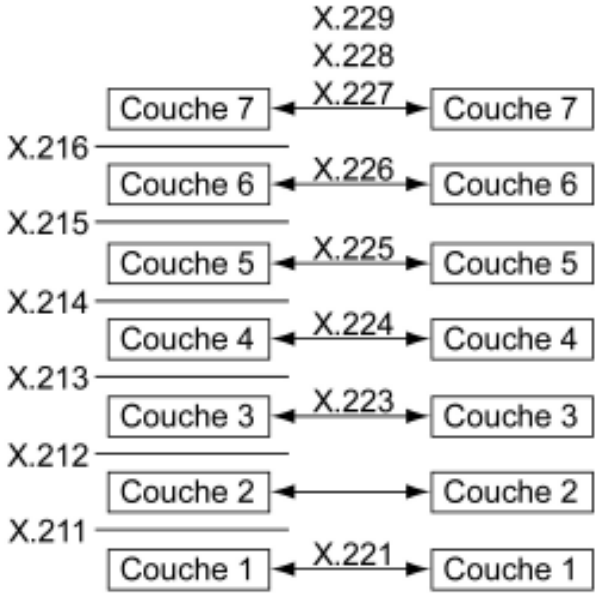


Figure 2.10 Recommandations des couches du modèle de référence de l'OSI

2.6 Encapsulation

Nous avons vu que dans le cas de deux terminaux distants, chacune des couches communique avec la couche homologue du terminal soit la même couche du même niveau. Par exemple, l'entité de la couche (N) du terminal A prépare un (N)PDU pour l'acheminer vers la couche (N) du terminal B. Nous savons également que ce PDU renferme des données qui ne sont nulles autres que le SDU (*Service Data Unit*/unité de données de service) et qu'il comporte un en-tête. également connu par le sigle de PCI (*Protocol Control Information*/informations de contrôle du protocole). Le PCI contient des instructions relativement au protocole, instructions que les entités des couches (N) des terminaux A et B s'échangent. À titre d'exemple, l'entité qui prépare un (N)PDU procède à l'encapsulation des données reçues de la couche supérieure.

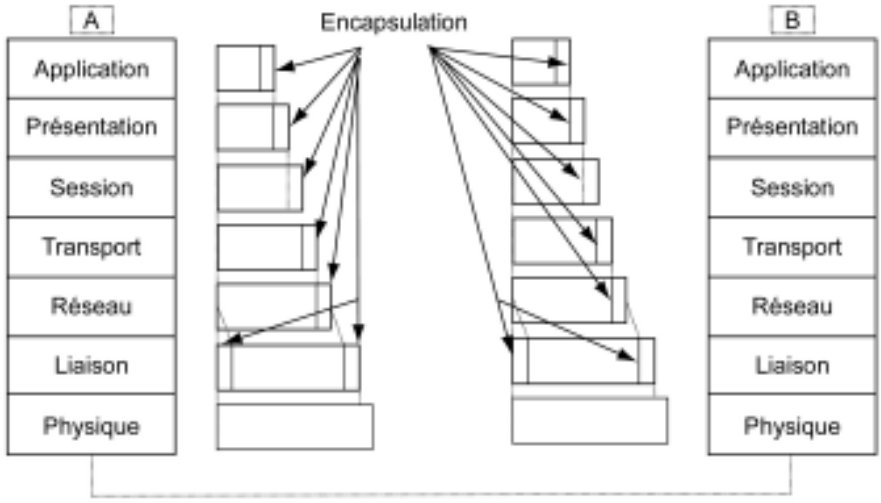


Figure 2.11 Procédure d'encapsulation - Modèle de référence de l'OSI

Cette procédure est reprise par chacune des couches (figure 2.11). En résumé, une entité de la couche (N+1) recevra un (N+1)SDU de la couche supérieure (N+2) qu'elle encapsule avec un PCI, ce qui nous donne alors un (N+1)PDU. Elle l'adresse ensuite à la couche inférieure (N) par le biais du (N)SAP. L'entité de la couche inférieure (N) reçoit ce (N+1)PDU, mais elle le considère comme un (N)SDU. Le processus similaire se répète au niveau de sa couche. La figure 2.12 illustre ce type de transformation.

Il existe un autre champ que les couches adjacentes exploitent : il s'agit du ICI (*Interface Control Information*/Informations de contrôle de l'interface). Chacun des PDUs est pris en relais par la couche adjacente avec un ICI qui remet des instructions à la couche inférieure. Cette dernière qui le reçoit et le perçoit comme un IDU (*Interface Data Unit*/Unité de données de l'interface) en extrait le ICI pour le traiter séparément. Le ICI n'a qu'une signification locale. Il représente le message d'instructions de l'entité d'une couche à l'entité de la couche sous-jacente. Le IDU est composé du ICI et du PDU.

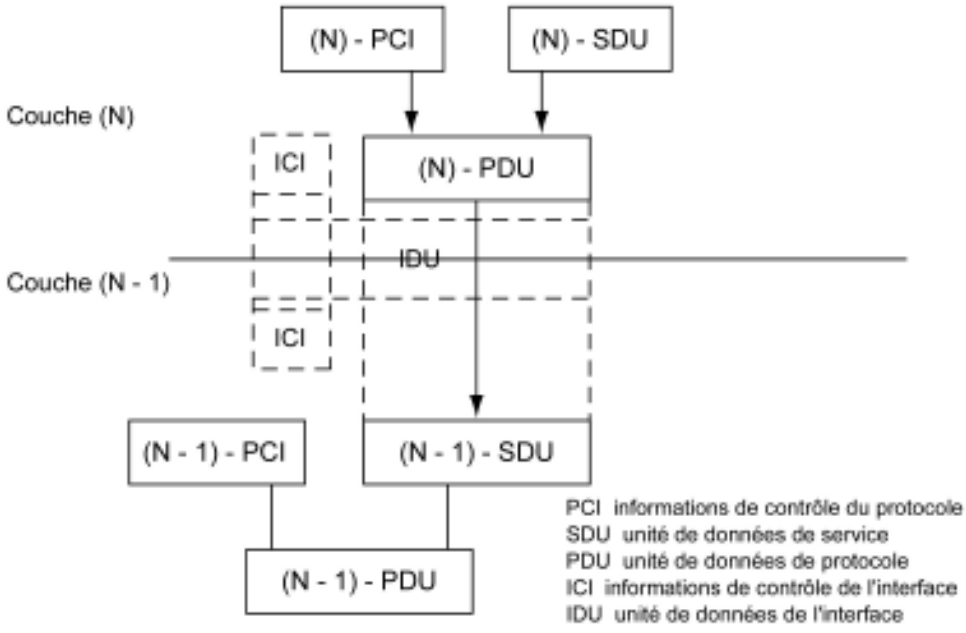


Figure 2.12 Procédure d'encapsulation

2.7 Segmentation

Lorsque les SDUs sont trop grands pour l'entité qui les reçoit, ils doivent être segmentés. Chacun de ces segments devient un nouveau SDU et est encapsulé pour former un PDU. La segmentation est nécessaire lorsque certaines couches sont limitées par la longueur de PDUs spécifiée par le réseau.

La figure 2.13 en illustre la procédure.

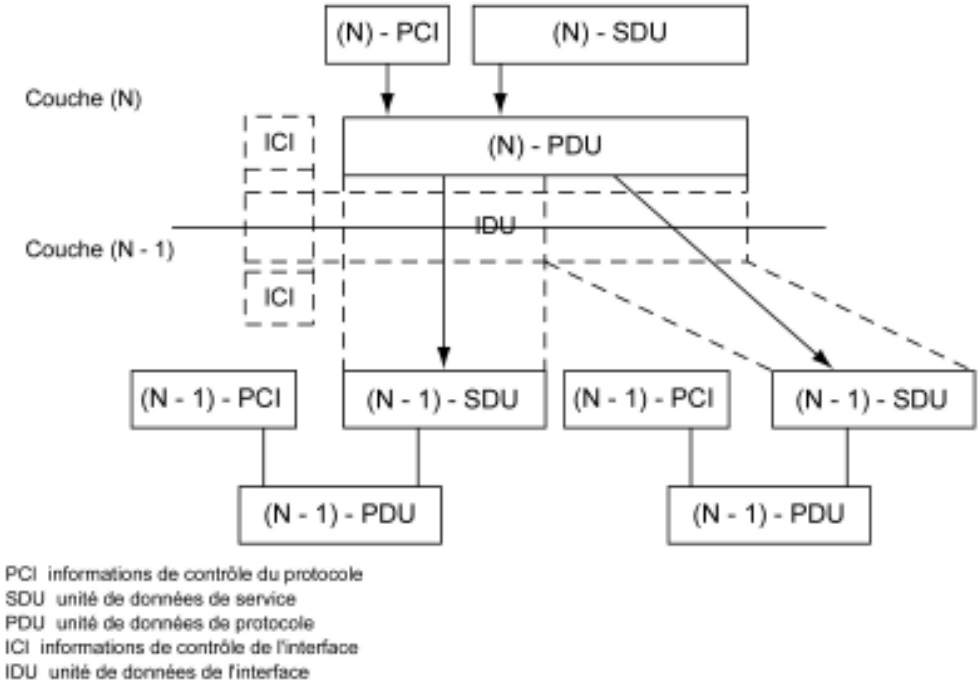


Figure 2.13 Segmentation et encapsulation

2.8 Services et primitives

Une relation prévaut entre les protocoles et les services de chacune des couches. Tout comme la figure 2.14 nous l'illustre, les PDUs engendrés par les protocoles ont recours aux services adjacents en vue de les acheminer vers leur point de destination. Les services permettent de transmettre des SDUs grâce au concours des primitives.

De la sorte, chacune de ces couches dispose d'un protocole mû par son entité lors de l'engendrement des PDUs. Chaque couche dispose également d'un service qui lui est propre et exclusif. Par ailleurs, nous avons déjà vu que le service sépare la couche (N) de la couche supérieure (N+1) et prend le nom de service (N). Les primitives et le SAP qui se trouvent sur ce service prennent également le nom de la couche (N). Lorsque nous traiterons ultérieurement des primitives, ces points n'en seront que plus clarifiés.

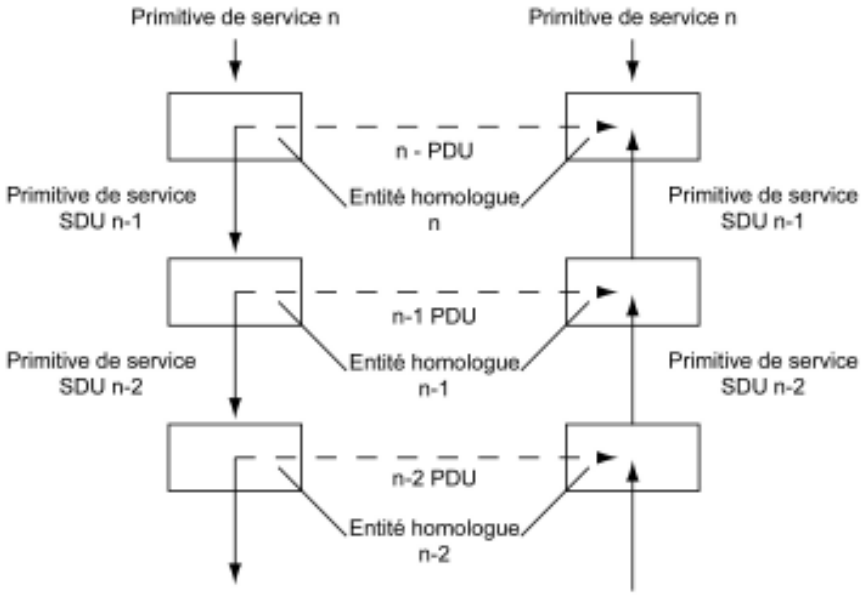


Figure 2.14 Relations entre protocoles et services

Les primitives doivent se conformer au type d'information transférée. C'est la raison pour laquelle il y a plusieurs types de primitives. Ces types de primitives permettent de savoir d'où émane l'information et de comprendre surtout, avec une très grande précision, le sens dans lequel cette information est acheminée.

Comme nous l'avons déjà mentionné, les quatre types de primitives utilisés sont les suivants :

- de demande (*request*);
- d'indication (*indication*);
- de réponse (*response*);
- de confirmation *(confirmation*).

La primitive de demande est créée par l'entité qui se trouve à l'origine de la communication d'un transfert de données formant un objet vers un système donné. L'entité en question devra recourir aux services de la couche inférieure et lui transférer son PDU à l'aide de la primitive de demande.

Le terminal distant recevra le PDU de son homologue par le biais de la couche sous-jacente. Celle-ci lui remettra ladite information sous la forme d'un SDU grâce à l'intervention de la primitive d'indication.

Le terminal distant ayant transmis le PDU peut exiger un accusé de réception. L'entité de la couche qui reçoit le PDU en génère alors cet accusé de réception. L'entité première, celle qui a déclenché cette transmission et enclenché tout ce processus, reçoit le nouveau PDU en guise de réponse au PDU qu'elle a envoyé. La figure 2.15 nous donne un bon aperçu de l'utilisation des primitives.

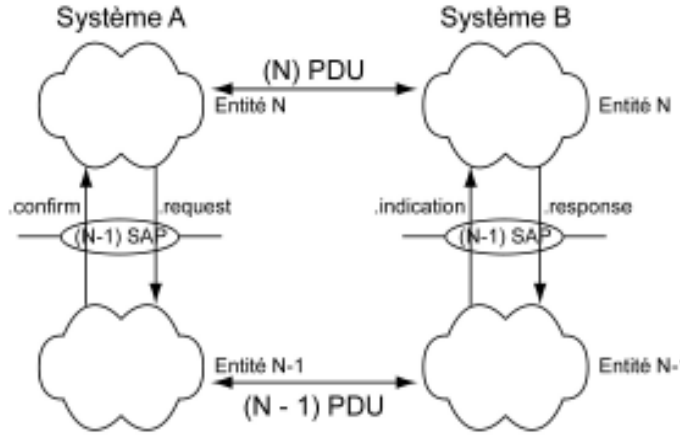


Figure 2.15 Types de primitives

2.9 Appels avec ou sans connexion

Il existe deux sortes de transfert d'informations. La première, connue sous le nom de lien orienté connexion (*connection oriented*), exige qu'il y ait établissement d'une connexion avant de procéder au transfert des données. La seconde, appelée lien sans connexion (*connectionless*), ne requiert aucun besoin d'établir une connexion avant de procéder au transfert des données. Dans le deuxième cas, les informations sont directement transmises par des PDUs.

2.9.1 Lien avec connexion

Le lien avec connexion débute par la transmission d'un PDU précis qui n'est rien d'autre qu'une démarche de connexion. Au reçu de cette demande, la couche du terminal distant émet un autre PDU de confirmation ou un de refus. En cas de refus, il ne pourra pas y avoir de communication. Par contre, dans le cas contraire et sur réception d'une confirmation, les systèmes (terminaux) sont branchés et peuvent s'envoyer des PDUs de données. Une fois que le transfert des données prend fin, l'un ou l'autre des terminaux émet une demande de déconnexion de l'appel et la confirmation de celle-ci doit être émise.

2.10 Classes de primitives

Il existe quatre classes de primitives, à savoir :

- *CONNECT*
- *DATA*
- *DISCONNECT*
- *UNIDATA*

Lorsqu'une entité de la couche (N) du terminal A est désireuse d'établir une connexion avec l'entité de la couche (N) du terminal B distant, elle transmet un PDU par lequel elle demande que la connexion soit établie. Toutefois, et pour parvenir à son but, l'entité en question doit avoir recours aux services de la couche sous-jacente. Tout comme nous l'avons déjà vu, la couche (N) envoie une primitive de demande et de classe *CONNECT* à la couche inférieure. En d'autres termes, l'entité (N) envoie une primitive *CONNECT.request* par un SAP, soit le (N-1)SAP. *L'entité* de la couche inférieure (N-1) prend connaissance de la connexion demandée par la couche (N).

Faisons un bref retour en arrière. Nous savons que, lorsque l'entité d'une couche (N) désire adresser la primitive *CONNECT.request* à la couche inférieure, elle doit créer l'instance d'un (N-1)SAP pour que la primitive en question puisse passer. Par ailleurs, il faut savoir aussi que, dans le cas de communication avec connexion, les instances de SAP n'existent pas avant la demande de connexion. Ils ne se forment qu'au moment de la connexion et cessent d'exister dès la fin de la communication. En période de déconnexion, ils n'existent plus.

Dans le terminal distant B, l'entité de la couche (N-1) reçoit un PDU qu'elle doit remettre à l'entité de la couche (N). Pendant cette communication, comme l'entité constate qu'il n'y a pas de SAP, elle comprend que la connexion est nouvelle et elle crée alors un (N-1)SAP pour remettre la primitive *CONNECT.indication* à l'entité de la couche (N). La figure 2.16 illustre bien l'établissement de la connexion et l'implication des primitives.

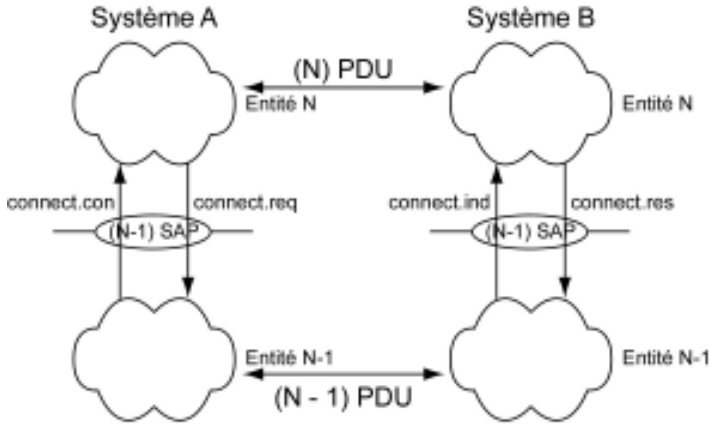


Figure 2.16 Primitives de connexion

Une fois la connexion établie, l'une ou l'autre des entités de la couche N peut transmettre des PDUs de données. Tout comme le montre la figure 2.17, elles ont recours aux primitives *DATA.request* et *DATA.indication*. Toutefois, il est à noter que ni *DATA.response* ni *DATA.confirm* n'existent puisque les primitives de classe *DATA* sont de types non confirmés.

Les primitives peuvent également indiquer dans quel service elles opèrent. Ainsi, si le service est N, les primitives prennent le préfix N, par exemple, *N-CONNECT.request, N-DATA.indication*.

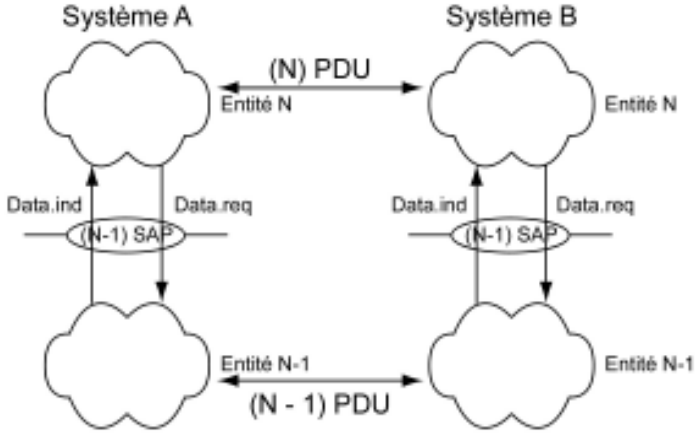


Figure 2.17 Primitives de données

Enfin, on a recours aux primitives de déconnexion lorsque l'une ou l'autre des entités de la couche (N) décide de demander la déconnexion tout comme l'indique la figure 2.18.

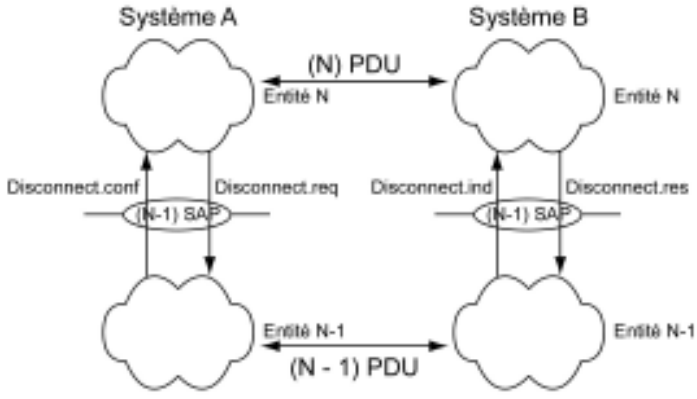


Figure 2.18 Primitives de déconnexion

Pour chacun des services entre deux couches adjacentes, il est possible d'avoir plus d'un SAP, chacun étant exploité par une connexion bien précise et exclusive.

Dans le cas d'une communication sans connexion, les SAPs existent toujours et les primitives utilisées pour transférer les données sont : *UNITDATA.request* et *UNITDATA.indication*.

Il importe de mentionner qu'il est possible que le protocole d'une couche et les services qui lui sont reliés soient tous deux orientés connexion alors que le protocole de la couche sous-jacente de même que les services afférents pourraient ne pas l'être.

Aussi, il est tout à fait normal de constater qu'une couche établit, dans sa phase initiale, une connexion tandis que la couche sous-jacente se trouve déjà à l'étape de l'acheminement des données. Nous pouvons observer ce fait à la figure 2.19 avec les couches réseau et liaison qui dans cet exemple sont tous deux orientés connexion.

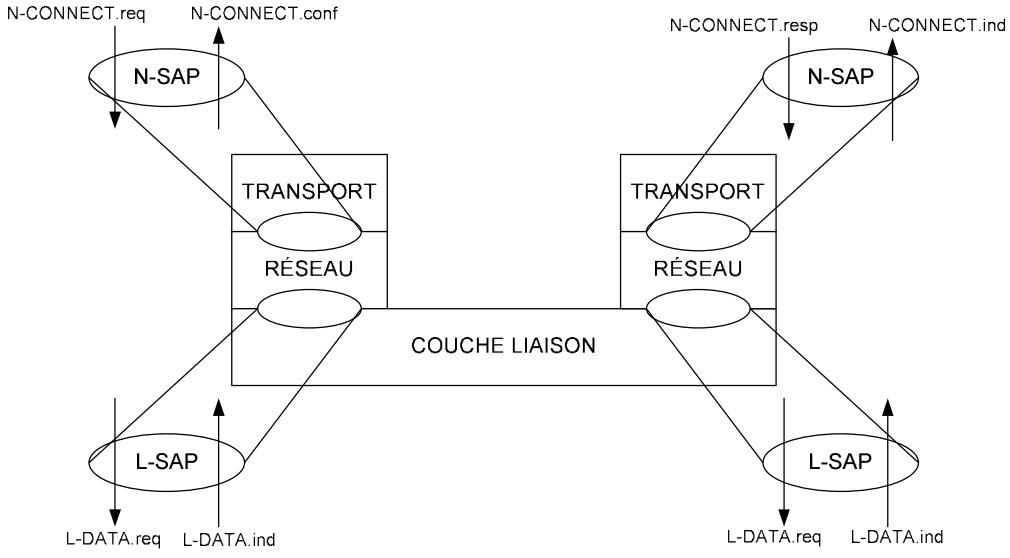


Figure 2.19 Différentes interactions des couches adjacentes

2.11 Les circuits virtuels

Le modèle de référence OSI a été essentiellement pensé et élaboré pour les réseaux de données du type de commutation par paquets. L'information est groupée par « paquet », encapsulée et acheminée au travers du réseau.

Le chemin assigné aux paquets incombe aux commutateurs du réseau et est tracé par eux. Nous avons déjà vu que, dans le cas d'une communication sans connexion, les commutateurs doivent déterminer quel sera le plus court des trajets pour les paquets. Dans le cas de la communication sans connexion, il n'y a aucun chemin commun. Par contre, dans le cas d'une communication avec connexion, les commutateurs ne calculent le trajet le plus court qu'à l'établissement de l'appel. Ce trajet ne vaut donc que tant que l'appel dure et il est connu sous l'appellation de circuit virtuel. Ce circuit n'existe que si la communication est orientée connexion. Par ailleurs, comme nous traitons de chemin et que le routage qui l'établit est à la couche réseau, nous pouvons affirmer ici que le circuit virtuel n'a de sens qu'à la couche 3.

Dans le cas de la commutation de circuits et contrairement au circuit virtuel, le circuit physique est entièrement dédié à l'appel pendant toute sa durée. Pour sa part, le circuit virtuel n'existe que dans le cas de la commutation par paquets. En

outre, ce circuit virtuel partage l'exploitation de liens avec d'autres circuits virtuels.

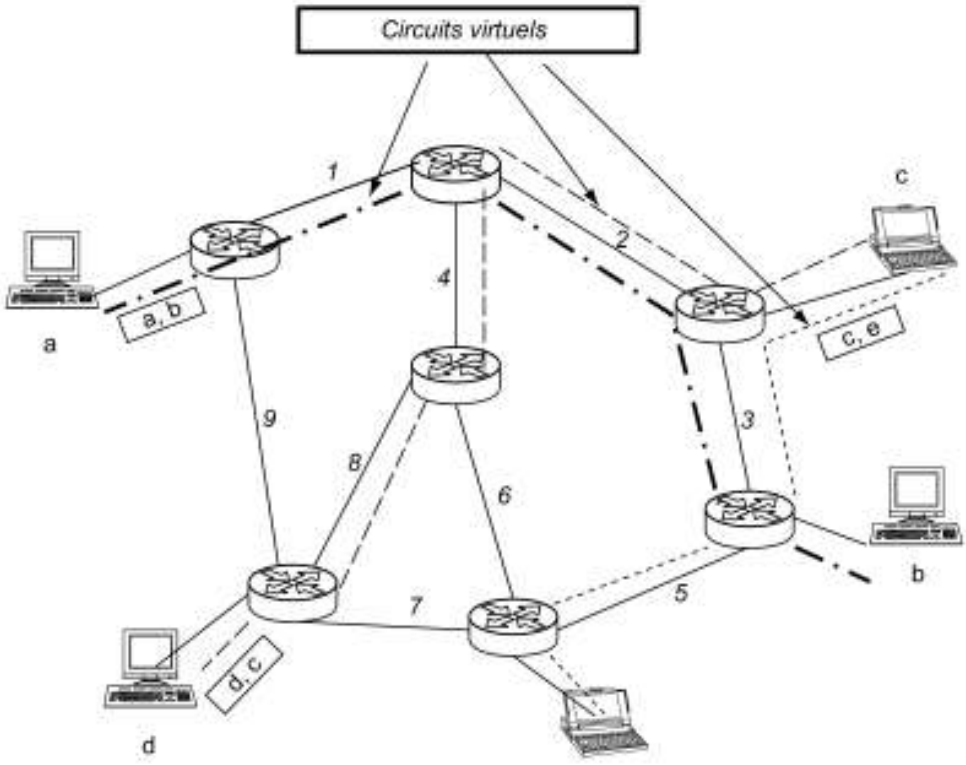


Figure 2.20 Circuits virtuels

Dans l'exemple illustré à la figure 2.20, nous pouvons noter que les terminaux « a » et « b » communiquent par le biais d'un circuit virtuel. Ce circuit virtuel emprunte les liens 1, 2 et 3. Le circuit virtuel entre les terminaux « c » et « d » passe par les liens 2, 4, et 8. Enfin, le circuit virtuel entre les terminaux « c » et « e » utilise les liens 3 et 5. De la sorte, on constate que les liens 2 et 3 sont partagés par chacun de ces circuits virtuels.

2.12 Modèle de référence Internet

Le modèle de référence Internet diffère de celui mis de l'avant et élaboré par OSI. En ce qui a trait à Internet, son modèle est également connu sous le nom de Modèle DoD, car il est relié à ARPAnet lui-même conçu et mis au point pour le DoD (*Department of Defense*). On lui prête également le nom de suite des protocoles TCP/IP.

Le modèle de référence Internet comporte les cinq couches suivantes :

- La couche physique
- La couche liaison
- La couche réseau ou IP
- La couche transport
- La couche application

Ces cinq couches fonctionnent généralement sur le même principe que dans le modèle de référence OSI. Pour leur part, les couches physique et liaison peuvent avoir recours tant à l'accès des réseaux locaux qu'à l'accès point à point. En ce qui a trait à la couche réseau, elle a principalement recours au protocole IP (*Internet Protocol*) et à ses autres protocoles. Quant à la couche transport, elle peut faire appel au protocole TCP ou au protocole UDP, selon les besoins et les cas. Enfin, pour ce qui est de la couche application, on peut dire qu'elle est assez diversifiée et qu'elle peut faire appel à n'importe lequel des protocoles d'application conçus pour Internet.

Le modèle de référence Internet a sa propre terminologie de spécialité. Par exemple, l'interface qui se trouve entre la couche transport et la couche application se sert d'interfaces de connexion (*sockets*) alors que, dans le modèle de référence OSI, on utilise des SAP.

Dans les faits, la couche application du modèle Internet inclut toutes les fonctionnalités des couches session, présentation et application du modèle OSI.

La plupart des systèmes d'exploitation comme Windows, Apple, Unix et Linux viennent avec des logiciels dans le *kernel* afin de pouvoir répondre à chacune des couches du modèle Internet.

2.13 Exercices

2.1

Le modèle de référence OSI comporte sept couches. Quelle relation prévaut entre les couches adjacentes?

1. Chacune des couches doit communiquer à travers un protocole avec les couches adjacentes pour la transmission des données.
2. Chacune des couches ne communique par protocole qu'avec le SAP.
3. Chacune des couches ne communique par protocole qu'avec la couche inférieure.
4. Chacune des couches ne communique pas par protocole avec les couches adjacentes, mais interagit avec elles.
5. Chacune des couches interroge ses couches adjacentes pour savoir si elles ont quelque chose à lui remettre.

2.2

Lorsqu'on dit que les couches doivent être homogènes, on veut signifier que :

1 Les couches sont programmées en se fondant sur une approche commune.
2 Les couches sont programmées à partir d'un même langage.
3 Chacune des couches comprend plusieurs fonctions complètes.
4 Chacune des couches comprend une partie d'une même fonction.
5 Chacune des couches peut avoir recours aux fonctions d'une couche adjacente.

2.3

Parmi les sept couches, on trouve la couche transport. L'une des particularités de cette couche veut que :

1 La couche transport dépend du type de réseau qui en fait le transport.
2 Elle n'établit jamais la connexion.
3 Elle n'implante que le dialogue entre les deux terminaux situés aux extrémités.
4 La couche transport est utilisée lors du transport des bits.
5 La couche transport tout comme le TCP est une application.

2.4

L'encapsulation est un processus utilisé pour :

1 Ajouter un en-tête au PDU.
2 Ajouter un en-tête au SDU.
3 Mettre tout le PDU dans une entité.
4 Mentionner que le PDU devient un SDU.
5 Ajuster le contenu du PDU.

2.5

L'encapsulation est capitale, car :

1 L'en-tête ajouté se conforme au protocole de la couche en question.
2 Les indications utilisées par les couches adjacentes sont transmises.
3 L'en-tête ne sert qu'aux données qu'il transporte.
4 L'encapsulation n'est utilisée que par le réseau.
5 L'en-tête indique l'adresse du serveur ou se trouve les données.

2.6

La relation suivante entre PDU et SDU est juste :

1 Le RPDU remplace le TSDU.
2 Le RPDU est un TPDU.
3 Le TPDU transmis par un système peut-être le même que celui reçu par son homologue.
4 Le PDU généré par la couche physique n'est dans les faits qu'un SDU de la couche physique.
5 La couche application n'établit aucune différence entre un APDU et un ASDU.

2.7

Un élément de réseau doit assumer toutes les fonctions, excepté une :

1 Il doit procéder à l'analyse de l'en-tête du RPDU pour savoir quel chemin prendre.
2 Il doit établir le lien point à point avec les terminaux adjacents.
3 Il doit s'assurer que la couche transport ne contient pas d'erreurs.
4 Il doit assumer la synchronisation sur tous ses liens.
5 Il doit réagir lors de la congestion d'un lien.

2.8

Un SAP est utilisé pour assumer la fonction suivante :

1 Création de PDU.
2 Permettre à une couche d'interagir avec une autre couche adjacente.
3 Procéder à l'encapsulation des données avant de les envoyer à la couche inférieure.
4 Valider la communication.
5 Créer des SDU.

2.9

En référence au modèle OSI, le terme service désigne :

1 La permission d'établir la communication entre entités homologues.
2 L'interface entre deux couches adjacentes.
3 L'information transmise par une couche.
4 L'offre que fait une couche à une autre couche de même niveau.
5 Les résultats de la communication.

2.10

Un protocole est utilisé pour assumer toutes les fonctions qui suivent à l'exception d'une seule :

1 Établir une communication avec la même couche d'un système distant.
2 Déchiffre l'en-tête des PDUs qu'il reçoit.
3 Agir en conformité avec le protocole du terminal de la couche à laquelle il appartient.
4 Savoir que les SDUs reçues par la couche doivent être passées à la couche adjacente supérieure.
5 Transmettre des SDUs directement entre systèmes homologues.

2.11

L'entité d'une couche a les caractéristiques suivantes, sauf une :

1 Dans une communication avec connexion, elle n'a cours que pendant la durée de la connexion.
2 Elle remet son PDU au SAP de la couche inférieure.
3 Elle remet son SDU au SAP de la couche supérieure.
4 Elle agit conformément au protocole du terminal.
5 Elle exerce un contrôle sur les couches adjacentes.

2.12

Les primitives sont utilisées en vue :

1 D'établir la communication par protocole entre les couches adjacentes.
2 D'enlever l'en-tête des PDU.
3 De permettre les échanges entre les entités des couches adjacentes.
4 De modifier les PDU et d'en faire des SDU d'une couche à l'autre.
5 De modifier les SDU et d'en faire des PDU d'une couche à l'autre.

2.13

La segmentation est la fonctionnalité qui permet :

1 De couper la communication.
2 D'interrompre le temps que dure la communication de la valeur d'un segment.
3 De couper les données en segment du même nombre de bits.
4 D'enlever l'en-tête du PDU.
5 De détruire le SAP en fin de la communication.

2.14

Une communication avec connexion est distincte de celle qui est sans connexion, sauf :

1 Pour le nombre de bits de données de la charge utile transmises.
2 Pour l'établissement de la communication.
3 Pour le chemin emprunté dans le réseau.
4 Pour la déconnexion de la communication.
5 Pour l'envoi en séquence des paquets.

2.15

Le circuit virtuel est un lien :

1 Qui n'existe pas véritablement.
2 Qui est d'une grande qualité.
3 Qui est d'une communication orientée connexion et qui partage des ressources.
4 Que l'on ne peut pas voir.
5 Qui n'établit pas de connexion.

3 La couche liaison pour réseau local et point à point

3.1 Introduction

Tout comme nous l'avons déjà vu, la couche liaison permet l'interconnexion point à point entre des terminaux reliés par le même lien physique. De la sorte, le terminal peut se brancher au commutateur auquel il est directement relié. La couche réseau assume le routage et établit une connexion virtuelle avec d'autres commutateurs de même qu'avec la ou les destinations.

Dans le cas des réseaux locaux, le lien direct est partagé. Chacune des stations émettrices transmet des données vues par toutes les stations du segment. Dans ce cas, la fonction de routage ne s'avère plus indispensable. Par contre, la couche liaison revêt une grande importance.

Ce chapitre exposera les concepts de la couche liaison dans le contexte du réseau local en premier, puis la couche liaison de l'accès point à point utilisé par les modems connectés au réseau Internet. Remarquons dans la figure 3.1 que ces deux protocoles sont importants pour compléter l'accès de terminaux à Internet. Dans le cas cité en exemple nous avons un routeur qui relie un réseau local à un modem DSL. Le modem est relié à Internet par le biais de son fournisseur de service par un lien point à point.

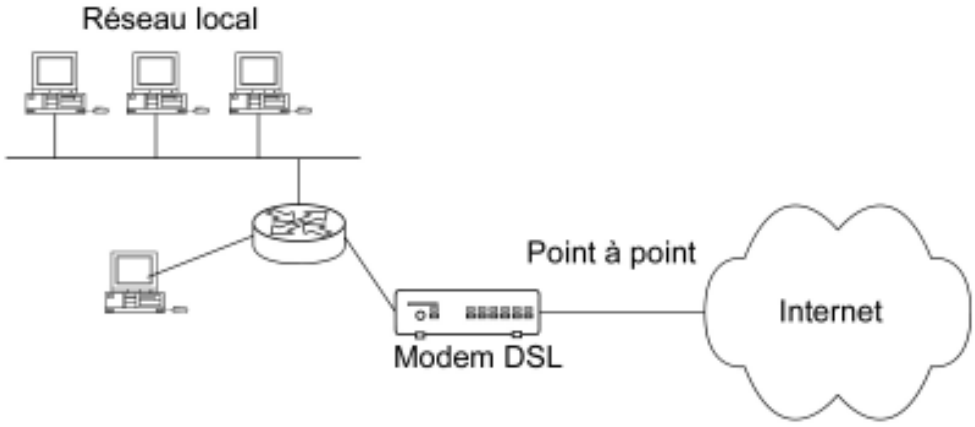


Figure 3.1 Accès réseau local et point à point

Dans les LAN, la couche liaison comprend des fonctions communes à tous les types de réseaux locaux (Ethernet, Anneau à jeton, Bus à jeton, FDDI). Toutefois, les autres fonctions de la couche 2 sont très spécifiques au réseau utilisé. Pour cette raison, le groupe d'étude du projet IEEE 802 a défini la couche liaison avec deux sous-couches : la sous-couche LLC (*Logical Link Control*/Protocole de commande de liaison logique) et la sous-couche MAC (*Medium Access Control*/Contrôle d'accès au support). La figure 3.2 illustre la relation qui prévaut entre le modèle IEEE 802 et le modèle OSI.

Le protocole LLC est la sous-couche commune à tous les réseaux locaux normalisés par le IEEE. Dans un premier temps, nous procéderons à la description détaillée de ce protocole localisé dans la partie supérieure de la couche liaison. La sous-couche MAC qui se trouve dans la partie inférieure de la couche liaison sera décrite plus loin. Il importe de souligner à nouveau que, selon le type de réseau local, le protocole MAC va différer.

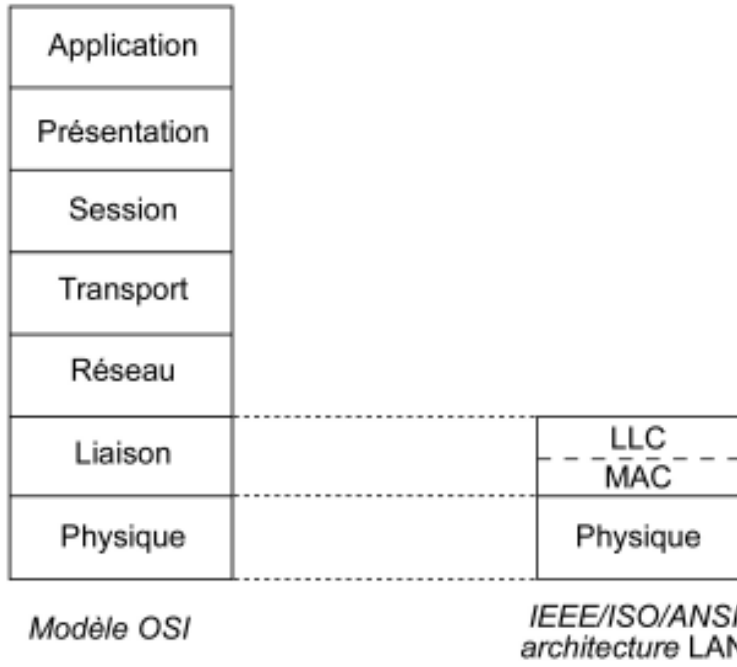


Figure 3.2 Le modèle OSI et le modèle IEEE 802

La normalisation des protocoles et des services des réseaux locaux a été entreprise par le IEEE. Ce dernier a confié un mandat à un groupe d'experts dans le cadre du projet nommé le projet 802. Au nombre de certaines normes développées par ce groupe, on y trouve les suivantes :

IEEE802.2	LLC
IEEE802.3	MAC et couche physique du CSMA/CD
IEEE802.4	MAC et couche physique du bus à jeton
IEEE802.5	MAC et couche physique du réseau à jeton
IEEE802.6	MAC et couche physique du DQDB
IEEE802.11	MAC et couche physique du WLAN

La figure 3.3 donne une vue d'ensemble du rapport qui prévaut entre le protocole LLC et les protocoles MAC et les couches physiques de chacun des réseaux locaux.

Il convient de souligner que la normalisation des réseaux locaux a été aussi réalisée par d'autres organismes tels ANSI (*American National Standard Institute*) et ISO (*International Standard Organisation*). ANSI a conçu et établi la norme pour le protocole MAC et pour la couche physique du FDDI.

Le réseau DQDB (*Distributed Queue Dual Bus*) a été normalisé en vue de définir le protocole et les services d'un réseau métropolitain MAN. Avant la création du DQDB, le FDDI avait préséance à titre de protocole pour le réseau métropolitain IEEE802.6. Une proposition mise de l'avant par l'Australie offrit un nouveau protocole plus efficace et plus robuste. Ainsi, le DQDB fut adopté pour remplacer le FDDI préliminaire.

Les compagnies qui avaient investi et opté pour l'élaboration du FDDI n'étaient pas disposées à envisager un nouveau protocole, car elles considéraient que le FDDI avait un meilleur avenir que le DQDB et que ce dernier n'avait pas encore fait ses preuves. L'ANSI a repris les travaux du FDDI et les a complétés pour aboutir à la norme telle qu'on la connaît aujourd'hui.

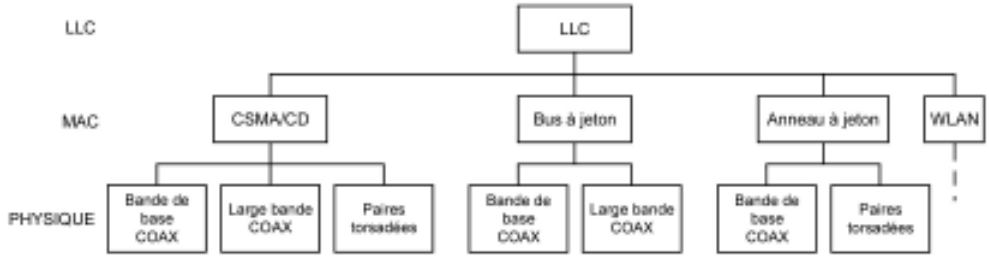


Figure 3.3
Rapports du LLC avec les différents MAC et les couches physiques

3.2 Les services du LLC

Les services du LLC sont en fait identiques à ceux décrits dans le modèle OSI pour la couche 2. Le LLC est basé sur le HDLC (*High-level Data Link Control*/Procédure de commande à haut niveau) qui définit toutes les fonctions possibles de la couche 2. Le LLC est un sous-ensemble du HDLC, n'ayant pas besoin de toutes les fonctionnalités de ce dernier.

À l'instar du modèle de référence générique conçu par l'OSI, le service LLC a recours à la couche supérieure. De manière précise, nous dirons que l'entité de la couche supérieure fait une demande de service en faisant appel au SAP, à savoir : le LLC-SAP. La couche supérieure, qui s'avère être dans ce cas la couche réseau, devient le demandeur de service de la couche LLC. Pour sa part, la couche LLC se trouve être le fournisseur de services pour la couche réseau.

La figure 3.4 illustre bien l'implication des couches LLC et réseau en regard d'échange de services.

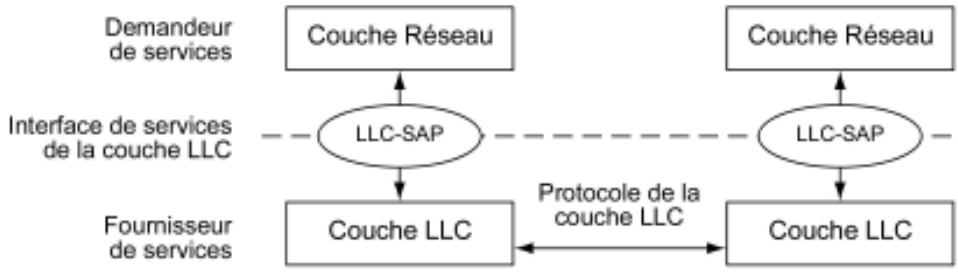


Figure 3.4 Représentation type des rapports prévalant entre les couches LLC et réseau

En général, il existe deux types de services liaison pour les LAN :

- Transmission sans connexion (*Connectionless service*)
- Transmission avec connexion (*Connection-oriented service*)

Le service de transmission sans connexion ne comprend qu'une seule phase, soit : la transmission des données. Il n'y a ni établissement d'appel ni demande de déconnexion. Les propriétés de ce service sont les suivantes :

- Surcharge minimale (*Minimum overhead*).
- Aucune association n'est établie pour procéder à la transmission des données.
- Chaque LLC-PDU est indépendant l'un par rapport à l'autre.
- Aucune vérification de séquence n'est requise. En effet, cette opération permettrait d'assurer que la réception se fasse dans le même ordre que la transmission, ce qui n'est malheureusement pas le cas.
- Aucun accusé de réception n'est émis.
- Le contrôle du flux ou du recouvrement des erreurs est inexistant.
- Pourrait être utile pour le *Multicasting* ou multidistribution.

Le service de transmission avec connexion par contre comprend trois phases :

- L'établissement d'une connexion.
- La transmission des données.
- La déconnexion.

Les propriétés du service avec connexion sont les suivantes :

- L'envoi du LLC-PDU est assuré.
- La connexion LLC est requise avant toute transmission des données.

- La connexion LLC dure pendant tout le temps que les données sont transmises.
- La connexion LLC cesse d'avoir cours lorsque la transmission des données n'est plus requise.
- La séquence LLC-PDU est validée et assurée.
- Un message d'accusé de réception est émis pour les données.
- La communication ne se fait que point à point.

L'ensemble des services est assuré par des primitives qui assument le transport des LLC-SDU émanant de la couche réseau vers la couche LLC adjacente et vice versa, en passant par un LLC-SAP.

3.2.1 Service non confirmé

Certains services n'ont pas de message d'accusé de réception tout comme dans le cas des primitives DATA. Une fois la transmission des données faite, ces services sont qualifiés de services non confirmés. Ceci est illustré dans la figure 3.5.

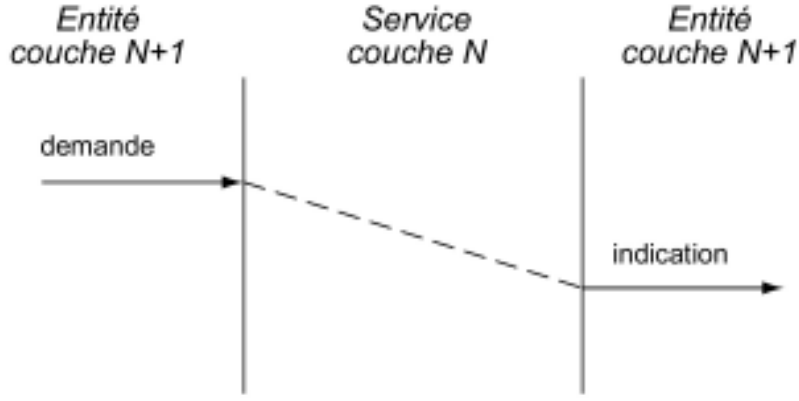


Figure 3.5 Service non confirmé

3.2.2 Service confirmé

Les services confirmés, illustrés dans la figure 3.6, font appel à des primitives de réponse et de confirmation de même que des primitives de demande et d'indication. Ceux-ci sont essentiels pour établir un appel. La transmission de données ne peut avoir lieu s'il n'y a pas de connexion. Si celle-ci n'est pas confirmée, cela signifie que le récepteur pourrait encore être en établissement d'appel alors que le transmetteur se trouverait pour sa part en mode de

transmission de données. Cette situation est évitée grâce au service de connexion confirmé.

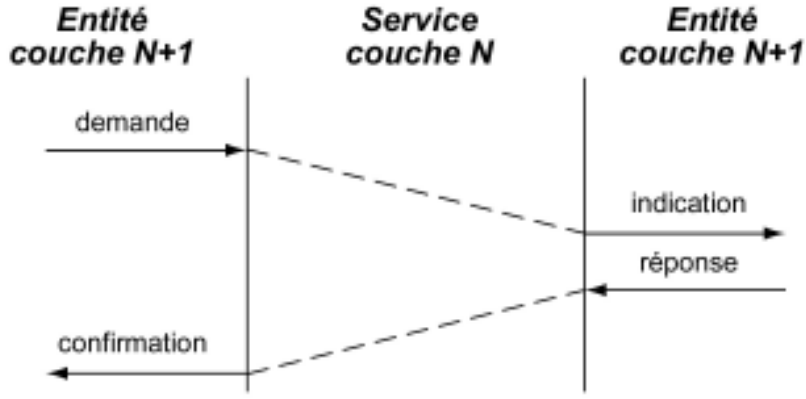


Figure 3.6 Service confirmé

3.2.3 Primitives de services sans connexion

Dans le cas du service LLC non orienté connexion, les primitives auxquelles l'on fait appel sont les *DL_UNITDATA.request* et *DL_UNITDATA.indication* illustrées dans la figure 3.7.

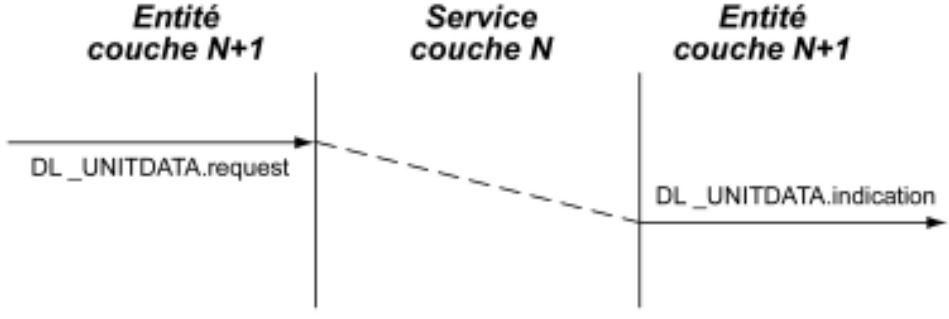


Figure 3.7 Représentation type des primitives d'un service sans connexion

3.2.4 Paramètres pour primitives de services sans connexion

Les primitives de services ont des paramètres qu'elles échangent avec la couche adjacente. Dans le cas des primitives de services sans connexion, les paramètres sont les suivants :

- *Source_address*/Adresse source du LSAP
- *Destination_address*/Adresse de destination du LSAP
- *Data*/Données (Il s'agit d'un LSDU qui doit être acheminé vers un système distant)
- *Priority*/Priorité de la communication

Les primitives et leurs paramètres génériques prennent la forme suivante :

DL_UNITDATA.request	*(source_address, destination_address, data, priority)*
DL_UNITDATA.indication	*(source_address, destination_address, data, priority)*

Ces primitives sont non confirmées.

3.2.5 Primitives de services permettant d'établir un appel de service avec connexion

Les primitives auxquelles on a recours lors du service avec connexion sont établies selon les règles mentionnées au chapitre 2. En premier lieu, on doit faire appel aux primitives de connexion. Ces primitives reçoivent un avis de confirmation tel qu'indiqué à la figure 3.8. Il s'agit des primitives *DL_CONNECT.request, DL_CONNECT.indication, DL_CONNECT.response* et *DL_CONNECT.confirmation*.

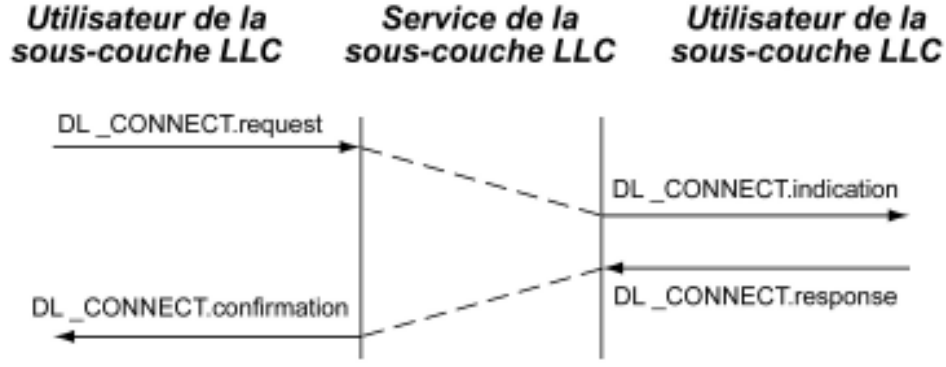


Figure 3.8 Primitives permettant d'établir un appel de service avec connexion

3.2.6 Paramètres des primitives de services d'appel

Les paramètres utilisés dans les primitives de services ne comprennent que les adresses source LSAP et destination LSAP et le degré de priorité. Ces primitives sont les suivantes :

DL_CONNECT.req (source_address, destination_address, priority)
DL_CONNECT.ind (source_address, destination_address, priority)
DL_CONNECT.resp (source_address, destination_address, priority)
DL_CONNECT.conf (source_address, destination_address, priority)

3.2.7 Primitives permettant de procéder au transfert des données de service avec connexion

Les primitives servant à procéder au transfert des données ne sont utilisées qu'une fois l'appel établi. Ces primitives non confirmées sont du type *DL_DATA.request et DL_DATA.indication* (cf. figure 3.9). Leur utilisation peut se faire peu importe le sens.

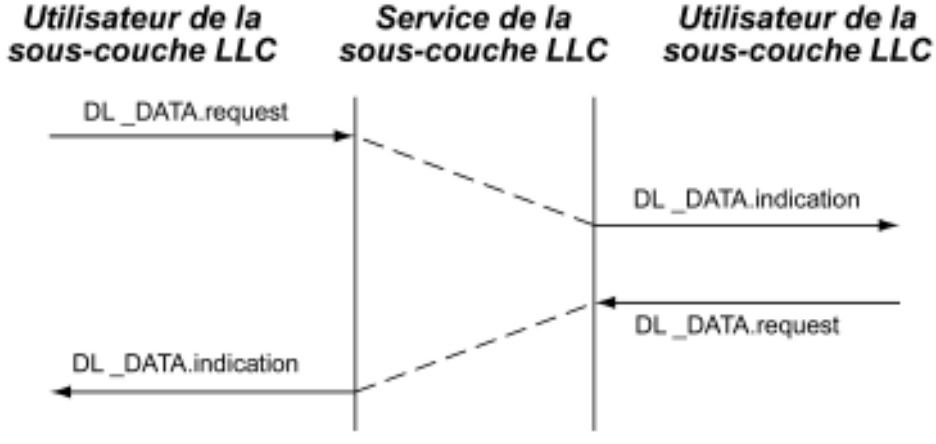


Figure 3.9 Primitives de données de service avec connexion

3.2.8 Paramètres des primitives de données

Les paramètres auxquels on a recours dans le cas des primitives servant à procéder au transfert des données sont les suivants : les données (LSDU) et la priorité.

DL_DATA.req (data, priority)
DL_DATA.ind (data, priority)

3.2.9 Primitives de déconnexion

Lorsque l'appel cesse et que les données ont été transmises, l'un ou l'autre des terminaux peut mettre fin à l'appel. Dans ce cas, les primitives utilisées sont illustrées à la figure 3.10 et il n'y a pas d'avis de confirmation. Il s'agit des deux primitives suivantes :

DL_DISCONNECT.request
DL_DISCONNECT.indication

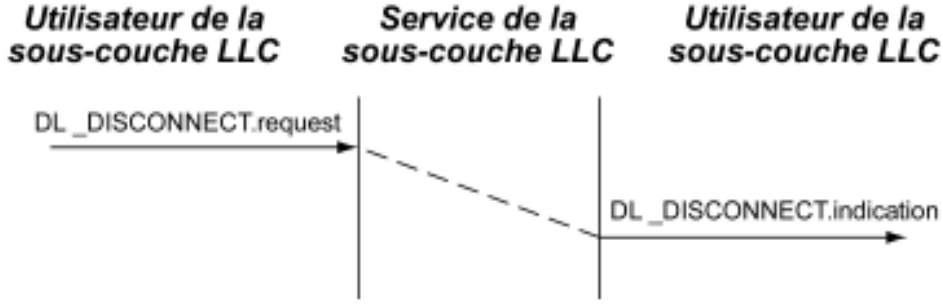


Figure 3.10
Représentation type des primitives de déconnexion d'un service avec connexion

3.2.10 Paramètres des primitives de déconnexion

Les paramètres requis dans le cas des primitives de déconnexion ne comportent que les adresses de la source et de la destination du LSAP et de la priorité. Ces primitives et leurs paramètres sont les suivantes :

DL_DISCONNECT.req *(source_address, destination_address, priority)*
DL_DISCONNECT.ind *(source_address, destination_address, priority)*

3.3 LLC : caractéristiques de protocole

Tout comme nous en avons déjà fait mention, les PDU échangés dans une communication entre deux terminaux distants au niveau du LLC sont régis par un protocole fondé sur celui dénommé HDLC (*High-level Data Link Control*). Le protocole LLC n'est qu'un sous-ensemble du HDLC.

Le protocole LLC permet aux entités de la couche liaison d'établir une connexion et de s'envoyer des données. Le protocole LLC permet aussi d'établir la communication sans connexion. Nous allons décrire le protocole LLC en

commençant par la description des PDU et, en suivant, par celle de la procédure qui est, dans les faits, la définition même du diagramme d'état du protocole LLC.

3.4 Format du LLC PDU

Le LLC PDU est transmis en mode asynchrone. Par contre, le PDU du LLC est toujours encapsulé par une trame MAC. La synchronisation de celle-ci se produit au niveau de la sous-couche MAC. De la sorte, il est possible d'en extraire directement la trame LLC sans avoir recours au fanion[1] que l'on retrouve fréquemment dans le protocole HDLC.

Le premier champ de la trame LLC est le champ adresse. Il comprend les adresses de la couche liaison de destination et de source DSAP et SSAP. Le champ suivant est le champ contrôle ; il indique de quelle nature est le PDU. Les champs adresse et contrôle se retrouvent dans toutes les trames LLC.

Contrairement aux trames du protocole HDLC, le LLC ne comportent pas le champ CRC (*Cyclic Redundancy Check*/Code cyclique de redondance), car ce dernier fait déjà partie de la sous-couche MAC qui encapsule la trame LLC. Ce CRC permet, par son intervention, de détecter les erreurs dans toute trame incluant le LLC.

Le champ information est inclus lors de la transmission des données. Il n'existe que pour certaines catégories de PDU. Il est bon de souligner ici que les PDU doivent leur spécificité au code champ de contrôle. Lors de la transmission de données, le champ information suit le champ contrôle. La figure 3.11 illustre le format générique de la trame LLC.

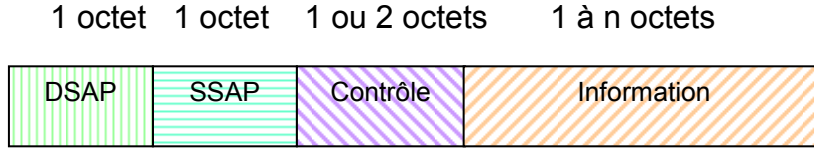


Figure 3.11 Représentation type de la trame LLC

[1] Le fanion est un octet codé de la façon suivante : 01111110 et permet la synchronisation de la trame pour les protocoles HDLC en général.

3.4.1 Absence de transparence dans les trames LLC

Contrairement à la trame HDLC, la trame LLC n'est pas transparente. Cette transparence n'est utile que dans la couche liaison lorsqu'il y a des fanions. Dans ce cas, il y a un risque que le contenu de la trame puisse parvenir dans n'importe quelle combinaison de code y compris celui qui ressemble au code du fanion. Dans le cas du HDLC, un mécanisme se déclenche et permet de créer la transparence.
Pour sa part, le LLC n'a pas besoin de la transparence puisqu'il n'a pas recours au fanion. Il n'y a donc pas risque de confondre les données susceptibles d'avoir le même code que le fanion.

3.4.2 Le champ adresse

Le champ adresse existe pour indiquer les accès de la couche liaison LLC. Les définitions de chacune de ces adresses, illustrées dans la figure 3.12, sont les suivantes :

Le champ DSAP (*Destination Service Access Point*/Point d'accès de destination des services) a 8 bits. Le premier bit (0) indique si l'adresse est celle d'un seul terminal ou d'un groupe de terminaux. Le bit zéro mis à une valeur binaire de 0 signale que l'adresse est celle d'un terminal seulement. Si la valeur binaire du bit zéro est mise à 1, cela mentionne que l'adresse indiquée en est une d'un groupe de terminaux. Les sept bits suivants, soit 1 à 7, incluent l'adresse DSAP. L'adresse de groupe est utilisée pour une multidistribution. Il importe de souligner que seule l'adresse destination peut être une adresse de groupe.

Le champ SSAP (*Source Service Access Point*/Point d'accès des services) a aussi 8 bits. Le bit zéro définit la trame ou, plus précisément, le LPDU comme trame renfermant une commande (0) ou une réponse (1). Les trames de commandes sont celles transmises pour amorcer une réaction du système distant. Une commande peut être une demande de connexion ou bien le processus de la transmission des données. Par contre, une réponse est représentée par une trame d'accusé de réception. Les 7 bits qui suivent, soit de 1 à 7, comportent l'adresse SSAP.

Certaines adresses sont prédéfinies. Notamment, lorsque le champ DSAP contient huit 1, cela signifie que l'adresse est globale; c'est-à-dire qu'elle représente le groupe qui inclut tous les DSAP actifs et est desservie par le service sous-jacent de la couche MAC.

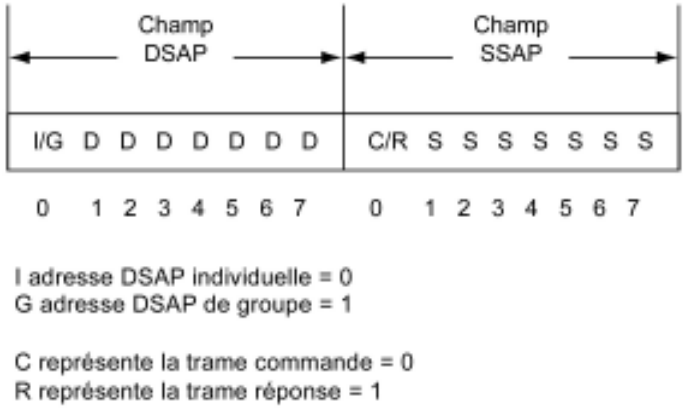


Figure 3.12 Type du format adresse

3.4.3 Le champ contrôle

Tout comme nous en avons fait part précédemment, le champ contrôle indique la nature de la trame. Ce champ peut être codé, suivant le type de trame, sur un ou deux octets. Le premier bit permet d'identifier la trame information, sinon il faut avoir recours aux deux premiers bits pour identifier les trames supervisions et non numérotées. Les trois catégories de trames ou LLC-PDU sont les suivantes :

- INFORMATION
- SUPERVISION
- NON NUMÉROTÉE (*UNNUMBERED*)

3.4.3.1 La trame information

La trame information, illustrée à la figure 3.13, transporte les données une fois l'appel établi. Dans ce cas, la trame information comprend le champ information. Le champ contrôle est codé sur deux octets. Le bit zéro est codé à 0. Ce code figurant au premier bit, le bit zéro, suffit à indiquer qu'il s'agit de la trame information. Une fois décodée en trame information, le reste du champ contrôle doit comprendre les deux compteurs suivants : transmission et réception. Sept bits sont réservés pour chacun d'entre eux, ce qui en limite le décompte de 0 à 127. Ce décompte se fait à chaque incrémentation de 0 à 127 puis il est réinitialisé à 0. De la sorte, on dit que les compteurs sont utilisés Modulo 128. S'ils n'avaient que trois bits, ils seraient utilisés en Modulo 8.

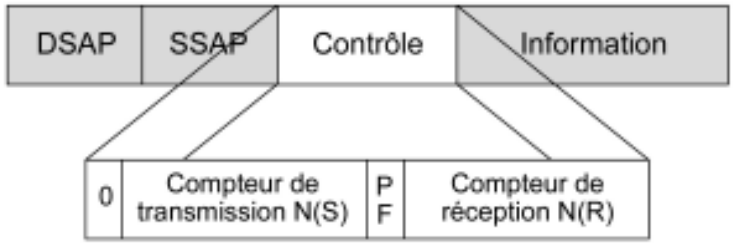


Figure 3.13 Représentation type de la trame information

Le transmetteur comprend un compteur de transmission interne qui est incrémenté aussitôt qu'il y a transmission d'une trame. Le transmetteur copie le compteur de transmission interne à celui de transmission N(S) qui se trouve dans le champ contrôle. Les bits 1 à 7 du premier octet du champ contrôle représentent le compteur N(S). Pour sa part, le récepteur des trames s'assure d'émettre une confirmation de la réception des trames. La procédure qui fait état du protocole et du fonctionnement des compteurs sera décrite en détail dans la section 3.5.

3.4.3.2 Le bit P/F

Le premier bit du second octet du champ information est désigné par P ou F pour *Poll* ou *Final*. Ce bit codé à 1 prend le nom F lorsque la trame qui le contient répond à une trame commande. Il prend la nom P dans le cas d'une trame commande. En d'autres termes, si un terminal désire recevoir une réponse immédiate à sa trame, il met le bit P à 1. Le terminal qui reçoit cet élément d'information met le bit F à 1 sur la trame réponse.

3.4.3.3 La trame supervision

Lorsque le premier bit du champ contrôle est à 1, on n'est pas en présence d'une trame information. Il nous faut voir le deuxième bit pour savoir si la trame en est une de supervision ou s'il s'agit d'une trame non numérotée. Si le deuxième bit est à 0, nous sommes alors en présence d'une trame supervision. Tel que représenté à la figure 3.14, le champ contrôle prend un format différent. Il comporte toujours deux octets. Or, les deux bits suivants, soit le 2 et le 3, nous indiqueront à laquelle des trois trames supervision nous avons affaire. Les bits 4 à 7 ne sont pas utilisés. Le bit 8 est le bit F ou P et les bits 9 à 15 sont utilisés par le compteur de réception.

Le rôle de la trame supervision est de transmettre un accusé de réception des trames, raison pour laquelle le compteur de réception s'y trouve. Celui-ci se compose de 7 bits au même titre que la trame information. Lorsque nous traiterons ultérieurement de la procédure du protocole, le contenu de ce compteur y sera expliqué.

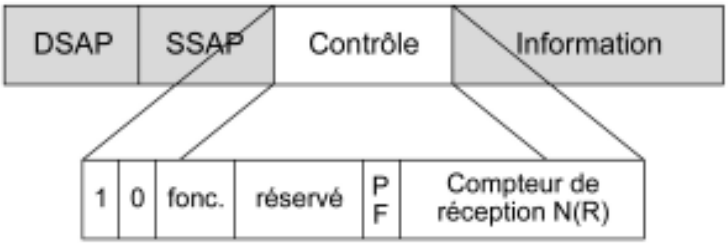


Figure 3.14 Représentation type de la trame supervision

3.4.3.4 Catégories de trames supervision

Il existe trois catégories de trames supervision : RR (*Receive Ready)*, RNR (*Receive Not Ready)* et REJ *(Reject)*. Les formats des champs contrôles de ces trames sont illustrés à la figure 3.15. Toutes les trames supervision sont des trames d'accusé de réception positif y compris la trame REJ tout comme nous le verrons par la suite.

La trame RR est utilisée par le récepteur pour adresser à l'expéditeur un accusé de réception positif de trames information reçues et acceptées sans qu'il n'y ait eu d'erreurs. La trame RR est indiquée dans le champ contrôle par les bits 2 et 3 codés par les valeurs binaires 0 0. Le protocole étant basé selon l'approche Go Back N, l'accusé de réception se fait pour l'ensemble des trames reçues et portant un numéro de transmission plus petit ou égal à celui du compteur de la trame RR. Des exemples seront présentés pour illustrer ce concept.

La trame RNR est identique à la trame RR avec néanmoins une exception : le récepteur demande au transmetteur d'interrompre la transmission des données. En règle générale, le RNR devient nécessaire lorsque la mémoire tampon du récepteur est congestionnée ou éprouve des problèmes de traitement des trames. Le RNR n'est alors rien d'autre qu'un moyen de contrôle de flux. Celui-ci est plutôt strict, car c'est du tout ou rien. Une méthode de contrôle du flux qui permettrait de faire fluctuer le débit lors de la transmission des trames serait favorable. Ceci existe et nous le verrons quand nous traiterons des fenêtres. Au moment où le transmetteur reçoit une trame RNR, il ne reprend sa transmission des trames information qu'après la réception d'une trame RR.

Finalement, dès que la trame information est mal reçue ou égarée, le récepteur émet une trame REJ pour demander de procéder à une retransmission de la trame indiquée dans le compteur de réception. La trame REJ est également une trame d'accusé de réception positif. Tout en rejetant une trame information spécifique, notamment celle spécifiée par le compteur N(R), la trame REJ confirme celles qui la précèdent. Par contre, toutes les trames information suivant la trame rejetée devront aussi être retransmises.

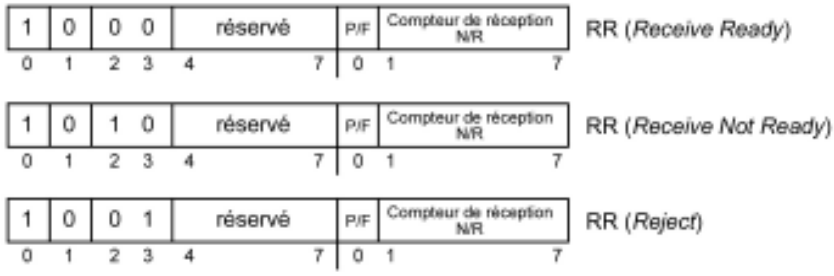


Figure 3.15 Représentation type de la trame supervision

3.4.3.5 Trames non numérotées

Lorsque les deux premiers bits du champ contrôle sont codés à 11 binaire, nous sommes en présence d'une trame non numérotée. Le champ contrôle des trames non numérotées ne comprend qu'un octet. Ses bits 2, 3 ainsi que 5, 6 et 7 sont réservés pour coder les types de trames non numérotées. Le bit 4 est réservé pour la fonction P/F (*Poll/Final)*. La figure 3.16 illustre le format de la trame non numérotée et la figure 3.17 les diverses catégories.

Certaines de ces trames seront utilisées pour l'établissement d'appel comme le SABME, la demande de déconnexion DISC, l'accusé de réception UA de l'établissement d'appel ou de déconnexion, ou le refus DM de la demande de connexion.

Dans le cas d'une communication sans connexion les trames de données UI sont transmises.

La trame XID permet d'obtenir une réponse d'un LLC. Il sera alors possible de savoir si le LLC est en mode connexion ou, du moins, de connaître son numéro de groupe.

Tout comme son nom l'indique, la trame TEST sert à effectuer des tests de LOOPBACK. L'utilisation et la réponse de la trame TEST sont optionnelles.

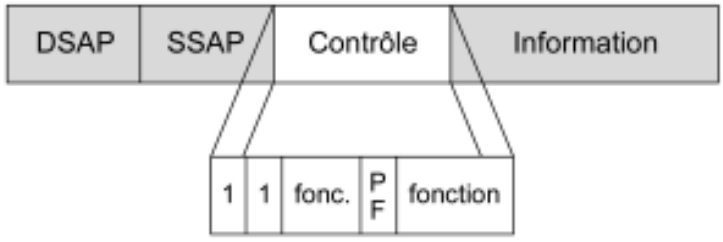


Figure 3.16 Représentation type d'une trame non numérotée

1	1	0 0	P/F	0 0 0	UI (*Unnumbered Information*/Information non numérotée)
1	1	1 1	P/F	1 0 1	XID (*Exchange Information*/Échange d'informations)
1	1	0 0	P/F	1 1 1	TEST (Test)
1	1	1 1	P/F	1 1 0	SABME (*Set Asynchronous Balanced Mode Extended*)
1	1	0 0	P/F	1 1 0	UA (*Unnumbered Acknowledgement*/Accusé de réception non numéroté)
1	1	1 1	P/F	0 0 0	DM (*Disconnect Mode*/Mode déconnexion)
1	1	0 0	P/F	0 1 0	DISC (*Disconnect*/Déconnexion)
1	1	1 0	P/F	0 0 1	FRMR (*Frame Reject*/rejet de trame)

Figure 3.17 Représentations type de trames non numérotées

3.5 Procédure du protocole LLC

3.5.1 Établissement de l'appel

Le LLC est un protocole qui peut être utilisé en mode orienté connexion ou non orienté connexion. Dans le premier de ces cas, le protocole doit procéder à la connexion entre deux terminaux avant de passer à la transmission des données proprement dite. La connexion se fait par la transmission de la trame SABME. L'un ou l'autre des terminaux est en mesure de procéder à la transmission de la trame pour établir l'appel. L'acceptation de la connexion par l'autre terminal est confirmée par la transmission de la trame UA. Le diagramme temps de la figure 3.18 nous montre clairement comment établir un appel réussi.

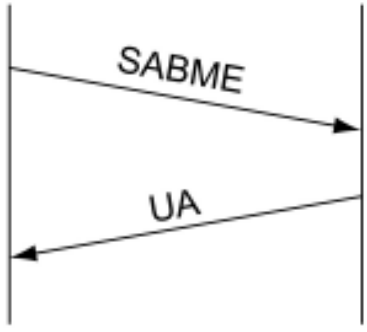


Figure 3.18 Procédure d'établissement d'un appel

Certaines trames sont des trames de commande, alors que d'autres en sont de réponse. Par exemple, la trame SABME est une trame de commande alors que la UA en est une de réponse. Tout côté qui transmet une trame de commande déclenche une minuterie qui permettra de déterminer si la réponse accuse un retard. Dans ce cas-là, la trame de commande devrait être retransmise.

Si l'appel est refusé, la trame DM est transmise à la place de la trame UA. En cas de refus, la transmission des données ne peut pas avoir lieu. Lorsque la trame SABME est transmise, le bit P est à 1. Le bit F de la réponse UA ou DM se trouve également à 1. Si le bit F de la trame réponse UA ou DM est à 0, il y a erreur dans la trame. Une fois la trame UA reçue, l'appel est établi et la transmission des données sous la forme de trames information peut être amorcée.

3.5.2 Déconnexion de l'appel

Lorsque la transmission des données est complétée, l'un ou l'autre des terminaux peut mettre fin à l'appel. La déconnexion se fait par la transmission de la trame commande non numérotée DISC. La déconnexion est confirmée grâce à la trame réponse non numérotée UA. La figure 3.19 illustre l'échange de trames.

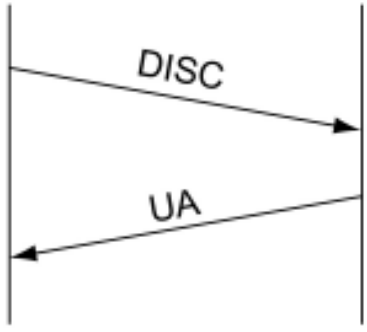


Figure 3.19 Procédure de déconnexion d'un appel

3.5.3 Transmission des données

Chaque entité de la couche liaison qui établit une communication avec son homologue compte deux compteurs internes : un V(S) désignant le compteur de transmission et un V(R) désignant le compteur de réception. Lorsqu'un appel est établi, et avant que la communication n'ait lieu, les compteurs sont mis à zéro de part et d'autre.

Tout comme nous l'avons déjà vu, les trames information comprennent les compteurs N(S) et N(R). Le transmetteur de la trame information copie les compteurs internes V(S) et V(R) dans les compteurs N(S) et N(R) respectivement. Une fois la trame transmise, le transmetteur incrémente son compteur interne de transmission V(S) et la minuterie se déclenche.

Avant toute chose, le récepteur de la trame information s'assure que le compteur N(S) de la trame reçue est en tout point égal à son compteur interne V(R). Dans le cas contraire, il y a détection d'une erreur de transmission. Une fois la trame acceptée, le récepteur incrémente son compteur interne V(R). Le récepteur peut par la suite envoyer une trame d'accusé de réception. La figure 3.20 nous donne une bonne représentation de transmission de la trame information.

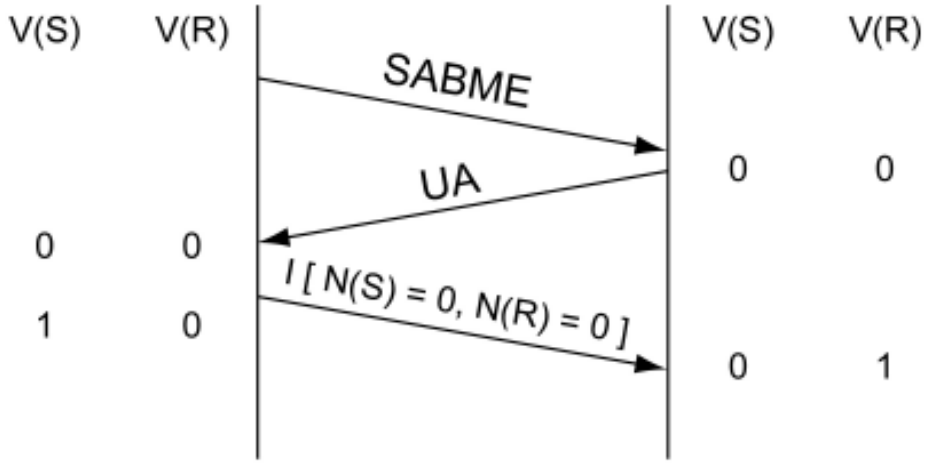


Figure 3.20 Représentation type de la transmission de la première trame

Ainsi, la transmission des données peut se poursuivre aussi bien d'un côté que de l'autre de la connexion. Le fait que cette communication soit orientée connexion, le transfert des données se fera de manière séquentielle. Ce séquencement est assuré. De plus, le lien permet de détecter les erreurs et d'y apporter les rectificatifs. La détection d'erreurs se fait au niveau de la sous-couche MAC à l'aide du CRC. Par contre, la rectification des erreurs est assumée par le protocole LLC avec le concours des trames supervision.

3.5.4 Accusé de réception et rectification des erreurs

Les trames supervision sont utilisées aux fins d'accusé de réception ou de rejet des trames information reçues. Cet accusé de réception se fait selon le protocole ARQ (*Automatic ReQuest*) connu sous le nom de GO BACK N. Toujours selon ce protocole, l'accusé de réception ne se fait pas obligatoirement après la transmission de chaque trame information. Une trame supervision transmise peut représenter un accusé de réception en regard de toutes les trames information reçues. Une seule trame est donc suffisante pour accuser réception de plusieurs trames information. Toutes les trames supervision sont qualifiées d'accusé de réception positif des trames information reçues y compris la trame supervision REJECT. Ce point est élaboré en détail dans ce chapitre.

Aussitôt qu'une trame information est transmise, un compteur d'acquittement se déclenche. Ce compteur est un paramètre de connexion qui défini l'intervalle de temps durant lequel le LLC doit recevoir un accusé de réception pour une trame information transmise. La valeur maximale de ce compteur est égale à T. Si aucun accusé de réception n'a été reçu pour cette trame dans le nombre de

secondes imparties du temps T, la trame non confirmée sera alors retransmise. Un compteur N2 relatif au nombre de retransmissions N est aussi activé lors de la transmission de chacune des trames. Si une trame information est retransmise N fois en raison d'un accusé de réception non reçu ou en raison d'un rejet de la trame, la transmission est interrompue sur le lien et des alarmes signalant qu'il y a une panne se déclenchent.

3.5.4.1 La trame supervision RR

Lorsque plusieurs trames information sont recueillies, une trame supervision RR peut être transmise pour accuser réception de la première trame, de certaines d'entre elles ou bien de toutes les trames reçues. La trame RR renferme dans son compteur N(R) la valeur du compteur V(R) du terminal qui la transmet. Ce compteur représente aussi le compteur N(S) de la dernière trame information reçue que l'on désire accuser réception plus 1. En fait, ce compteur N(R) nous indique le numéro de la trame suivante qui est attendue ou non encore confirmée.

Dans le cas où quatre trames information sont reçues par le système B avec les compteurs N(S) dont les valeurs sont 5, 6, 7, et 8, une trame RR peut être transmise pour accuser la réception de la première trame information seulement, c'est-à-dire la 5. La trame RR mettrait alors son compteur N(R) à 6. S'il y a accusé de réception des trames 5, 6, et 7 mais pas de la trame 8, alors le compteur N(R) de la trame RR se mettra à 8. Ceci ne signifie pas que la trame 8 est pour autant rejetée, mais seulement qu'elle n'a pas encore été confirmée aux fins de réception. La figure 3.21 illustre l'utilisation de la trame RR. Cet exemple nous indique clairement que la trame 8 est acceptée par le système B sans avoir été toutefois confirmée aux fins de réception.

À la réception d'une trame RR, les compteurs V(S) et V(R) ne sont nullement modifiés. Le récepteur de la trame RR qui est le transmetteur des trames information conserve la valeur de ses compteurs inchangée.

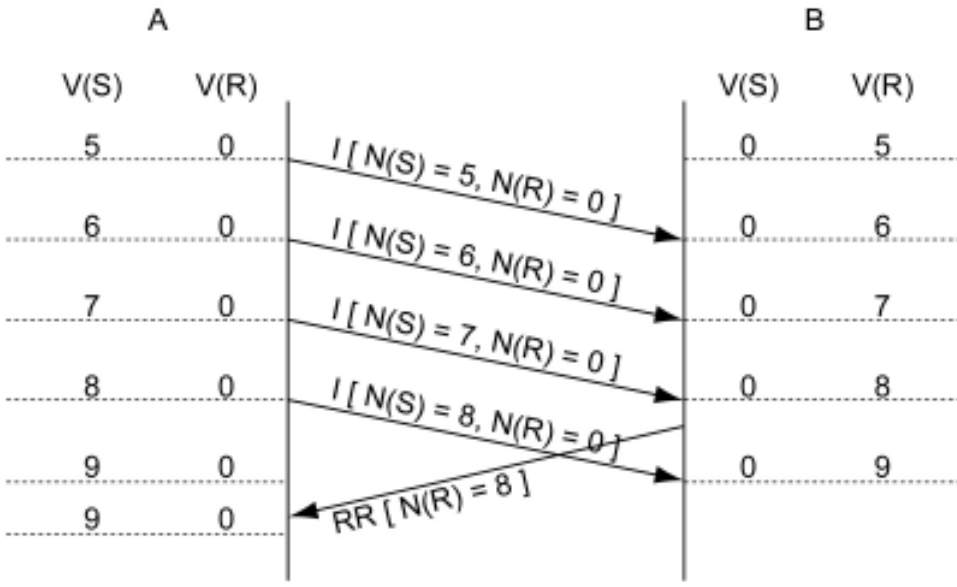


Figure 3.21 Représentation type d'un accusé de réception par la trame RR

3.5.4.2 La trame supervision RNR

La trame supervision RNR est semblable à la trame RR à une exception près : avec la trame RNR, le transmetteur reçoit un avis d'interruption de la transmission de trame de données. La trame RNR confirme également la réception de trames information tout comme dans le cas de la trame RR. Cela nous permet d'exercer un contrôle sur le flux des trames information et d'éviter toute congestion. La transmission de données reprend lorsqu'une trame RR est reçue par le transmetteur. La figure 3.22 nous illustre le fonctionnement de la trame RNR.

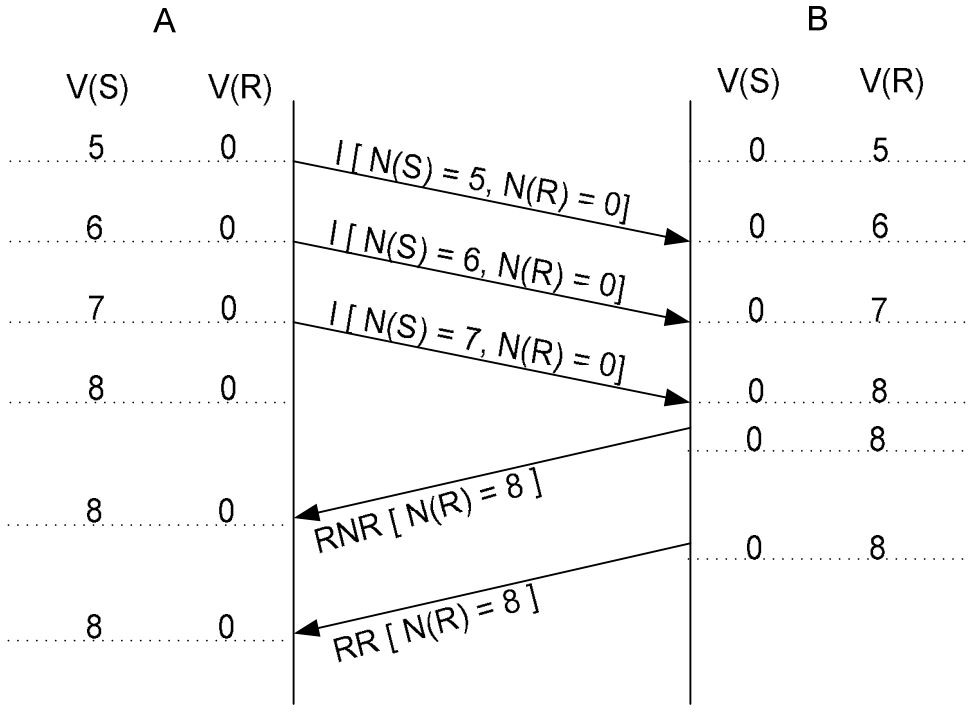


Figure 3.22
Représentation type de l'accusé de réception émis par la trame RNR

Il est possible que le transmetteur ait envoyé une trame information peu de temps avant d'avoir reçu la trame RNR. Dans un tel cas, le récepteur aura déjà transmis la trame RNR avant la réception de la dernière trame information. Si, avant de transmettre la trame RR, le temps d'attente t excède le temps de réponse alloué T pour acquitter la trame information, les trames reçues après la trame RNR ne seront pas confirmées aux fins de réception. Toutefois, si ce temps est en deçà du temps alloué pour une réponse $(t < T)$, alors la trame RR transmise pour redéclencher le processus de transmission servira également d'accusé de réception des trames information reçues après la trame RNR. Nous pouvons remarquer qu'à la figure 3.23, le système B, bien qu'il ait transmis une trame RNR, confirme la réception de la dernière trame transmise par le système A à l'aide de la trame RR, car le temps d'attente t est plus petit que le temps alloué T ($t<T$).

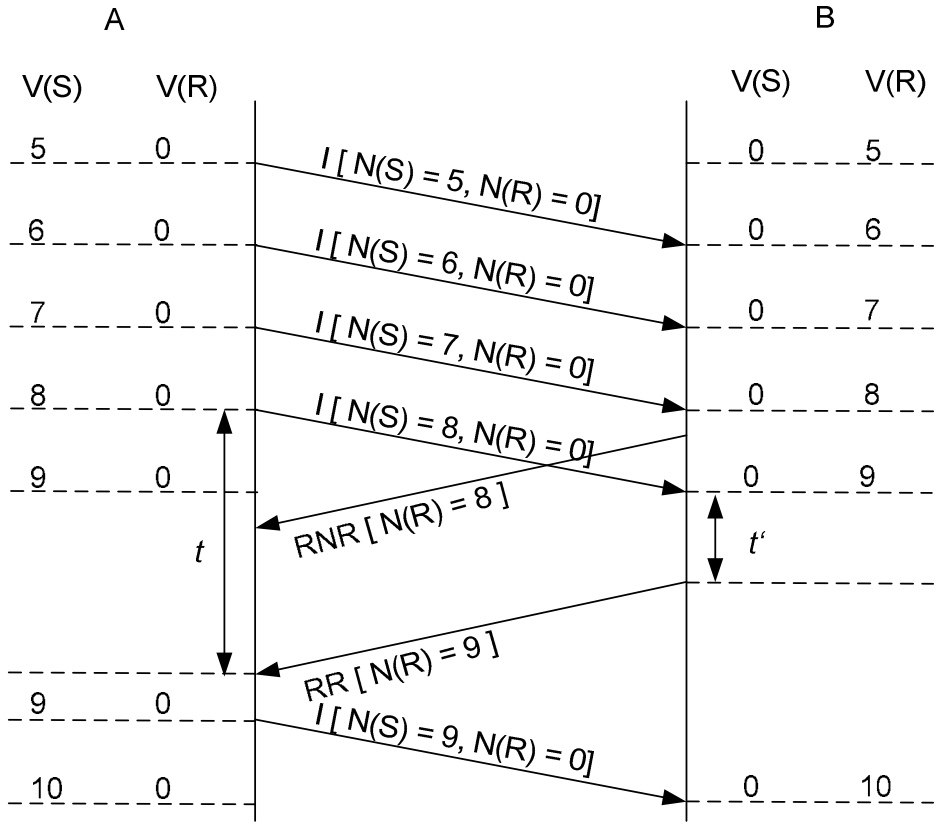


Figure 3.23
Représentation type du processus d'accusé de réception par la trame RNR

3.5.4.3 La trame supervision REJECT

Comme nous l'avons déjà mentionné précédemment, la trame supervision REJECT assume aussi le rôle de trame d'accusé de réception positif. Par contre, cette trame rejette la trame information figurant dans son compteur N(R) de même que toutes celles qui suivent. Cependant, elle accuse réception implicitement de celles dont le compteur est plus petit que la valeur indiquée par N(R). La trame REJECT demande qu'il y ait retransmission des trames information à partir de la valeur indiquée dans son compteur. La trame REJECT a donc rejeté toutes les trames information à partir de celles dont le compteur est égal à son compteur N(R) (figure 3.24).

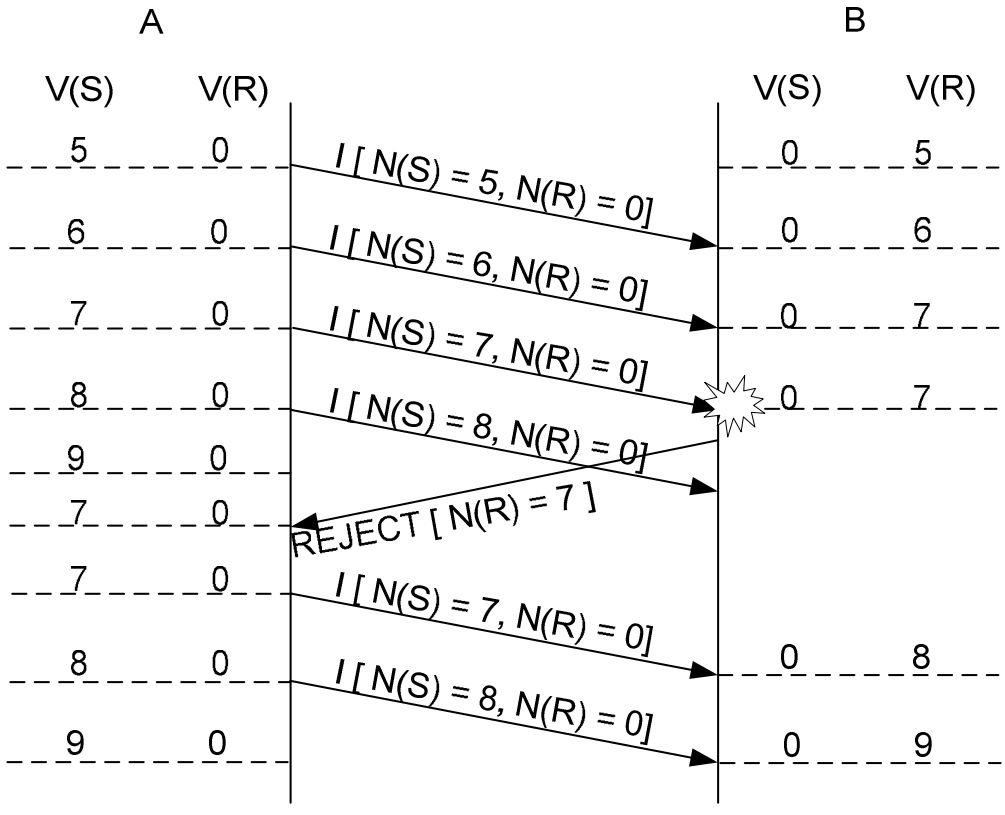


Figure 3.24
Représentation type d'accusé de réception et de retransmission avec REJECT

3.5.4.4 Accusé de réception par la trame information

Nous avons vu que les trames supervision accusent réception des trames information. Toutefois, lorsque le récepteur doit également transmettre des trames information au transmetteur des premières trames non confirmées, il n'a pas besoin d'envoyer en plus des trames supervision. Ses trames information lui servent aussi d'accusé de réception des trames. Notons que les trames information possèdent les deux compteurs N(S) et N(R) et leur utilisation a été pensée en fonction de l'atteinte de cet objectif. Cette méthode est connue sous le nom de *Piggy Backing*. La figure 3.25 illustre ce cas.

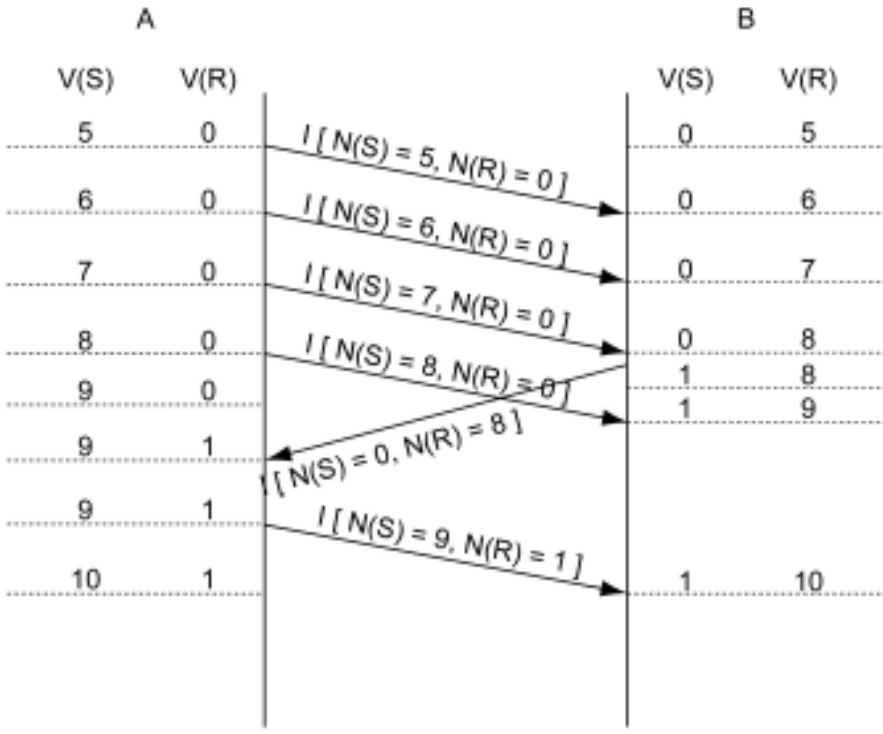


Figure 3.25
Représentation type d'accusé de réception des trames avec la trame information

3.5.5 Compteurs de retransmission

Chaque entité de la couche liaison assure la transmission et réception de trames information. Comme nous l'avons vu précédemment, si une trame est mal reçue, elle devra être retransmise. Le nombre de fois que la trame peut être retransmise dépend du compteur de retransmission N2. Si le compteur est à 4 et qu'une trame particulière est retransmise quatre fois en raison d'erreur de transmission ou de non réception, la ligne est mise hors service.

3.6 Contrôle du flux

Afin de contrôler le flux d'informations lorsqu'il y a congestion de la mémoire tampon ou du circuit, le transmetteur peut avoir recours à la trame RNR.

L'autre méthode pratiquée est celle de la fenêtre. On désigne par ce terme le nombre de trames information qu'il est possible de transmettre successivement sans n'avoir encore reçu leur accusé de réception. Par contre, une fois le nombre

de trames information transmises égale la fenêtre et qu'aucun accusé de réception de ces trames n'est reçu, le transmetteur doit interrompre sa transmission. L'accusé de réception d'au moins une de ces trame information doit être reçu pour pouvoir en transmettre une nouvelle. S'il obtient un accusé de réception pour toutes les trames transmises, il peut alors poursuivre et transmettre le même nombre de nouvelles trames information dans la mesure où il en a autant à transmettre.

Si l'accusé de réception ne parvient pas après un temps prédéterminé *T*, les trames sont dès lors considérées comme égarées et seront retransmises.

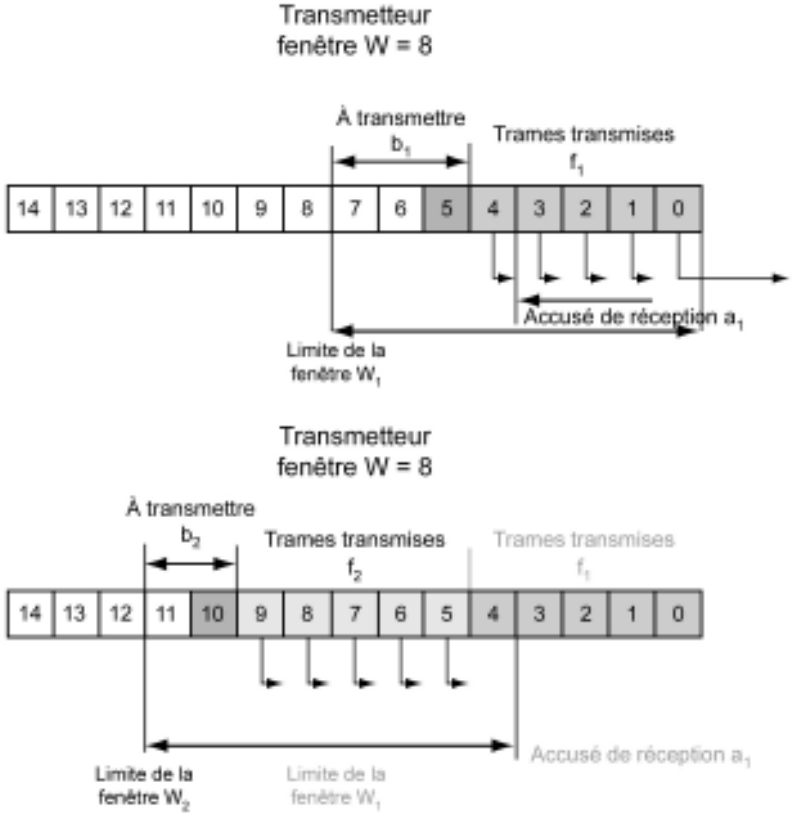


Figure 3.26 Rotation de la fenêtre

La rotation de la fenêtre est basée simplement sur un mécanisme qui permet de poursuivre la transmission des trames selon le nombre précisé par la fenêtre et par les accusés de réception. Ce mécanisme permet de procéder à une transmission continue et de permettre au récepteur d'exercer un contrôle sur le transmetteur en régulant la transmission des accusés de réception des trames.

Dans l'exemple illustré à la figure 3.26, nous remarquons que la fenêtre débute à 8. Cela signifie que l'un ou l'autre des systèmes peut procéder à la transmission de huit trames de manière ininterrompue et sans en attendre leur accusé de réception. Cinq trames sont transmises de la trame 0 à la trame 4 durant un temps f_1. Conformément à la fenêtre de la figure 3.26, trois autres trames peuvent encore être transmises, soit les trames 5, 6 et 7. Une trame RR arrive et accuse réception de la trame 3 ainsi que des trames 0, 1 et 2. Il ne reste plus que la trame 4 qui n'a pas été confirmée comme ayant été reçue. Dès l'accusé de réception de cette dernière, la fenêtre effectue une rotation. Cela revient à dire que, grâce à l'accusé de réception et attendu qu'une trame a été transmise, mais dont la réception n'est pas encore confirmée, sept autres trames peuvent être transmises avant qu'un nouvel accusé de réception ne soit émis. La nouvelle position de la fenêtre se trouve dès lors à 11. À supposer que cinq autres trames (de la 5 à la 9) soient transmises, il ne reste plus que deux autres trames qui peuvent être transmises avant d'interrompre la transmission si l'accusé de réception des trames n'est pas reçu.

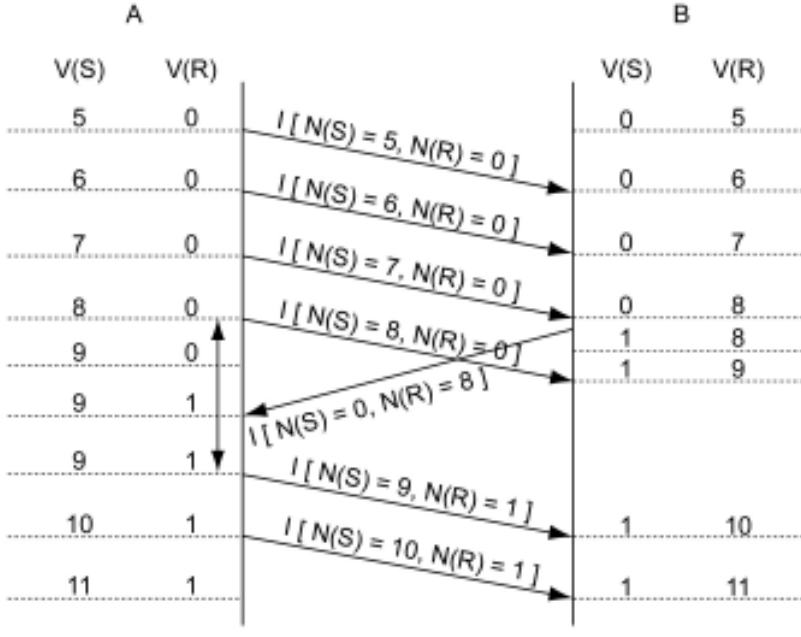


Figure 3.27 Rotation de la fenêtre

L'application de la fenêtre au diagramme temps nous permet de voir l'effet causé d'un accusé de réception parvenant avec retard. Dans l'exemple illustré à la figure 3.27, la fenêtre W est de 4. Le système A transmet quatre trames information de manière ininterrompue. Il doit arrêter la transmission attendu qu'il n'a pas reçu d'accusé de réception. Le système B envoie un accusé de réception pour la

trame 7. De la sorte, il confirme la réception des trames 5, 6 et 7. Le système A reprend alors la transmission dès l'accusé de réception. Cela lui permet de procéder à la transmission de trois autres trames information dans l'éventualité où il en aurait trois autres à transmettre (la figure 3.27 nous montre qu'il n'y a que deux nouvelles trames transmises).

3.7 LLC sans connexion

Lorsque le protocole LLC fonctionne en mode sans connexion, les PDU d'informations sont échangés entre les terminaux sans établir de connexion et sans émettre d'accusé de réception des trames transmises. En effet, dans ce mode, il n'y a pas de contrôle du flux des trames ou de rectification d'erreurs.

La trame UI (*Unnumbered Information*/Information non numérotée), illustrée dans la figure 3.28, est utilisée pour la transmission des données dans le mode sans connexion. Cette trame est transmise par un terminal à un ou plusieurs autres terminaux. Le recours aux trames UI n'a aucun effet sur les compteurs V(S) et V(R). Il n'y a naturellement aucune connexion ni émission d'accusé de réception lors de son utilisation. La trame UI en est une de commande. Si on y a recours comme trame de réponse un message d'erreur parvient au terminal expéditeur. Le bit P ne peut pas être utilisé dans la trame UI puisqu'il n'y a pas de possibilité d'envoyer une réponse à cette trame.

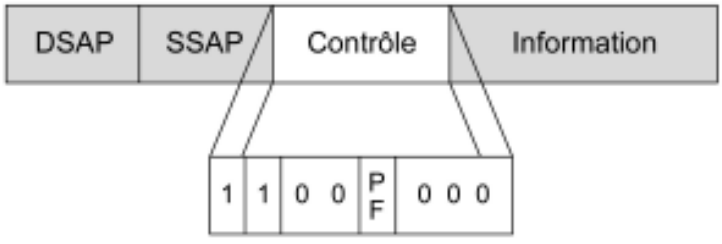


Figure 3.28 Représentation type de la trame UI

3.8 Les protocoles de liaison SLIP et PPP

L'utilisation du protocole LLC est importante dans un environnement de réseaux locaux. Par contre, dans le cas de la connexion point à point, un protocole plus spécifique et plus simple est recommandé, car il prendra en considération les contraintes imposées par les modems. Les protocoles pouvant répondre à de tels

critères sont le SLIP (*Serial Line Internet Protocol*) et, tout particulièrement, le PPP (*Point-to-Point Protocol*/Protocole point à point).

3.8.1 Le protocole SLIP (Serial Line IP)

Un grand nombre d'accès résidentiels n'utilise pas de routeur mais plutôt la ligne téléphonique pour accéder au ISP (*Internet Service Provider*/Fournisseur de services Internet) et de là à Internet. Le protocole de liaison utilisé ne peut pas être un LLC ou un HDLC. L'un des protocoles conçu à ces fins est le SLIP (*Serial Line IP*) qui peut être relativement rapide mais très sensible aux erreurs.

Le groupe de travail IETF (*Internet Engineering Task Force*) a défini le SLIP comme étant non conforme à la norme et il en fait la description dans le RFC 1055.

Pour le fonctionnement du protocole SLIP, il faut appliquer les quelques règles suivantes :

Règles précisant l'encadrement (*framing)* utilisé par le protocole SLIP (cf. figure 3.29) :

1- Chaque datagramme IP inclus dans une trame SLIP se termine par END (0xc0) qui est en soi un caractère particulier. Afin de prévenir les bruits en ligne, ce caractère particulier est aussi transmis en début de trame.
2- Si un octet du IP est identique au caractère END (0xc0), la séquence composée des deux octets 0xdb et 0xdc est transmise à la place du caractère END (0xc0). (0xdb correspond au ESC du protocole SLIP). Si un octet du IP est identique à ESC (0xdb), alors la séquence composée des deux octets 0xdb et 0xdd est transmise à sa place.

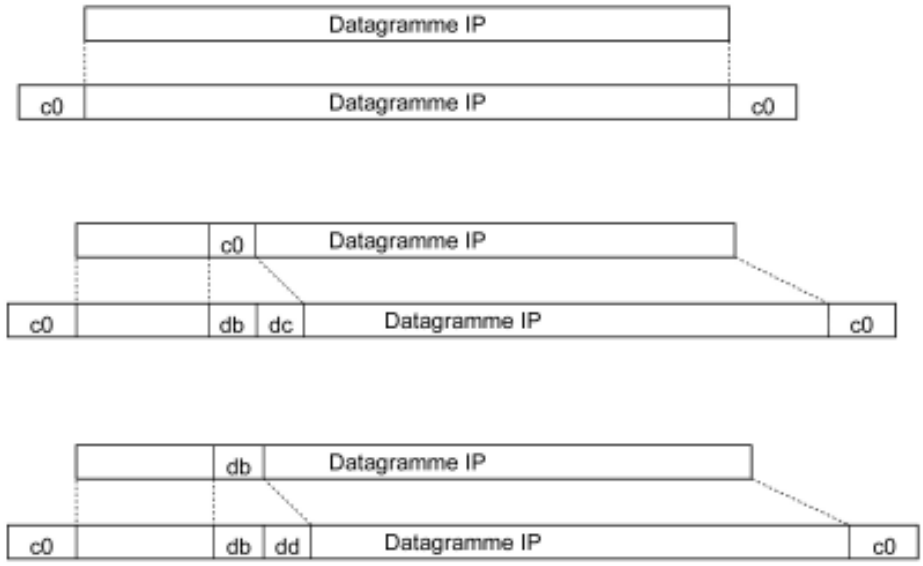


Figure 3.29 Représentation type des trames du protocole SLIP

Tout comme nous l'avons souligné, le protocole SLIP n'est pas des plus fiables. C'est la raison pour laquelle il ne doit être utilisé que lorsque les lignes de transmission (lignes téléphoniques) sont de qualité supérieure et qu'elles ne sont pas sujettes aux bruits. Les lacunes de ce protocole sont :

1- Chacune des deux extrémités doit connaître l'adresse IP.
2- Il n'existe pas de champs pour les autres protocoles de couche réseau. Le protocole SLIP peut seulement être utilisé par le réseau IP (les deux extrémités de la communication ont au préalable établi le recours au protocole SLIP).
3- La somme de contrôle (*checksum*) n'est pas ajoutée au protocole SLIP. Cela signifie qu'il n'y a pas de rectification advenant des cas d'erreurs et il y a risques d'interruption de la communication.

3.8.2 Le protocole PPP (Point-to-Point Protocol/Protocole point à point)

Afin de combler les lacunes du protocole SLIP, un protocole plus efficace a été conçu et mis au point. Il s'agit du protocole PPP qui est nettement plus efficace que le protocole SLIP, car il est encapsulé dans une trame de type HDLC et rectifie toutes les déficiences du protocole SLIP. Le protocole PPP est mentionné dans plusieurs documents RFC (*Request Fort Comment*) dont il sera fait état selon l'application.

Le protocole PPP fonctionne sur toute sorte de supports physiques. Le protocole PPP peut fonctionner sur un lien point à point tout simple comme sur des liens à haut débit. Les liens physiques que l'on retrouve souvent dans le protocole PPP sont ceux qui ont été définis par les normes EIA RS232, RS422, RS485, V.35 ainsi que par l'interface HSSI (*High-Speed Serial Interface*). Les vitesses de transmission dépendent de la couche physique. Le protocole PPP peut fonctionner sur un canal RNIS (réseaux numériques d'intégration de services) de base et sur les supports T1 ou E1 de même que sur un OC-3 du réseau optique synchrone (SONET) ou un module de transport synchrone de niveau 1 (STM1) de la norme de hiérarchie numérique synchrone (SDH).

Le protocole PPP comprend plusieurs composantes permettant d'établir une liaison stable.

Les éléments composant le protocole PPP sont illustrés à la figure 3.30. Nous en donnons ci-dessous une description :

1- Le PPP peut soutenir soit un lien asynchrone de 8 bits de données sans bits de parité, soit un lien synchrone orienté bit (*bit* O*riented*). Ceci est conforme aux spécifications du protocole HDLC (*High-level Data Link Control*). Le RFC 1662 donne la définition d'utilisation du protocole HDLC pour le protocole PPP.
2- Un LCP (*Link Control Protocol*/Protocole de contrôle de liaison) pour établir, configurer et tester la connexion du lien des données. La définition en est donnée dans le RFC 1661.
3- Le protocole d'authentification est défini par le RFC 1334. Le RFC 1994 a procédé à une mise à jour de certains des aspects du RFC 1334, notamment en regard du protocole CHAP (*Challenge Handshake Authentification Protocol*). Par ailleurs, le RFC 2284 fait état du protocole EAP (*Extensible Authentification Protocol*).
4- Une famille de NCP (*Network Control Protocol*/Protocole de gestion de réseaux) est propre à différents protocoles de la couche réseau. Dans notre cas, le IPCP (*Internet Protocol Control Protocol*/Protocole de contrôle IP) est précisé dans le RFC 1332.

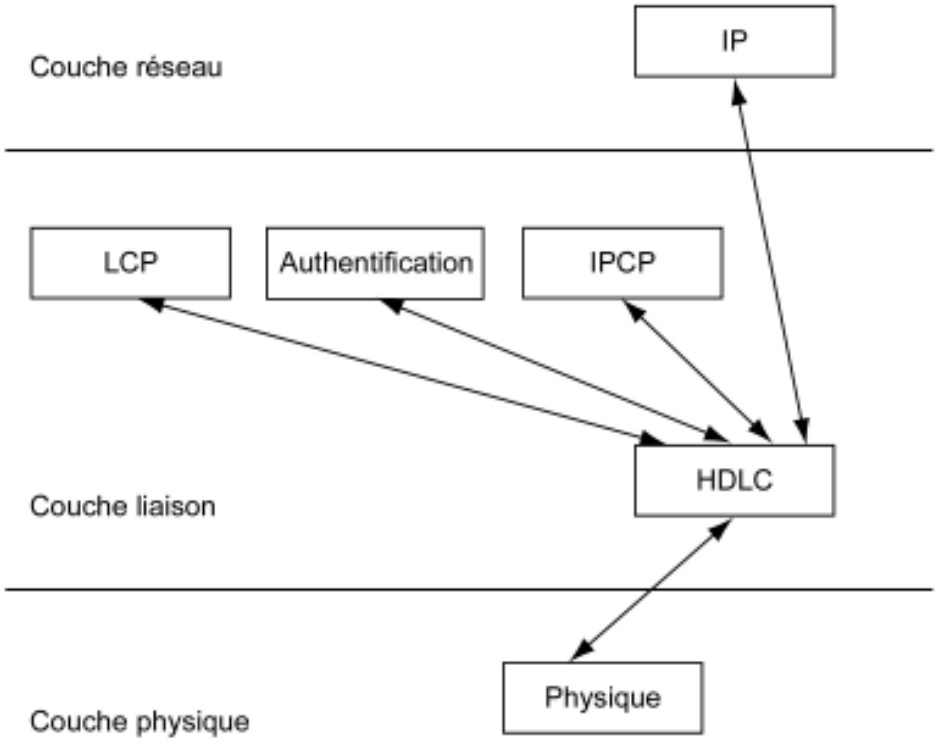


Figure 3.30 Représentation type de la couche liaison du PPP

En tout premier lieu, l'établissement de la liaison est amorcé par le LCP (*Link Control Protocol*/Protocole de contrôle de liaison) (cf. le RFC 1661). Par la suite, il y a authentification de la liaison si nécessaire. Seulement alors, le module NCP (*Network Control Protocol*/Protocole de gestion de réseaux) assume la relève et négocie les paramètres de l'interface réseau. Chacun de ces modules génère des PDUs qui ont été encapsulés dans des trames HDLC. Une fois que le NCP a défini l'interface, la couche réseau peut alors entamer ses procédures. La suite reste conforme aux mécanismes définis par l'OSI, c'est-à-dire les SDUs reçus de la couche réseau par la couche liaison seront aussi encapsulés dans des trames HDLC.

3.8.2.1 La trame HDLC dans le protocole PPP

Le protocole PPP s'est inspiré de celui du HDLC (cf. le RFC 1662). Le format de cette trame est illustré à la figure 3.31. Tout comme le HDLC, la trame est délimitée par des fanions dont la valeur est 0x7e en hexadécimal. Le récepteur synchronise son horloge à la réception de ces fanions. Il existe aussi un champ

adresse qui est toujours à la valeur 0xff. Le champ contrôle est toujours à 0x03. Puisque les valeurs des champs adresse et contrôle conservent toujours les mêmes valeurs, le protocole PPP peut se permettre dans certaines applications de ne pas procéder à la transmission de ces deux champs.

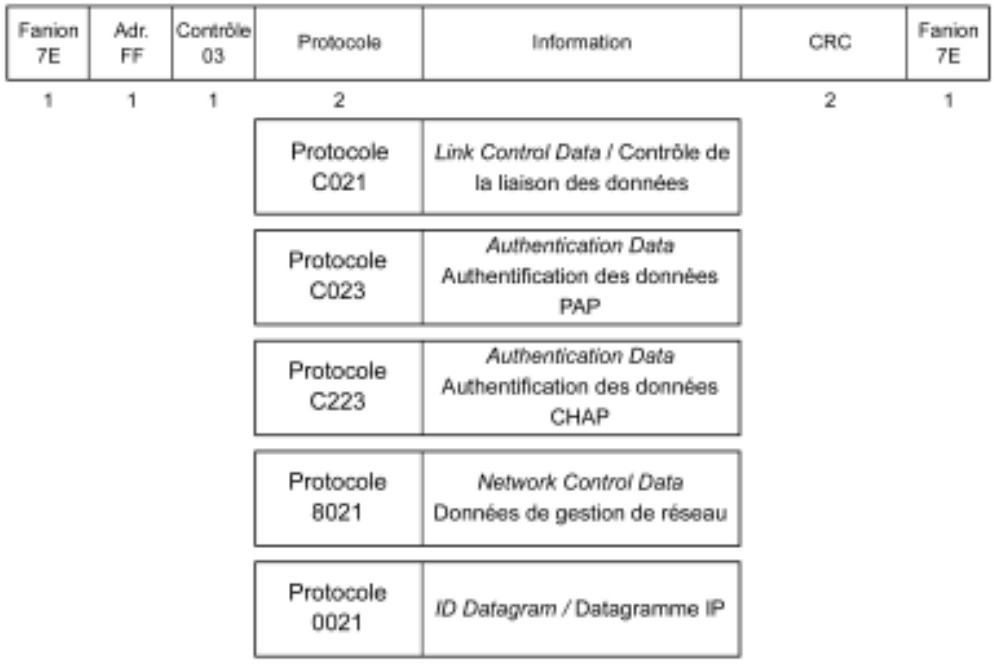


Figure. 3.31 Représentation type des trames PPP selon le protocole HDLC

Le champ protocole suit le champ contrôle pour préciser le protocole de l'information encapsulée. Par exemple, si le protocole IP se trouve dans une trame, le code du champ protocole prendra la valeur 0x21.

Le champ suivant qui est le champ information renferme le paquet encapsulé. On retrouve le champ CRC suivant le champ information qui assume la détection d'erreurs de bit dans la trame.

Conformément au RFC 1547, le protocole PPP doit être transparent au niveau des données. Cela signifie que le contenu du champ information de la trame PPP peut renfermer n'importe laquelle des combinaisons de bits dans les octets qui s'y trouvent. Des données ayant une séquence de bits identique au fanion pourraient être interprétées comme telles alors qu'ils n'auraient rien à voir avec la synchronisation. Il faut donc envisager un mécanisme permettant d'identifier et d'isoler les données ayant la même séquence que le fanion.

Les séquences escape du protocole PPP sont utilisées dans chacun des cas suivants :

1- Lorsque le caractère 0x7e se trouve dans les données d'information. Étant donné que cette forme ressemble à celle d'un fanion, cette valeur sera remplacée par les deux caractères 0x7d et 0x5e avant leur transmission. 0x7d est la valeur attribuée à escape.
2- Lorsque l'octet 0x7d se trouve parmi les données. Cet octet sera remplacé par une séquence de deux octets dont les valeurs seront 0x7d et 0x5d. Ces octets seront transmis à la place de celui dont la valeur est 0x7d. Sur réception de cette séquence, le récepteur saura que celle-ci remplace l'octet 0x7d.
3- Lorsque, par défaut, un octet a une valeur moindre que 0x20, il est remplacé en procédant au complément du 6e bit de cette valeur et en transmettant à sa place 0x7d suivi de l'octet ainsi complété. Exemple : l'octet 0x01 est transmis comme 0x7d 0x21

	00000001	=	0x01
01111100	00100001	=	0x7d 0x21

Cette transformation s'impose parce que certaines commandes du modem ont des valeurs plus petites que 0x20. Étant donné qu'il est possible d'avoir recours au protocole PPP dans le cas d'une communication par modem, il importe donc de transformer ces octets avant de procéder à la transmission des données, car ces dernières pourraient être confondues par des commandes.

3.8.2.2 Système de fonctionnement des modules PPP

Il importe de savoir que le démarrage des procédures fonctionnelles du PPP prend son origine dans les couches supérieures et que c'est à partir de là que tout démarrage est amorcé. Tout comme nous en avons déjà fait mention, le lancement de chacun des modules fonctionnels se fait séquentiellement.

Le NCP (*Network Control Protocol*/Protocole de contrôle réseau) reçoit une demande d'ouverture *Open* de la couche supérieure. Le NCP n'entame pas encore ses procédures, mais envoie une demande *Open* au module d'authentification. Le NCP n'amorce ses procédures qu'une fois que le signal *Up* est reçu du module d'authentification.
Tout comme le NCP, le module d'authentification émet une demande d'ouverture *Open* au LCP. Il en attend aussi un signal *Up* avant de procéder à sa validation.

Le LCP ne peut entamer de procédures permettant d'établir une liaison qu'une fois la couche physique elle-même établie et opérationnelle et que le module d'authentification lui ait émis un signal *Open*.

Une fois que le module LCP a établi sa connexion (tel que nous le verrons ultérieurement), il envoie le signal *Up* au module d'authentification pour l'informer qu'il est opérationnel. À ce moment seulement, le module d'authentification amorce ses procédures.

Le module d'authentification pourrait ne pas avoir à procéder à une validation. Toutefois, ce module, une fois opérationnel, doit adresser au module NCP un signal *Up*, et ce, peu importe le cas. Le NCP entre alors en jeu pour négocier et définir l'interface réseau. Une fois l'interface définie, la couche réseau en est, à ce moment seulement, prévenue.

3.8.2.3 Protocole LCP et établissement d'une liaison

Le RFC 1661 présente un diagramme d'état que nous reproduisons à la figure 3.32.

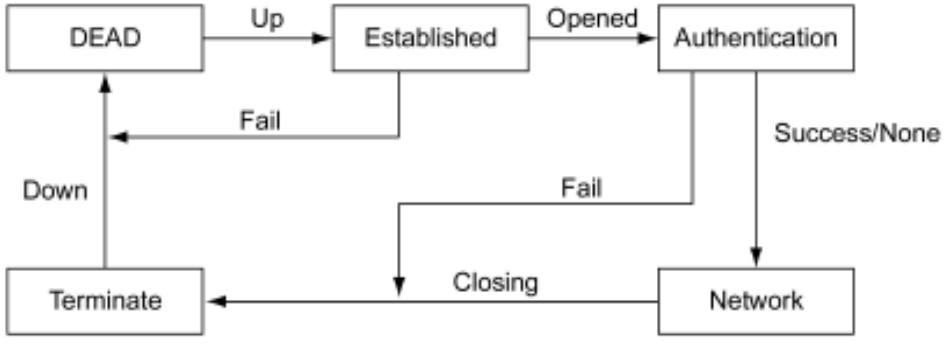


Figure 3.32 Représentation du fonctionnement du protocole PPP

Le point de départ et de fin de l'établissement d'une liaison est l'état *Dead*. Lorsque la couche physique nous prévient qu'elle est opérationnelle (*Up*), l'état d'établissement de la liaison est amorcé et le module LCP la prend en charge. L'état *Authentication* est prévenu de l'établissement de la liaison grâce au signal d'ouverture *Opened*. Le module d'authentification, à qui cet état incombe, en valide l'accès au besoin. Si l'accès est validé, le signal *Success* fait la transition à l'état *Network*. Dans le cas où il n'y a pas de signal, le contrôle passe directement à l'état *Network*. En cas d'échec, avec l'envoi du signal *fail*, le contrôle se dirigera vers l'état *Terminate*. Le NCP assume ce dernier état et définit l'interface réseau.

Lorsque la liaison prend fin, chacun des modules doit être réinitialisé. Cette procédure se fait par le signal *Closing* et par l'état *Terminate*. Une fois toutes les étapes complétées, les modules sont déclarés *Closed*. Ceci nous est indiqué par le signal *Down* et le retour à l'état *Dead*.

Naturellement, tout échec dans l'établissement d'un état déclenche la fin de toute liaison.

3.8.2.4 Les messages du protocole LCP

La négociation du LCP a pour but d'établir la liaison avec la couche liaison distante. À cette fin, des PDU sont envoyés. Chaque PDU[2] comprend des paramètres négociables.

Toute tentative d'établissement d'une liaison débute par le message *Configure-Request*. Chacun des terminaux désireux d'établir une communication doit négocier ses propres paramètres et doit donc établir sa propre liaison.

Le format du message LCP illustré à la figure 3.33 comprend quatre champs : code, identification, longueur et paramètres.

Figure 3.33 Représentation type du PDU LCP

Le code dont la longueur est de 1 octet permet d'identifier la nature de la liaison du protocole LCP. Par exemple, le type *Configure-Request* aura un code dont la valeur est à 1. Le tableau 3.1 donne l'assignation des codes aux différentes catégories de LCP.

[2] Nous utilisons le terme PDU pour inclure tous les champs de la couche liaison. Le terme message est utilisé pour décrire le champ information HDLC exclusion faite du code « Protocole ».

Tableau 3.1 Liste des codes LCP

Code	Nature de la liaison du LCP
1	*Configure-Request* (Configuration-demande)
2	*Configure-Ack* (Configuration-A/R)
3	*Configure-Nak* (Configuration-accusé de réception négatif)
4	*Configure-Reject* (Rejet)
5	*Terminate-Request* (Cessation-demande)
6	*Terminate-Ack* (Cessation A/R)
7	*Code-Reject* (Code rejeté)
8	*Protocol-Reject* (Protocole rejeté)
9	*Echo-Request* (Demande par écho)
10	*Echo-Reply* (Réponse par écho)
11	*Discard-Request* (Demande-renoncer)

L'identifiant d'une longueur de 1 octet permet de faire un suivi entre l'émission de la formulation de la demande et l'envoi de la réponse.

Le champ longueur comporte 2 octets. Il indique la longueur des champs LCP, y compris les champs code, identification, longueur et paramètres.

La longueur du champ paramètres varie de 0 à plusieurs octets. Le format de ce champ dépend de la nature de la liaison du LCP.
Les messages LCP sont répartis en trois catégories :

- Les messages utilisés lors de l'établissement de paramètres de liaison sont : *Configure-Request, Configure-Ack, Configure-Nak* et *Configure-Reject*.
- Les messages utilisés lors de la cessation de la liaison sont : *Terminate-Request* et *Terminate-Ack*.
- Les messages d'entretien de la liaison sont : *Code-Reject*, *Protocol-Reject*, *Echo-Request*, *Echo-Reply* et *Discard-Request*.

La négociation des paramètres débute par le message *Configure-Request*. Les échanges de messages subséquents vont dépendre de plusieurs facteurs tels que le rejet de paramètres non reconnus. Quelques exemples de situations particulières seront présentés.

Ces paramètres sont aussi connus comme étant des options de configuration. Lorsque ceux-ci ne sont pas spécifiés au moment de la négociation, leur valeur par défaut est la seule à être prise en compte. Chaque paramètre comprend trois champs :

- le type de paramètre, long de 1 octet;
- la longueur, incluant les champs type;
- la longueur et valeur, long de 1 octet;
- la valeur du paramètre dont la longueur variable va de 0 à plusieurs octets.

La figure 3.34 illustre le format du paramètre qui se trouve dans un PDU de la couche liaison. S'il n'y a pas de valeur dans le paramètre, la longueur est à 2 (type et longueur).

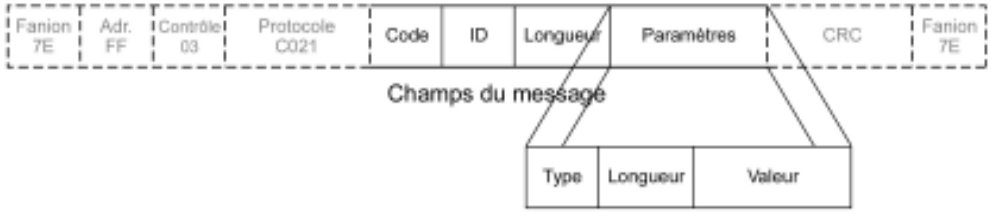


Figure 3.34 Représentation du format du paramètre d'un PDU de la couche liaison

Les différents types de paramètres selon le RFC 1661 avec leur code correspondant sont indiqués dans le tableau 3.2.

Tableau 3.2 Types de paramètres (options)

Code de type	Option correspondante de configuration
0	Réservé
1	*Maximum-Receive-Unit*/Unité de réception maximale
3	*Authentication Protocol*/Protocole d'authentification
4	*Quality-Protocol*/Protocole qualité
5	*Magic-Number*/Nombre magique
7	*Protocol-Field-Compression*/Protocole du champ de compression
8	*Address-and-Control-Field-Compression*/Contrôle des champs compression et adresse

Option 1 - Maximum-Receive-Unit / Unité de réception maximale

Cette option de configuration est transmise pour aviser l'entité distante de la grandeur maximale des paquets pouvant être reçus. Si l'option n'est pas transmise, la grandeur des paquets par défaut est de 1500 octets.

Le champ valeur de cette option nous signale le nombre d'octets maximum accepté. Ce champ comprend 2 octets.

Option 3 - Authentication Protocol / Protocole d'authentification

Cette option de configuration permet de négocier la méthode d'authentification à laquelle on aura recours au besoin.

Par défaut, l'authentification n'est pas requise. Il est possible de négocier qu'une seule méthode à la fois. Si une méthode est refusée par l'entité distante, une nouvelle méthode peut être proposée.

Les méthodes d'authentification ne sont pas nécessairement les mêmes dans les deux sens. L'authentification pourrait ainsi être utilisée dans un sens, mais pas dans l'autre.

Le champ valeur comprend le code de l'authentification proposée. Les valeurs possibles sont :

C023	PAP (*Password Authentification Protocol*/Protocole d'authentification PAP)
C223	CHAP (*Challenge Handshake Authentification Protocol*/ Protocole d'authentification CHAP)

Il y a aussi un champ de données qui renferme, au besoin, des informations supplémentaires exigées par le protocole.

Option 4 - Quality Protocol / Protocole qualité

L'option *Quality Protocol* permet l'utilisation d'un protocole de surveillance de la qualité du lien. Lorsque l'option n'est pas exploitée, l'information de la surveillance n'est pas transmise.

Cette option n'est pas exigée dans les deux sens. Le champ valeur, long de 2 octets, a recours au code C025 pour indiquer l'utilisation du *Link Quality Report*. Les données qui suivent incluent des informations exigées par le protocole.

Option 5 - Magic Number / Nombre magique

Cette option permet d'utiliser une méthode de détection de liens bouclés ou d'identifier d'autres anomalies de la couche liaison.

Lorsque cette option est utilisée, le transmetteur du message *Configure-Request* qui a transmis un numéro unique dans le champ valeur s'assure que ce numéro a bien été reçu par le message *Configure-Request* dans l'autre sens. Si tel n'est pas le cas, il en déduit qu'il existe un lien bouclé. Une alarme peut alors se déclencher.

Option 7 – PFC (Protocol Field Compression / Protocole du champ de compression)

L'option PFC est utilisée pour négocier l'application d'une méthode de compression du champ du protocole PPP.

Option 8 – ACFC (Address and Control Field Compression / Contrôle des champs compression et adresse)

L'option ACFC est utilisée pour négocier l'application d'une méthode de compression des champs adresse et contrôle de la couche liaison.

3.8.2.5 La négociation du LCP

Pour mieux comprendre l'opération de la négociation du LCP, il est bon de nommer les entités du LCP qui y sont impliquées, notamment les entités A et B. La négociation d'un LCP commence par la transmission du message *Configure-Request,* indifféremment par l'entité A ou B. Les options désirées figurent dans le premier message. Si l'entité distante B accepte les options, elle transmet un message *Configure-Ack* en réponse et en y faisant figurer les mêmes options et valeurs.

Par contre, si la valeur de l'une des options est inacceptable, l'entité distante B transmet, dans le message *Configure-Nak*, l'option qui fait l'objet d'un litige et la valeur désirée.

L'entité A doit alors retransmettre les messages *Configure-Request* avec les options proposées ou en exclure définitivement les options qui lui semblent être inacceptables.

Afin d'envisager cette étape de la négociation complète, il faut que le message *Configure-Ack* soit reçu.

Après réception du message *Configure-Request,* si l'entité B rejette une option, cette entité transmet le message *Configure-Reject* avec la mention rejetée. L'entité A doit alors transmettre un nouveau *Configure-Request* en excluant l'option rejetée. Le message *Configure-Ack* doit être reçu avant de terminer la liaison avec *Terminate-Request* en réponse à *Terminate-Ack*.

La négociation est requise dans les deux sens.
La figure 3.35 nous en donne un bon exemple fort simple. Deux entités du LCP négocient les paramètres suivants : longueur de trame (MRU), authentification et compression des champs adresses et contrôle (ACFC).

La longueur proposée par A est de 1700 octets. L'authentification est PAP. L'entité B rejette la compression. Par la suite, elle propose une longueur de 1500 octets. L'entité A confirme.

Entités	Messages	Options
A	*Configure-Request*	01 06A4, 03 C0 23, 07
B	*Configure-Reject*	07
A	*Configure-Request*	01 6A4, 03 C023
B	*Configure-Nak*	01 05DC
A	*Configure-Request*	01 05DC, 03 C023
B	*Configure-Ack*	01 05DC, 03 C023

Figure 3.35 Exemple de négociation d'un message

La figure 3.36 illustre la trame complète du premier *Configure-Request* transmis par l'entité A.

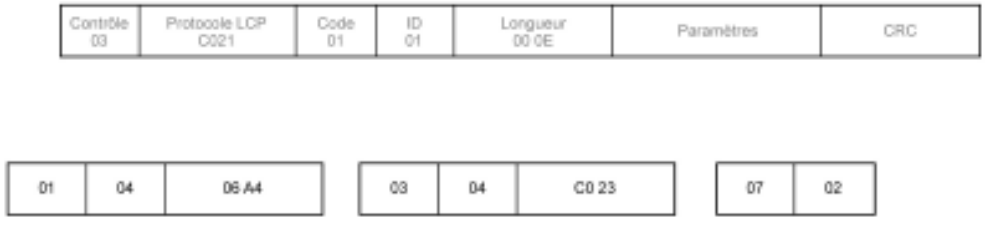


Figure 3.36 Exemple du premier *Configure-Request*

Nous pouvons noter que la longueur inclut toujours, s'il y a lieu, les champs code, longueur, valeur et données. Ainsi, la longueur des paramètres est de 14 (0E), parce que cette longueur inclut les champs code 01, l'ID 01, la longueur 00 0E et le contenu des paramètres. Le paramètre d'authentification PAP a une longueur de 4, parce que la longueur inclut les champs code 03, la longueur 04 et

la valeur C0 23. Par contre, le paramètre compression a une longueur de 2, parce que cette longueur ne comprend que le code 07 et la longueur. Il n'y a pas de valeur dans ce cas.

Les codes des trames transmises sont :

A : 03 C0 21 01 01 00 0E 01 04 06 A4 03 04 C0 23 07 02
B : 03 C0 21 04 01 00 06 07 02
A : 03 C0 21 01 02 00 0B 01 04 06 A4 03 04 C0 23
B : 03 C0 21 03 02 00 07 01 04 05 DC
A : 03 C0 21 01 03 00 0B 01 04 05 DC 03 04 C0 23
B : 03 C0 21 02 03 00 0B 01 04 05 DC 03 04 C0 23

3.8.2.6 L'authentification

La demande d'authentification suit la négociation du LCP. Cette demande est optionnelle. Comme nous l'avons déjà mentionné, il existe plusieurs protocoles d'authentification. Le plus simple étant le PAP (*Password Authentification Protocol*), est spécifié dans le RFC 1334. Le CHAP (*Challenge-Handshake Authentication Protocol*) nouvelle version est défini dans le RFC 1994. Il existe aussi d'autres protocoles d'authentification notamment le EAP (*Extensible Authentification Protocol*) que l'on retrouve dans le RFC 2284. Il existe aussi des protocoles d'authentification propriétaire comme le MS-CHAP et le Shiva SPAP.

Le protocole d'authentification CHAP est décrit ici sommairement à titre d'exemple. Notons que ces protocoles ne permettent d'effectuer que l'identification et l'authentification des stations vis-à-vis de leurs pairs. Ils ne s'occupent pas de confidentialité.

Le protocole CHAP est basé sur deux messages : *Challenge* et *Response*. Il est utilisé pour vérifier périodiquement l'identité du système distant. Cette vérification se fait en trois étapes et démarre aussitôt que le lien est établi. Le système qui désire effectuer la vérification, connu aussi sous le nom de authentificateur, transmet le message *Challenge* au système distant. Il y inclut sa propre identité. Le système distant lui retourne, par le message *Response*, la valeur du résultat d'un calcul qu'il a effectué avec une fonction connue sous le nom de « *one-way hash* ». Il transmet ainsi son identité en ne la faisant connaître qu'à l'authentificateur. Ce dernier vérifie cette valeur avec le résultat de son propre calcul. Si la vérification est faite, il transmet un accusé de réception par le message *Success*; autrement, il transmet le message *Failure* et la connexion est terminée. L'authentificateur décide de répéter cette vérification périodiquement.

3.8.2.7 Protocole de contrôle de IP

Le protocole de contrôle IP, IPCP (*IP Control Protocol*) est utilisé par le protocole PPP pour négocier les paramètres d'utilisation du protocole réseau et plus spécifiquement IP.
Le IPCP est décrit dans le RFC 1332. Le processus de négociation pour le IPCP utilise le même format de message et la même procédure que pour le LCP.

3.9 Exercices

3.1

Le protocole LLC est utilisé pour :

1. Traiter les fichiers localement.
2. Améliorer la sécurité.
3. Établir un contact avec le MAC du terminal distant.
4. Compléter les fonctionnalités manquantes dans la sous-couche MAC pour la couche liaison.
5. Amorcer une application du terminal et vérifier les erreurs.

3.2

La primitive DL_DATA est :

1 Une primitive confirmée.
2 Une primitive non confirmée.
3 Une primitive de circuit virtuel.
4 Une primitive de datagramme.
5 Une primitive de connexion.

3.3

La primitive DL_CONNECT est :

1 Une primitive confirmée.
2 Une primitive non confirmée.
3 Une primitive de circuit virtuel.
4 Une primitive de datagramme.
5 Une primitive de connexion.

3.4

La primitive DL_DISCONNECT est :

1 Une primitive confirmée.
2 Une primitive non confirmée.
3 Une primitive de circuit virtuel.
4 Une primitive de datagramme.
5 Une primitive de connexion.

3.5

Le format du PDU LLC comprend :

1 Toujours les champs adresse, contrôle et information.
2 Toujours les champs adresse, contrôle et CRC.
3 Toujours les champs adresse, contrôle, information et CRC.
4 Peut ne pas contenir le champ adresse.
5 Peut ne pas contenir le champ CRC.

3.6

Le PDU LLC est détecté (déterminez le début et la fin) grâce :

1 À un fanion au moins placé en début de trame.
2 Au champ adresse placé en début de trame.
3 Au champ contrôle.
4 S'il n'y a pas de champ information.
5 Au CRC en fin de trame.

3.7

Le PDU LLC est déchiffré (analysez le contenu) grâce :

1 À un fanion au moins en début de trame.
2 Au champ adresse placé en début de trame.
3 Au champ contrôle.
4 S'il n'y a pas de champ information.
5 Au CRC placé en la fin de trame.

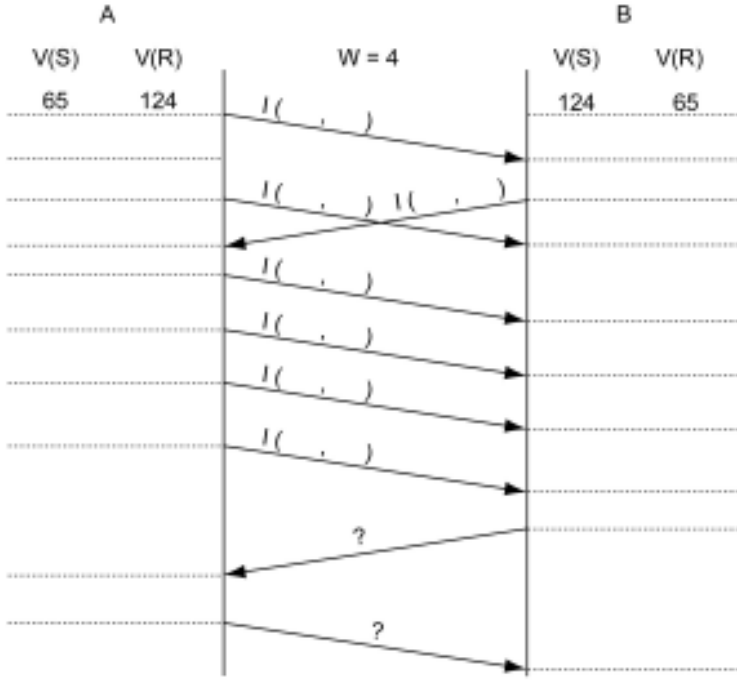
3.8

La transparence de la trame signifie que :

1 La trame est invisible.
2 La trame n'est pas reconnue.
3 La trame peut contenir n'importe quelle combinaison de bits.
4 La trame doit contenir un fanion.
5 La trame comprend des bits de sécurité.

3.9

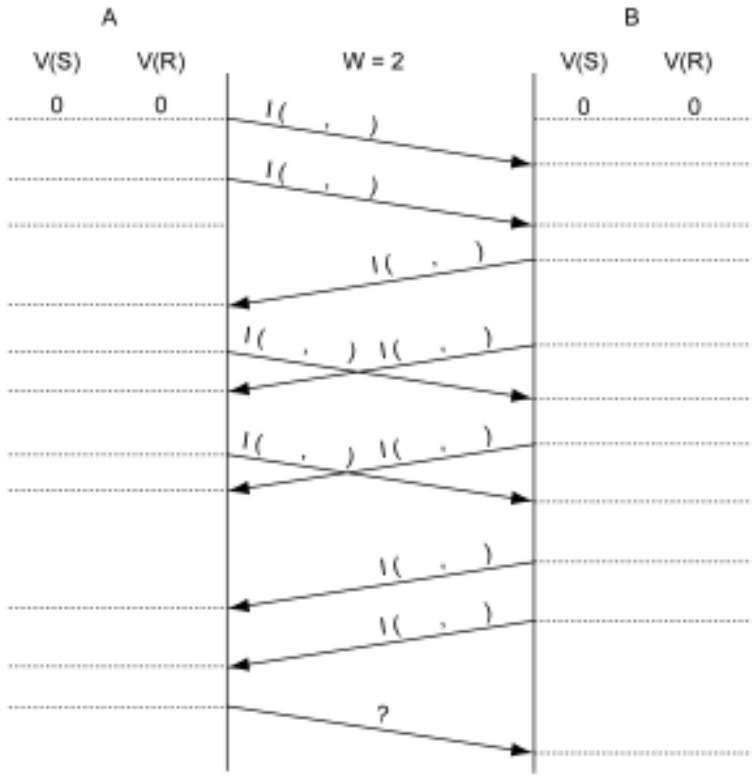
Complétez le diagramme suivant selon le protocole LLC



Complétez le diagramme en incluant les valeurs des compteurs V(S), V(R), N(S), N(R) et indiquez le type de trame avec leurs valeurs de compteurs s'il y a lieu. Notez que la fenêtre est à 4 et que l'un des systèmes pourrait ne pas se comporter normalement.

3.10

Complétez le diagramme suivant selon le protocole LLC



Complétez le diagramme en incluant les valeurs des compteurs V(S), V(R), N(S), N(R) et indiquez le type de trame avec leurs valeurs de compteurs s'il y a lieu. Notez que la fenêtre est à 2 et que l'un des systèmes pourrait ne pas se comporter normalement.

3.11

Deux entités LCP entament une négociation. L'entité A amorce la négociation en indiquant les options suivantes :

Longueur de trame (MRU) – code : 01, valeur : 2300 (octets)
Authentification Microsoft CHAPv2 – code : 03, valeur : C223, données : 81
Compression du champ PPP (PFC) – code : 07, valeur : aucune
L'entité B rejette le PFC complètement.
L'entité A représente ses options sans le PFC.
L'entité B refuse l'authentification Microsoft CHAPv2, mais propose plutôt l'authentification Microsoft CHAPv1 (code : 03, valeur : C223, données : 80) et modifie la longueur de trame (MRU) à 1500 octets.

a) Indiquez les messages LCP, avec leurs options, échangés durant cette négociation :
b) Indiquez en hexadécimal chaque trame HDLC correspondant à ces messages et options incluant tous les champs sauf le FCS.

4 Protocoles à compétition

4.1 Introduction

ALOHA est l'une des premières techniques à compétition ayant recours à la diffusion radio par paquets. Elle a été conçue et mise au point par une équipe de chercheurs de l'Université d'Hawaii au cours des années 1970. Cette technique veut que la transmission soit ouverte à qui désire transmettre. Les stations faisant partie de ce réseau transmettent des paquets lorsqu'elles le désirent sans devoir se soucier pour autant des autres stations. La station qui veut transmettre attend alors un temps prédéterminé (*Time slot*/Intervalle de temps) pour recevoir un accusé de réception. Si ce dernier n'a pas été reçu, la station retransmet en présumant qu'il y a eu erreur lors de la transmission. Après un certain nombre d'essais, la station cesse de transmettre.

Si deux stations tentent de transmettre simultanément, il y a collision entre les paquets et ceux-ci sont alors endommagés. La collision est détectée par un mécanisme de détection d'incidents qui se trouve dans chaque station. Le protocole ALOHA est simple mais inefficace lorsque l'utilisation des canaux est moyennement en demande. L'utilisation maximale du médium de transmission est en deçà des 18%.

Afin de pallier à ce problème, le protocole CSMA (*Carrier Sense Multiple Access*/Accès multiple avec écoute de porteuse) a été élaboré et mis au point. Une station émettrice ou station source doit d'abord écouter la ligne de transmission avant de transmettre. Si le médium est occupé, la station attend et la collision est ainsi évitée. La station ne transmettra que lorsque la ligne sera libre, c'est-à-dire lorsqu'aucune station ne transmet. Par contre, il peut arriver que deux stations émettrices écoutent au même moment et qu'il y ait une transmission simultanée, cela cause alors une collision.

Dans ce dernier cas, une nouvelle amélioration a été apportée grâce à l'élaboration du protocole CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*/Accès multiple avec écoute de porteuse et détection de collision). La différence entre CSMA/CD et CSMA est la fonction de détection. Outre le fait

d'écouter la ligne de transmission avant de transmettre, une station source qui se trouve dans un réseau CSMA/CD va continuer d'écouter même pendant sa transmission. En cas de collision, les stations émettrices la détecteront immédiatement et interrompront leur transmission. Elles envoient alors un signal de brouillage. Le processus de déférence est utilisé pour éviter que d'autres collisions ne surviennent. Si toutes les stations devaient attendre le même laps de temps avant de retransmettre, d'autres collisions se produiraient. Avec le processus de déférence, chaque station attend pendant un temps d'une durée aléatoire et connu sous le nom de *Backoff Delay*.

L'algorithme de la transmission est illustré à la figure 4.1. Lorsqu'une collision a lieu, plusieurs essais sont effectués selon l'algorithme BEB (*Binary Eponential Backoff*/ Algorithme de repli exponentiel binaire).

4.2 Temps d'attente avant la retransmission

Chaque station qui a subi une collision va opter pour un nombre aléatoire qui correspond au nombre d'intervalles de temps (*slots)* et va attendre un temps équivalent à ce nombre avant de retransmettre. Chaque intervalle de temps (*slot)* est d'une durée équivalente à deux fois le temps de propagation à travers le segment. Le choix du temps d'attente peut avoir des effets sur l'occurrence de nouvelles collisions. Nous considérons les deux cas extrêmes :
Dans le cas d'une marge de valeur faible, on fait face à une plus grande probabilité de collisions répétées car deux stations pourraient avoir fait le même choix.
Pour une marge de valeur élevée, il y a une plus faible probabilité d'opter pour la même valeur qui établira le temps d'attente; par contre, le temps moyen d'attente avant la retransmission est plus grand. Les stations qui veulent transmettre pourraient attendre inutilement.

4.3 Algorithme de temporisation de retransmission (*Truncated Binary Exponential Backoff)*

Afin de réduire le nombre de collisions et d'éviter les longues attentes avant la transmission, un algorithme a été pensé et conçu. Cet algorithme permet aux stations impliquées dans une collision, de choisir aléatoirement un chiffre qui correspondra à un nombre d'unités d'intervalles de temps (*slots)*. Chaque station se devra d'attendre le temps correspondant au nombre choisi d'unités d'intervalles de temps avant de transmettre.

Au début, le choix se fait entre 0 et 1. C'est-à-dire, si une station choisit 0, la transmission se fait immédiatement. Par contre, si elle choisit 1, alors elle devra attendre le temps correspondant à une unité d'intervalle de temps. S'il y a toujours collisions, la plage de chiffres à choisir augmente et chaque station choisit au hasard un nouveau chiffre pour définir le nombre d'unités d'attente. Comme nous l'avons mentionné plus haut, plus la plage d'unités est petite, plus grand est le risque de collision. Par contre, plus le choix d'unités est élevé plus grand est le risque d'une attente longue. Cet algorithme se nomme algorithme de temporisation de retransmission (*Truncated Binary Exponential Backoff - TBEB).*

La marge r qui définit le nombre d'unités d'attente est représentée par : $0 \leq r < 2^k$, où k désigne le nombre de tentatives de retransmission par la station. Pour la première tentative, soit k = 1, la marge est de 0 et 1. C'est-à-dire que la station va attendre 0 ou 1 unité de temps. L'unité de temps est égale à 2 fois le temps de propagation le long d'un segment.

Si, après le premier essai, il y a collision parce qu'au moins deux stations ont choisi le même nombre d'unités, le 2e essai surviendra lorsque k = 2 et que la marge de choix d'unités d'attente sera 0 à 3. La station attendra 0, 1, 2, ou 3 unités de temps avant de retransmettre. Le nombre d'unités est choisi de manière aléatoire. Tant que les collisions se succèdent, la marge continue de croître jusqu'à ce que k atteigne la valeur 10 (soit un nombre compris entre 0 et 1023). Si les collisions persistent, k reste à la valeur 10 jusqu'au 16e essai, ce après quoi l'erreur « collisions excessives » est rapportée au MAC.

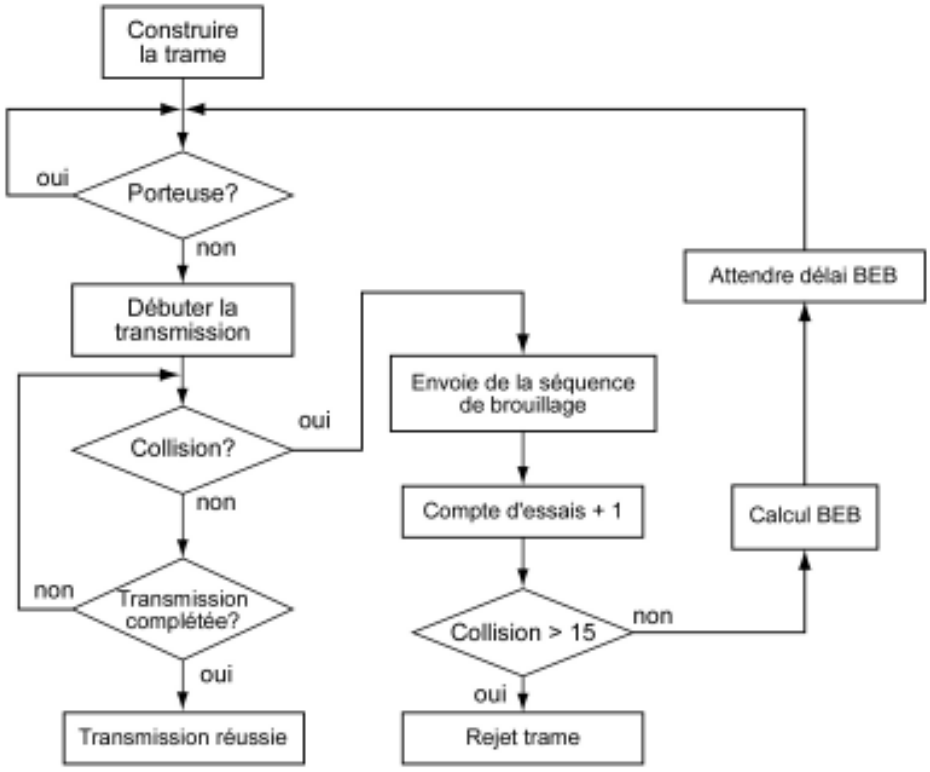


Figure 4.1 Représentation type de l'algorithme de transmission CSMA/CD

L'algorithme de transmission du CSMA/CD est illustré à la figure 4.1. La porteuse représente la transmission par d'autres stations. La station CSMA/CD surveille la ligne pour savoir si elle est occupée avant de transmettre. Même au cours de la transmission, la station continue de surveiller la ligne, mais cette fois-ci, pour détecter toute possibilité de collision. Après chaque collision, le BEB est appliqué et un compteur de tentatives est mémorisé.

L'algorithme de réception est synchronisé dans ses fonctions à celui de transmission. La figure 4.2 illustre cet algorithme qui doit être analysé au même titre que celui de la transmission présentée à la figure 4.1.

Le récepteur va accepter toutes les trames qui traversent le segment. Chaque trame va alors être analysée et si l'adresse de la trame ne correspond pas à la sienne, la trame reçue est ignorée. La trame n'est acceptée et remise à la couche supérieure qu'après lui avoir ôté l'en-tête et que la station se soit assurée que la trame ne comporte également aucune erreur de transmission.

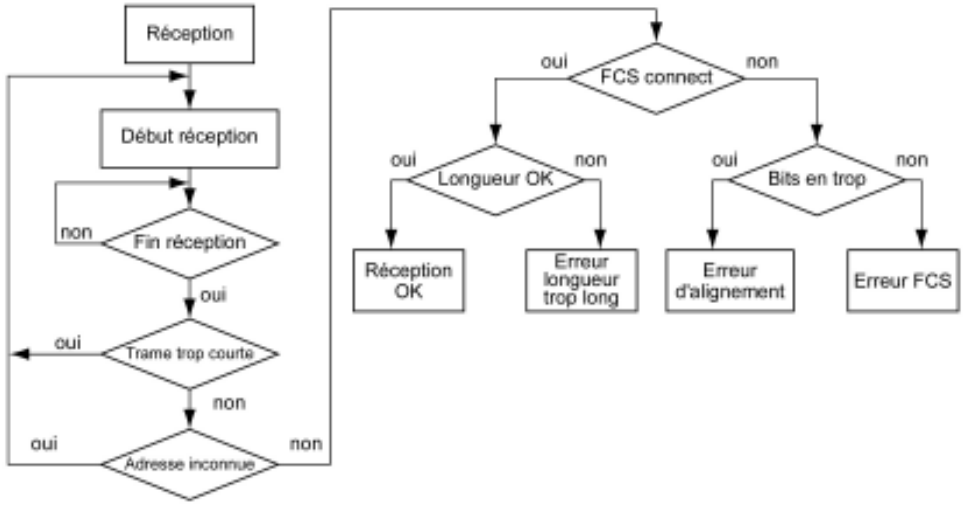


Figure 4.2 Représentation type d'algorithme de réception CSMA/CD

4.4 Longueur minimale de la trame

Afin de comprendre pourquoi une longueur minimale est très importante, considérons la figure 4.3. Supposons que la station A décide de transmettre à la station la plus éloignée du LAN : la station B. Les temps de transmission sont normalisés à 1 et le délai de propagation *a* à 0.5.

Le terminal A se met à transmettre au temps t_0. La trame est propagée le long du segment et doit normalement parvenir à la station B, à $t_0 + a$. Si au moment où la trame est presque arrivée à la station B, c'est-à-dire à $t_0 + a - \varepsilon$, la station B qui ne voit encore rien sur la ligne se met à transmettre. La collision survient juste à ce moment là, soit à $t_0 + a$. La collision va être propagée vers le terminal A et ce dernier en prendra connaissance. Cela signifie qu'il faudra un temps égal à $t_0 + 1 - \varepsilon$ pour que la station A réalise qu'il y a eu collision. Mais pour que cette même station comprenne que cela est causé par sa propre transmission d'une trame, il faudrait qu'elle soit toujours en train de transmettre. La longueur de la trame minimale est déterminée à partir de ce raisonnement.

La longueur minimale d'une trame est telle que la durée d'émission de celle-ci est égale à la durée de la tranche canal, c'est-à-dire deux fois le temps de propagation du signal à travers le bus (2 *a*).

Une trame plus courte est considérée par tous les utilisateurs du réseau comme le résultat d'une collision. Une longueur maximale est fixée afin d'empêcher qu'un

utilisateur mal intentionné n'occupe le canal pendant une trop longue durée. Cette longueur maximale est un choix préétabli.

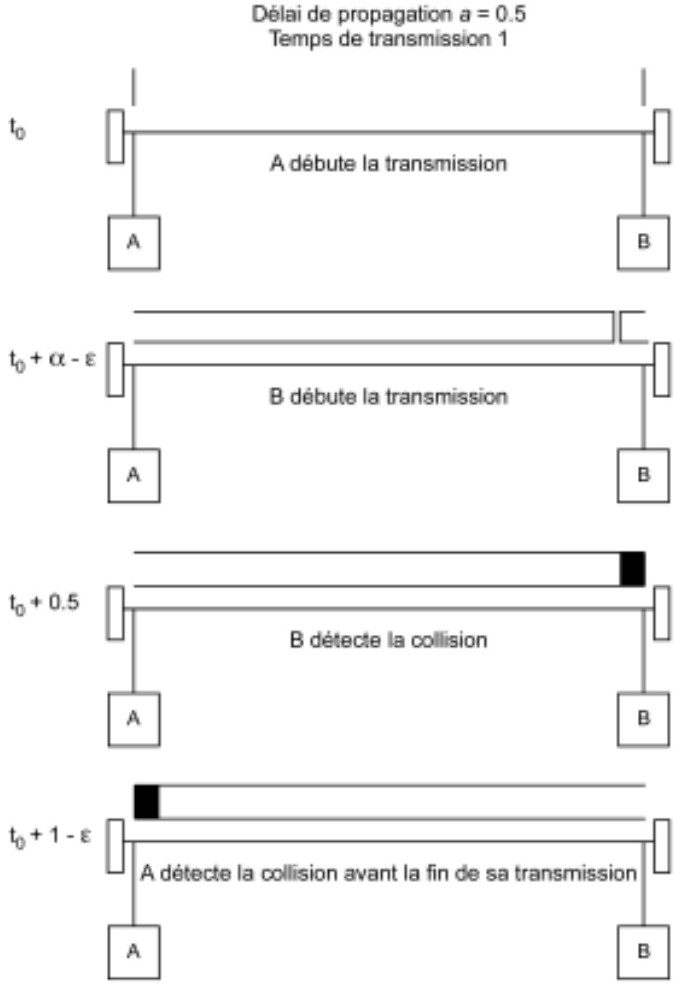


Figure 4.3 Représentation type de propagation de la collision

4.5 Calcul de la longueur maximale du segment

La fenêtre temps, la plus petite, permet de faire deux choses bien distinctes : la détection et l'établissement d'un rapport de collision. Elle est le facteur qui détermine la longueur minimale de la trame. Naturellement, la vitesse de transmission ainsi que l'étendue du LAN sont aussi des facteurs qui y sont directement reliés.

En règle générale, les réseaux à compétition doivent avoir la longueur de la plus petite trame correspondant à la transmission continue, laquelle est égale à deux fois le temps de propagation d'un bout à l'autre du segment. Si la vitesse de transmission R (en bits par seconde) et la longueur de la plus petite trame ℓ (en bits) sont connues, le temps de transmission de la trame T est égal à $T = \ell/R$.

Dans le cas d'un réseau CSMA/CD dont la vitesse est R = 10 Mbps, la longueur minimale de la trame sera : ℓ = 64 octets, c'est-à-dire 512 bits. Chaque bit a une durée de $1/10^7$ sec, ou 0.1µsec/bit. Par conséquent, la durée de 512 bits sera T = 0.1µsec/bit x 512 bits = 51.2 µsec. Ceci signifie tout simplement qu'une station transmettant dans un réseau de 10 Mbps doit au moins prendre 51.2 µsec pour transmettre sa plus petite trame (64 octets) afin de permettre, en cas de collision avec la station la plus éloignée, de recevoir le signal résultant de cette collision survenue juste au moment où cette dernière amorçait sa transmission.

Puisque le temps T détermine le temps de transmission de la plus petite trame, il définira le temps de propagation aller-retour à travers le segment à ε prés. La longueur du segment peut donc être calculée par la formule suivante :

$L = a\,T/2$,

où *a* représente la vitesse de propagation à travers le câble.

Pour suivre notre exemple, la longueur du plus grand segment acceptable a alors une durée de propagation de 25.6 µsec (51.2/2). Si la vitesse de propagation *a* dans le médium utilisé est de 0.66c (où c est la vitesse de la lumière = 300,000 km/sec) ou 198,000 km/sec, ce segment doit avoir une longueur maximale de L = 5068 mètres ($198 \times 10^6 \times 25.6 \times 10^{-6}$).

Il va de soi qu'une valeur bien inférieure doit être utilisée pour assurer les variations de la vitesse de propagation à travers le médium. La distance maximale a été établie à 2500 mètres.

4.6 Fonctions de la sous-couche MAC

La sous-couche MAC comprend des fonctions qui permettent de préparer les MAC-PDU ou trames en effectuant l'encapsulation au moment de la transmission des données. Ces fonctions sont illustrées à la figure 4.4. Comme nous l'avons vu dans le modèle OSI et tel qu'illustré à la figure 4.5, la trame est encapsulée avec un PCI (*Protocol Control Information*/Informations de contrôle du protocole).

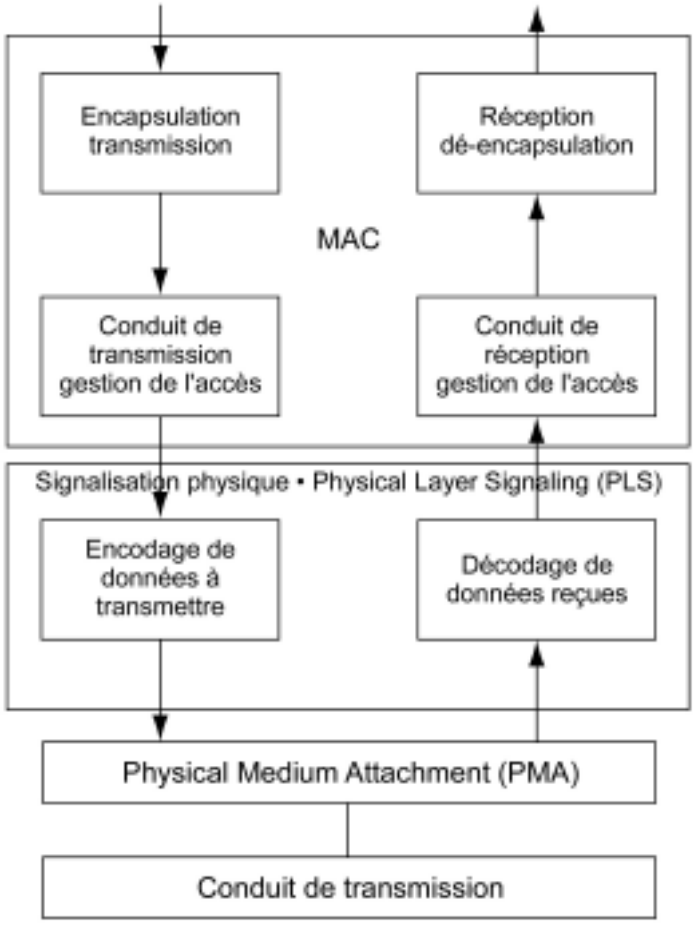


Figure 4.4 Représentation des fonctions de la sous-couche MAC

Figure 4.5 Représentation d'un MAC-PDU

Le PCI est utilisé pour :

- synchroniser la station de destination avec le signal;
- délimiter le début et la fin de la trame MAC;
- acheminer les adresses MAC de la source à la destination;
- détecter les erreurs de transmission.

4.6.1 Medium Access Management (MAM)

Ce module est responsable du contrôle de la transmission physique de la trame MAC.

À travers la sous-couche de signalisation physique PLS (*Physical Layer Signaling*), le MAM (*Medium Access* Management) tente de savoir si le conduit de transmission (*carrier*) est libre ou non. Si ce dernier l'est, il donne le contrôle au MAC (en attendant une durée précise après l'état occupé). Durant la transmission de la trame MAC au PLS, le MAM continue de surveiller le conduit de transmission (*carrier*). S'il y a collision avec une autre station qui transmet en même temps, le MAM interrompt la transmission des données et envoie à la place un signal de brouillage. Durant la réception, le MAM valide la trame MAC avant d'entamer la fonction dé-encapsulation.

4.6.2 Les trames

La trame CSMA/CD est composée de plusieurs champs (figure 4.6). Le premier est le préambule. Il comprend sept octets de valeur binaire 01010101. Le préambule est surtout utile pour permettre à la station réceptrice de se synchroniser avant la réception des données.

L'octet suivant, le SFD (*Start Frame Delimiter*), est utilisé pour indiquer le début de la trame. Son code binaire est 10101011.

Le champ suivant comprend les adresses MAC source et destination. Ce champ comprend 12 octets dont les six premiers indiquent l'adresse MAC de destination et les six octets suivants, l'adresse MAC de la source.

Il existe certaines variantes de format Ethernet. Par contre dans tous les cas, les champs précédemment décrits demeurent inchangés.

Les formats couramment utilisés sont :

Le CSMA/CD normalisé par le IEEE et désigné par le format 802.3;
le IEEE 802.3 avec le SNAP (*SubNet Access Protocol*);
le format Ethernet v2 (version 2 ARPA).

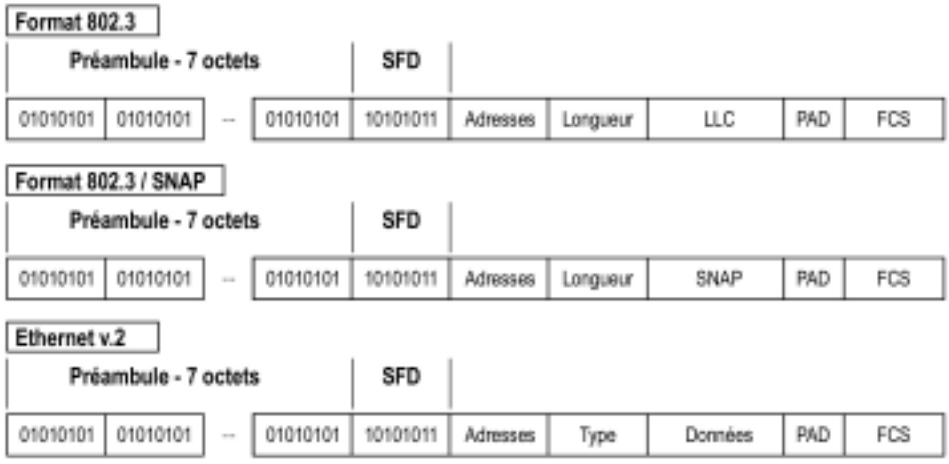


Figure 4.6 Représentation des catégories de trame Ethernet

Tout système peut exploiter aussi bien le format Ethernet v2 que le format IEEE 802.3. La seule différence réside dans le champ qui suit le champ adresse qui en est un de catégorie « type » pour la trame Ethernet et de « longueur » pour les trames 802.3.

Si la valeur de ce champ se situe entre 0x0000 et 0x05DC (1500 décimales), cette valeur indique la présence d'un champ « longueur » et le récepteur en déduit que son format est un IEEE 802.3. Par contre, si cette valeur est plus élevée, cela indique que l'on est en présence d'un champ « type » et que la trame transmise est celle d'Ethernet v2. Certaines valeurs du champ « type » sont réservées à Ethernet pour indiquer la présence d'un protocole de niveau supérieur dans le champ des données. Le tableau 4.1 fait état des valeurs les plus couramment admises.

Le format SNAP (*Sub Net Access Protocol*), illustré dans la figure 4.7, est parfois utilisé comme un substitut du LLC en codant les adresses LLC à des valeurs AA hex et en codant le champ « contrôle » à la valeur 03. Le protocole SNAP s'avère utile lorsque les paquets de la couche supérieure ont besoin d'un champ « type » ou code de protocole afin d'identifier l'information dans ses paquets. Les trois octets qui suivent le champ « contrôle » pourraient contenir le code du vendeur (*Organizational Unit Identifier* - OUI). Les deux octets suivant contiennent le code du protocole de la couche réseau. Lorsque le champ OUI est à 0, le code du protocole est le même que celui utilisé par les trames Ethernet dans le champ type.

Le protocole SNAP résout des problèmes d'alignement. En utilisant un en-tête de huit octets (64 bits), l'alignement des informations est respecté pour des mots de 32 bits.

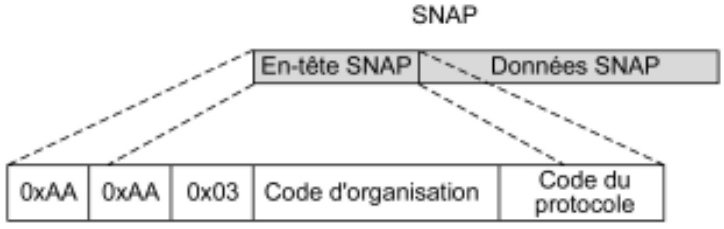


Figure 4.7 Représentation type du format du champ de données pour la trame 802.3/SNAP

Pour un réseau TCP/IP, le OUI est à 0. Les deux octets qui suivent contiennent la valeur Ethertype 0x800. Ceci permet de connecter un LAN qui n'est pas du format IEEE 802 à un LAN IEEE 802 tout comme l'Ethernet par exemple. La figure 4.8 nous montre un exemple d'interconnexion.

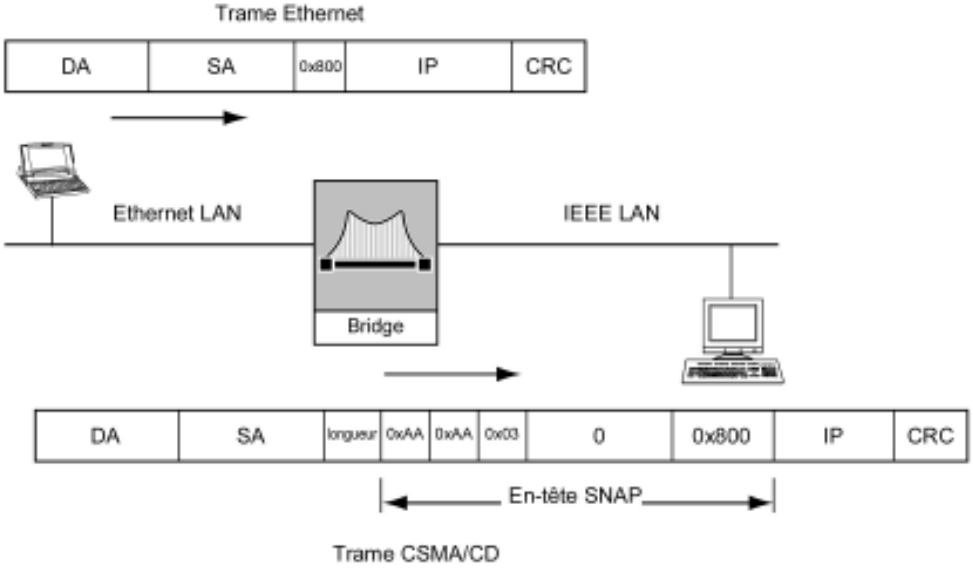


Figure 4.8 Représentation type de l'interconnexion du LAN Ethernet au CSMA/CD

La description du protocole SNAP figure dans le RFC 1042.

Tableau 4.1 Type de protocole utilisé

Protocole figurant dans la couche supérieure	Valeur hexadécimale
IP	0800
ARP	0806
Novell IPX	8137
802.3	0000-05DC

4.6.3 Adresse MAC

Chacun des terminaux qui a accès au réseau Ethernet ou au CSMA/CD est doté d'une carte ayant une adresse MAC exclusive. L'adresse MAC est celle de la couche MAC. Elle comporte 48 bits (six octets) et est contenue dans le module. Elle permet d'identifier la station et lui est exclusive.

Les trois premiers octets de l'adresse MAC renferment le code du manufacturier de la carte d'accès. Tout comme nous en avons déjà fait mention, ce code de manufacturier est connu sous le nom de OUI (*Organization Unique Identifier).* Les codes sont assignés par le IEEE. Les trois autres octets de l'adresse MAC relèvent du contrôle du manufacturier de la carte d'accès qui leur assigne une valeur exclusive.

Exemple d'adresse MAC : **00-60-97-8F-4F-86**

Dans ce cas, la valeur 00-60-97 est un OUI assigné à CISCO. Les trois autres octets 8F-4F-86 ont été assignés par CISCO pour une carte d'accès donnée. Les chiffres de cet OUI ne seront plus assignés par CISCO à toute autre carte d'accès.

4.6.4 Trame de diffusion individuelle (Unicast)

Les adresses MAC servent à identifier les stations source et destination. Les deux adresses se trouvent dans la trame de diffusion individuelle (*Unicast),* ainsi qualifiée, parce que la communication se fait point à point. La trame est en effet reçue par toutes les stations, mais la couche MAC de chacune d'entre elles traite une copie de la trame et s'assure que l'adresse destination lui correspond. Si tel est le cas, elle envoie une copie de la trame à la couche supérieure. Dans le cas contraire, elle n'assume rien. Le fait qu'une trame soit traitée au niveau de la couche MAC plutôt qu'à celui d'une couche supérieure, épargne des cycles CPU à la station qui peut les utiliser à d'autres fins.

4.6.5 Trame de diffusion générale (Broadcast)

Il existe une adresse qui permet à toute station de transmettre les informations vers toutes les autres stations. Cette adresse destination est l'adresse MAC de diffusion générale (*broadcast*) FF-FF-FF-FF-FF-FF. Lorsque l'interface reçoit la trame avec pour adresse destination l'adresse de diffusion générale, elle accepte la trame

4.6.6 Trame de multidistribution (Multicast)

Les trames multidistribution (*multicast*) diffèrent des trames de diffusion générale. Dans le cas des trames *multicast,* l'adresse destination est une adresse de groupe et non de toutes les stations. Ces groupes pourraient avoir des intérêts communs. Seules les stations appartenant à ce groupe acceptent la trame.

Par exemple, Cisco utilise une adresse multidistribution pour donner aux équipements réseau de Cisco des informations à travers le protocole CDP (*Cisco Discovery Protocol*). Dans ce cas, l'adresse *multicast* utilisée par Cisco est 01-00-0C-CC-CC-CC. Seuls les équipements Cisco reconnaîtront cette adresse de groupe et accepteront ces informations.

4.7 Taux de transmission paquets par seconde

Il faut bien comprendre que plus le nombre de trames transmises est élevé, plus grandes sont les possibilités d'avoir une collision. Il serait bon de calculer la charge moyenne possible que le réseau devrait pouvoir endurer en vue d'offrir un service acceptable.

En d'autres termes, il s'agit de calculer le nombre de paquets par seconde des trames de longueurs différentes.

Le nombre de paquets par seconde est défini comme suit :

Paquet/sec (PPS) = 1 / (IFG + temps préambule + temps trame)

Pour un réseau à 10 Mbps, IFG (*Inter Frame Gap*) est le temps d'attente entre la transmission de chaque trame. Il est de 9.6 µsec.

Pour ce qui est du temps préambule : c'est le temps de transmission des 8 octets du préambule et du SFD (*Start Frame Delimiter*/Délimiteur de début de trame*)* soit 64 bits, ce qui représente 6.4 µsec.

En ce qui a trait au temps trame : c'est le temps de transmission d'une trame comportant X octets. Comme il a été mentionné précédemment, le temps de transmission de la trame T est égal à $T = \ell/R$. Ainsi, lorsque ℓ est de 64 octets donc 512 bits, et que R est 10Mbps, T est égale à 51.2 µsec.

Le PPS d'une trame de 64 octets est :
1 / (9.6 + 6.4 + 51.2) µsec = 14,880 PPS.

Pour une trame maximale de 1518 octets, le PPS est :
1 / (9.6 + 6.4 + 1214.4) µsec = 812 PPS.
Plus la trame est grande plus petit est le nombre de PPS.

4.7.1 La couche physique

Lorsque la norme IEEE802.3 a été conçue et admise, seuls les protocoles de la sous-couche MAC ainsi que ceux de la couche physique du CSMA/CD ont été inclus dans cette norme. La sous-couche LLC IEEE802.2, étant commune à tous les protocoles de réseaux locaux, a été élaborée séparément.

La couche physique du 802.3 a été scindée en deux sous-couches : la sous-couche de signalisation physique (PLS) et la sous-couche de médium physique de rattachement. La figure 4.9 nous en montre leur superposition.

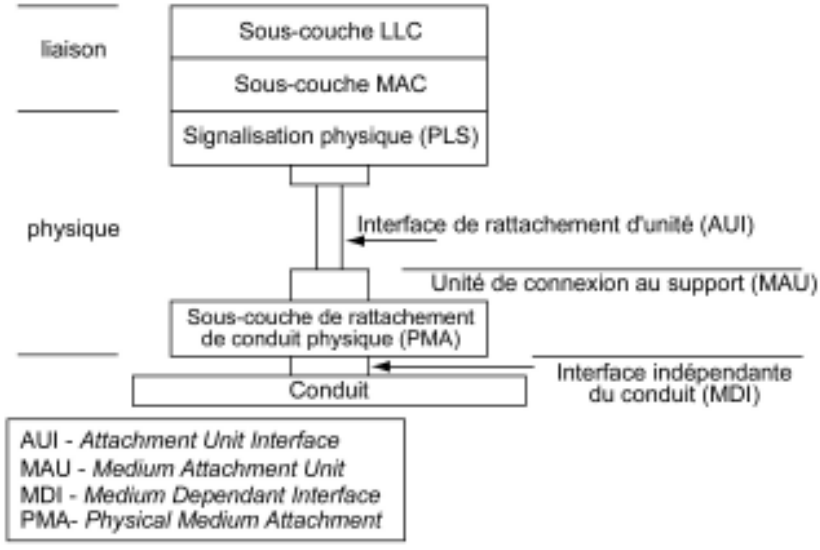


Figure 4.9 Représentation type des sous-couches d'un LAN

Il existe différentes options de montage physique pour le CSMA/CD. Quoique dans tous les cas le protocole MAC agisse de la même façon, ces différences nous donnent une indication quant au comportement du réseau. La notation utilisée pour identifier ces configurations physiques est établie comme suit :

<vitesse en Mbps> <méthode de signalisation> <longueur maximale du segment en centaines de mètres>

Quelques configurations sont :

10BASE-5 - 10 Mbps, bande de base (*Baseband)*, 500 mètres
10BASE-2 - 10 Mbps, bande de base *(Baseband)*, 180 mètres
10BASE-T - 10 Mbps, bande de base *(Baseband)*, paire torsadée *(Twisted pair*)

4.7.2 Sous-couche de signalisation physique (PLS)

La sous-couche de signalisation physique est la sous-couche supérieure à la couche physique. Elle est interfacée à la sous-couche MAC. La figure 4.10 en donne une illustration.

Tout MAC PDU prêt à être transmis est encodé par le PLS avant d'être acheminé le long du conduit de transmission. Avant de procéder à la transmission des données, le PLS s'assure qu'il n'y a pas de bruit sur le canal. L'encodage dépend du système de transmission. Dans le cas d'une transmission à 10 Mbps, les données sont encodées selon le codage Manchester. Lorsque la transmission a lieu d'après la méthode 100BaseTX, l'encodage se fait selon la méthode 4B/5B. D'autres méthodes d'encodage existent et sont appliquées à d'autres systèmes de transmission. Les méthodes de codage Manchester et 4B/5B seront traitées plus loin dans ce volume.

Le PLS assume également la réception des données et leur décodage en signal binaire. Une fois le décodage accompli, les données sont remises à la sous-couche MAC.

Avec le PMA (*Physical Medium Attachment*/Raccordement au support physique*)*, le PLS fait l'écoute constante de la ligne de transmission. Il détecte les collisions et prévient la sous-couche MAC de l'état du conduit à savoir s'il est libre, s'il y a transmission ou bien s'il y a détection d'une collision.

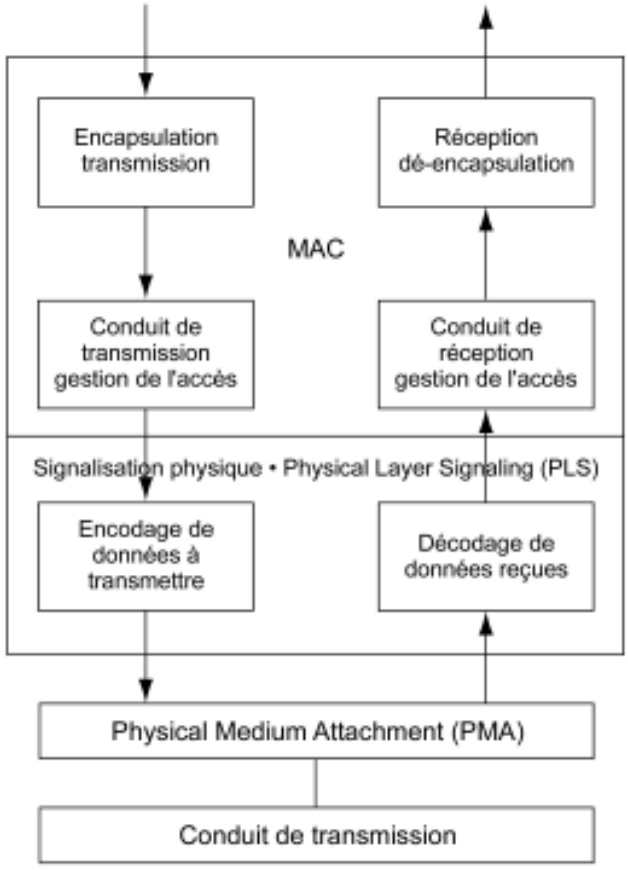


Figure 4.10 Représentation type des fonctionnalités du 802.3 avec la couche physique

4.7.3 Codage Manchester

La technique d'encodage utilisée dans le CSMA/CD est connue sous le nom de codage Manchester. Le signal est à 20 MHz. Les valeurs binaires sont reconnues par la transition du signal au centre de l'emplacement du bit. Le 0 binaire est le signal qui transite au milieu de l'espace d'un bit passant du niveau positif au niveau négatif; tandis que le 1 binaire correspond au signal qui transite au milieu de l'espace d'un bit passant du niveau négatif au niveau positif. La figure 4.11 illustre ces transitions et nous donne un exemple d'encodage de bits. Les transitions survenant au début de l'emplacement du bit ne sont pas prises en compte et elles ne sont faites que pour établir la position du signal d'une nouvelle transition au milieu de l'espace d'un bit.

Il est incontestable qu'une synchronisation de bits doit se faire au préalable en vue de déterminer l'emplacement du début de chaque bit et pouvoir ainsi en connaître son milieu. Nous comprenons maintenant pourquoi nous avons besoin du préambule au début de chaque trame. En effet, celui-ci permet à la couche physique de se synchroniser au niveau des bits.

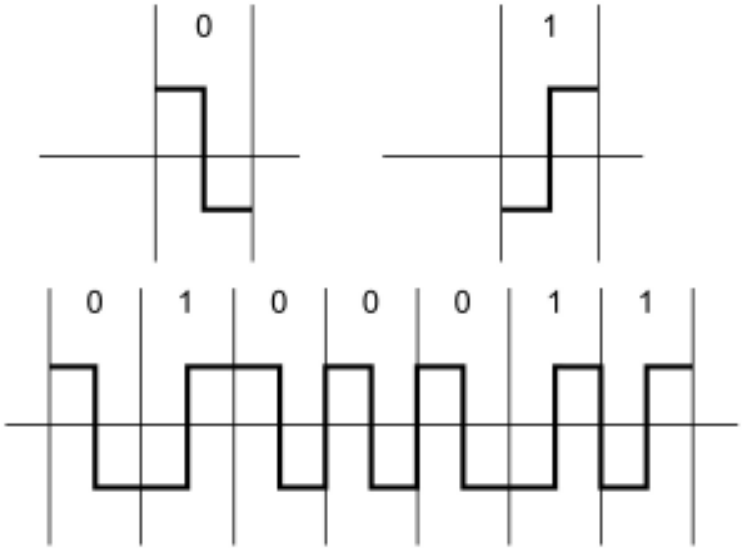


Figure 4.11 Représentation type du codage Manchester

En vue de pouvoir transmettre à 10 Mbps, le signal se trouve à 20 MHz. Deux cycles s'avèrent nécessaires pour chaque bit. Ceci permettra de procéder à la transition requise au début et au milieu de chaque bit.

4.7.4 Fonction de l'unité de connexion au support (MAU)

L'unité de connexion au support connue sous l'acronyme MAU (*Medium Attachment Unit*) peut se trouver en mode normal ou en mode surveillance. En mode normal, le MAU sert de connexion entre le conduit physique et les couches supérieures du terminal. Les données peuvent ainsi passer du conduit aux couches supérieures et vice-versa.

Le mode surveillance ne permet pas de procéder au transfert des données des couches supérieures au conduit. Par contre, la réception des données par le conduit et leurs transferts aux couches supérieures ainsi que la détection de collision sont activés.

Les fonctions de la sous-couche physique, unité de rattachement de conduit (MAU), sont :

- **Transmission** en série d'un train de bits sur un conduit de transmission.
- **Réception** en série d'un train de bits sur un conduit de transmission.
- **Présence de collision** détecte la présence de deux ou plusieurs stations qui transmettent simultanément.
- ***Monitor Inhibit*** inhibe la fonction de transmission lorsque les fonctions ***réception*** et ***présence de collision*** restent opérationnelles.
- ***Jabber Interrupt*** interruption automatique de la transmission.

4.7.5 Types de conduit de transmission

Plusieurs types de câbles sont utilisés pour former un segment Ethernet / CSMA/CD. Le câble coaxial en a été l'un des pionniers. On distingue plusieurs catégories de câbles coaxiaux. La première signalisation en bande de base sur câble coaxial a été faite sur un câble de 50 Ohms et de 10 mm d'épaisseur (*Thick Ethernet*). Ce câble servait à la configuration du 10BASE-5, mais il était trop volumineux et trop rigide. Aussi, un câble bien plus souple et malléable de 50 Ohms et de 5 mm d'épaisseur (*Thin Ethernet*) l'a remplacé. Il est utilisé dans la configuration du 10BASE-2. Pour sa part, la fibre optique permet d'atteindre des vitesses très élevées. Le plus populaire des conduits est la paire torsadée du 10BASE-T. Avec le temps et l'évolution des technologies et des techniques, le conduit s'est adapté à toutes les situations données et a évolué connaissant un grand essor au plan des transmissions.

4.8 Fast Ethernet

De nos jours, l'évolution des modules d'accès nous permet de transporter et d'acheminer des informations sur le réseau Ethernet / CSMA/CD à la vitesse de100 Mbps. C'est l'avènement du Fast Ethernet. Cette nouvelle technologie est des plus attrayantes et permet à un même réseau de supporter un 10 Mbps aussi bien qu'un 100 Mbps. Sur le plan opérationnel, le protocole MAC demeure inchangé. Par contre, l'interface d'accès change.

La norme IEEE qui décrit les aspects propres au 100 Mbps est connue sous le nom de la norme IEEE802.3u.

Le Fast Ethernet est dix fois plus rapide que l'Ethernet standard. De ce fait, le facteur temps s'en trouve réduit d'autant. La durée de la trame minimale de 64 octets (512 bits) est de 51.2 μsec dans le cas d'un accès à 10 Mbps. Dans le cas

du Fast Ethernet à 100 Mbps, cette durée passe à 5.12 µsec et le IFG (*Inter Frame Gap*) à 0.96 µsec. Cette vitesse a pour effet de réduire aussi le diamètre du réseau. Dans les deux cas mentionnés ci-dessus, le format de la trame MAC demeure inchangé. Par contre, l'encodage est différent. À une vitesse de 100 Mbps, il n'est plus possible d'avoir recours au codage Manchester sur du câble à paires torsadées. Aussi, en raison de cet essor, de nouvelles méthodes d'encodage ont vu le jour. De plus, le mode de transmission duplex intégral ou semi-duplex peut aussi avoir un effet sur la vitesse de transmission.

4.8.1 Modes duplex intégral (Full-Duplex) et semi-duplex (Half-Duplex)

Le CSMA/CD 10 Mbps fonctionne par définition en semi-duplex (*Half-Duplex).* Nous avons déjà vu qu'il ne peut y avoir qu'une seule transmission à la fois sur le support physique. Dans le cas contraire, une collision survient. De plus, la vitesse du segment n'est que de 10 Mbps et le débit est partagé entre les utilisateurs du réseau. Dans le cas où seulement deux utilisateurs veulent utiliser le réseau à pleine capacité, ils auront un débit d'un peu moins que 5 Mbps chacun. La situation s'aggrave lorsque le nombre d'utilisateurs augmente. Ce problème devient important surtout lorsqu'on considère que le système 10BASE-5 ou celui du 10BASE-2 utilise un câble coaxial auquel toutes les stations sont branchées. Même dans le cas du 10BASE-T où chaque station est directement connectée à un concentrateur (*Hub)*, il y a risque de collision car les *Hubs* eux-mêmes n'opèrent qu'en mode bidirectionnel à l'alternat. Ces concentrateurs (*Hubs)* peuvent être interconnectés entre eux par un câble coaxial. Pour cette raison, ils doivent opérer en mode semi-duplex. Dans des cas particuliers, il est possible d'offrir le duplex intégral avec le 10BASE-T.

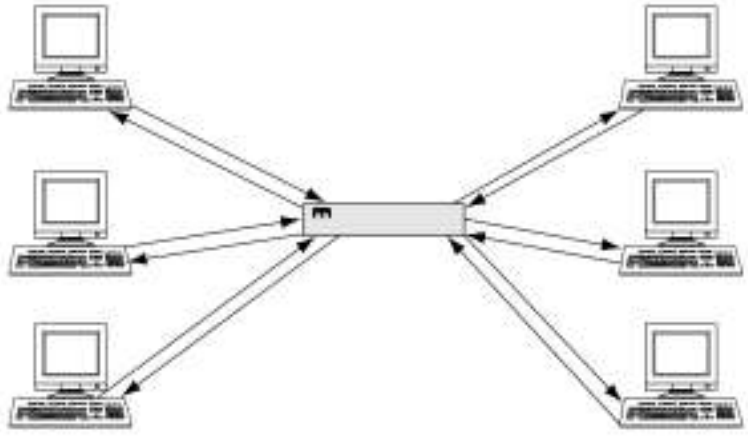


Figure 4.12 Représentation d'une connexion en mode duplex intégral

Dans le cas du 100BASE-X, les stations sont toutes connectées à un commutateur de 100 Mbps avec un lien point à point. Ces commutateurs sont eux-mêmes interconnectés entre eux par des fils à paires torsadées ou par des fibres optiques et non par câbles coaxiaux. Ce genre de branchement permet alors d'agir en mode duplex intégral. Pour opérer dans ce mode, il est primordial qu'il n'y ait pas de collisions. Les stations sont reliées aux commutateurs par deux paires de liens, l'une pour la transmission et l'autre pour la réception. Ceci permet d'obtenir un débit de 100 Mbps tant à la transmission qu'à la réception, soit 200 Mbps. En d'autres termes, cela revient à dire que la transition d'un 10 Mbps à un 100 Mbps offre un service accru de 20 fois sa capacité si le mode duplex intégral est mis en pratique.

Les commutateurs (*switches*) comme le Catalyst de Cisco permettent aux stations d'établir une connexion de point à point à l'aide d'un câble de catégorie 5 en mode duplex intégral. Une série de connecteurs RJ-45 est disponible pour connecter un grand nombre de stations. Un exemple de connecteur disponible sur le Catalyst est illustré à la figure 4.13.

Figure 4.13 Représentation type des connecteurs femelles RJ-45 sur le Catalyst 5500 de Cisco

Le commutateur traite chaque connexion séparément sans possibilité aucune de collision. Par contre, il doit être assez rapide pour honorer tant les transmissions reçues que celles dont il doit assurer le relais.

4.8.2 La fonction pause

Le comité du IEEE 802.3x s'est penché sur la question du mode duplex intégral. La norme IEEE802.3x offre aussi l'option d'un dispositif de contrôle du flux au niveau du MAC. Si cette option est acceptée par les deux parties, cela permettra au terminal récepteur de contrôler le flux d'informations qu'il reçoit en transmettant la trame pause. Un des champs contient une valeur indicatrice de la durée du temps de la pause en terme de nombre d'intervalles de temps (*time slots*), comme lorsque sa mémoire tampon est trop pleine. L'adresse destination (DA) est dans ce cas une adresse spécifique au mode pause, notamment 01-80-C2-00-00-01 en hexadécimal. Sur réception de cette trame, le transmetteur cesse toute transmission pendant ce laps de temps demandé. Si le récepteur récupère avant expiration du temps, il transmet alors une autre trame pause. La valeur du temps

de pause indiquée dans cette nouvelle trame est mise à 0. La trame pause est illustrée à la figure 4.14.

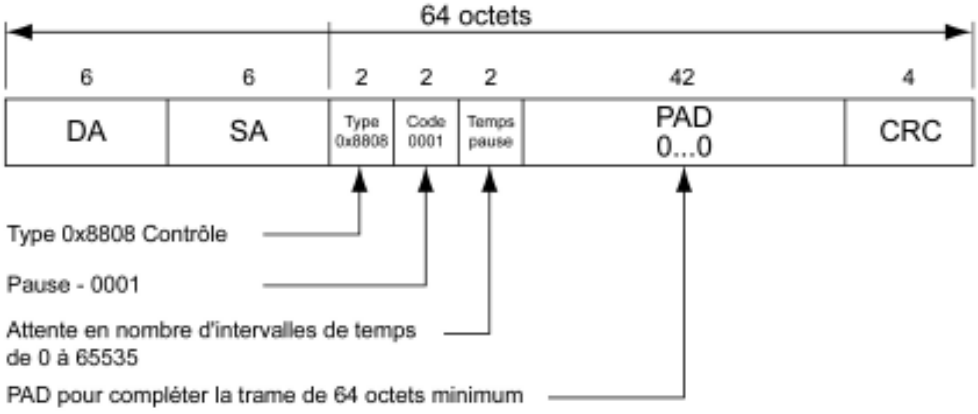


Figure 4.14 Représentation de la trame contrôle pause

Les trames pause n'opèrent que dans le mode duplex intégral.

Il est évident que lorsqu'une station de 100 Mbps est utilisée avec des concentrateurs (*Hubs)* en mode semi-duplex, elle ne peut opérer qu'en semi-duplex.

4.9 L'autonégociation

Lorsqu'il faut faire un choix de débit automatique entre la station et le commutateur et tenter d'obtenir le plus élevé, ou lorsqu'il faut automatiquement déterminer si le mode duplex intégral est possible, l'autonégociation est recommandée (cf. IEEE802.3 2000 section 28). Elle consiste en un échange de mot code (*Link Code Word*) mentionnant les modes préférés de part et d'autre. Le code donne aussi une indication des genres de configurations utilisées. Cela est rendu possible dans le cas du 10/100/1000BASE-T, 100BASE-TX, et 100BASE-T4.

Le LCW (*Link Code Word)* est inclus dans les rafales du FLP (*Fast Link Pulse*). Les premières rafales FLP sont faites lors de la mise sous tension des deux équipements connectés point à point. En moyenne, elles sont transmises toutes les 16 millisecondes.

Chaque rafale de FLP comprend 33 positions d'impulsions de durée moyenne de transmission de 2 millisecondes. Les 17 impulsions impaires sont des impulsions

représentant l'information d'horloge. Chacune de ces impulsions d'horloge est transmise toutes les 125 µsec.

Les 16 autres positions paires représentent le LCW. Une impulsion dans les positions du LCW indiquerait un 1 logique alors que l'absence d'impulsion désignerait un 0 logique. Chaque impulsion a une durée moyenne de 100 nanosecondes. Une partie du FLP est illustrée à la figure 4.15. Les bits d'horloge et les bits du LCW sont entrelacés.

La transmission des rafales FLP est périodique et n'entrave en rien le cours normal de l'exploitation du réseau. L'autonégociation n'a lieu que pour des liens point à point. Les fonctions de cette dernière sont : la réception, l'arbitrage et la transmission. L'interaction avec des équipements 10BASE-T doit toujours être possible. C'est pourquoi la séquence d'impulsions NLP (*Normal Link Pulse*) générée par le 10BASE-T doit également être admise pour pouvoir établir le moment précis où le début de chaque rafale sera généré.

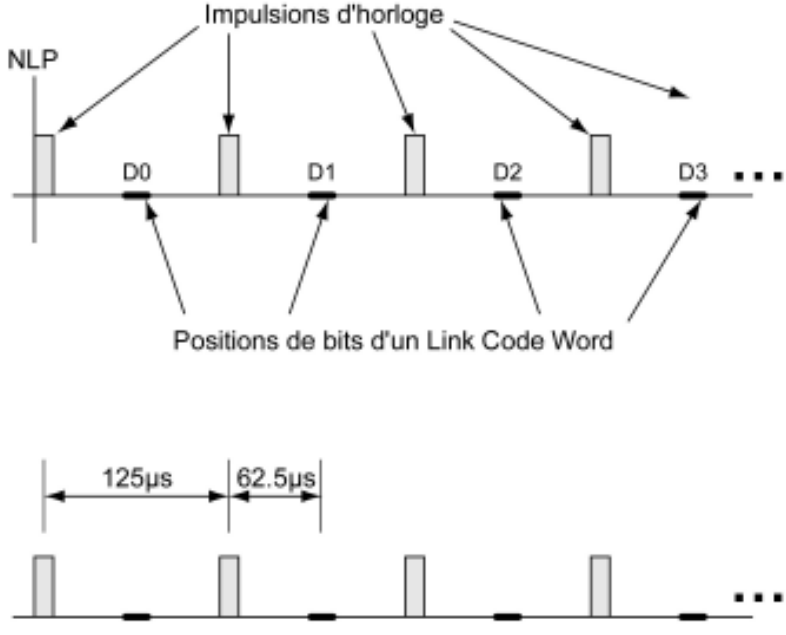


Figure 4.15 Représentation d'un FLP

Le codage d'un LCW comprend les champs suivants : un champ de sélecteur, un champ d'habileté technologique, un bit d'erreur distante, un bit d'accusé de réception et un bit signalant l'existence d'un autre mot ou « page » (*New Page bit*) avec d'autres informations. Le LCW est représenté à la figure 4.16.

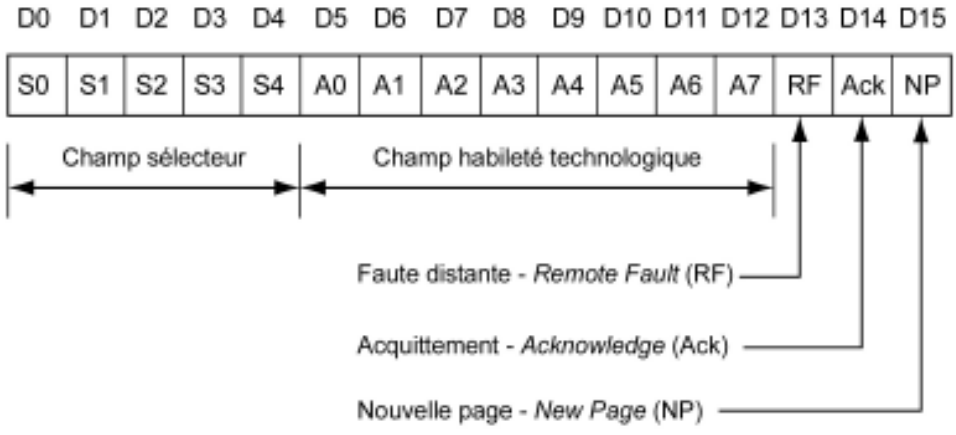


Figure 4.16 Représentation du LCW

Le champ sélecteur identifie le type de message transmis durant l'autonégociation. Deux codes sont spécifiés. Lorsque le champs sélecteur est codé à 1 (S0 S1 S2 S3 S4 = 10000), les autres champs sont interprétés selon la définition de la norme du IEEE 802.3.

Le champ habileté technologique comprend les huit bits suivants : A0 à A7. Les cinq premiers bits de ce champ, A0 à A4, indiquent le genre de technologie implémentée. Le tableau 4.2 précise les divers genres de technologies. Dans le cas où l'implémentation du 100BASE-TX avec deux paires de fils de la catégorie 5 est souhaitable, le bit A2 est à 1.

Tableau 4.2 Technologies implémentées

Bit	Technologie	Câblage
A0	10BASE-T	2 paires catégorie 3
A1	10BASE-T full duplex	2 paires catégorie 3
A2	100BASE-TX	2 paires catégorie 5
A3	100BASE-TX full duplex	2 paires catégorie 5
A4	100BASE-T4	4 paires catégorie 3

Les bits A5 et A6 sont utilisés pour signaler que le terminal a implémenté la fonction optionnelle de pause dont nous avons traité précédemment. Cela ne peut avoir lieu que si la configuration utilisée est en mode duplex intégral.

Le bit A5 qui correspond au bit pause indique que l'équipement est capable d'offrir la fonction pause symétrique. Le bit A6 ou bit ASM_DIR indique que l'équipement est en mesure d'assurer la fonction pause asymétrique. Dans le cas de la fonction pause symétrique, on signifie que l'équipement peut transmettre une trame pause et entrer en fonction dès sa réception.

Le bit erreur distante permet d'indiquer qu'il y a eu une simple erreur. Ce bit reflète l'état interne du registre.

Le bit accusé de réception est utilisé pour indiquer qu'un terminal a bien reçu le LCW.

Lors de l'autonégociation, les terminaux concernés peuvent recourir à plusieurs possibilités de configuration. Aussi a-t-on établi une priorité en la matière en regard de la configuration des technologies admises conformément aux valeurs du champ sélecteur du IEEE 802.3. En commençant par la plus élevée, la liste des priorités est la suivante :

-1000BASE-T Duplex intégral (*full duplex*)
-1000BASE-T
-100BASE-T2 Duplex intégral
-100BASE-TX Duplex intégral
-100BASE-T2
-100BASE-T4
-100BASE-TX
-100BASE-T Duplex intégral
-100BASE-T

Si le bit NP (*Next Page*) est à 1, cela signifie qu'un autre mot de 16 bits suit, comportant des informations supplémentaires.

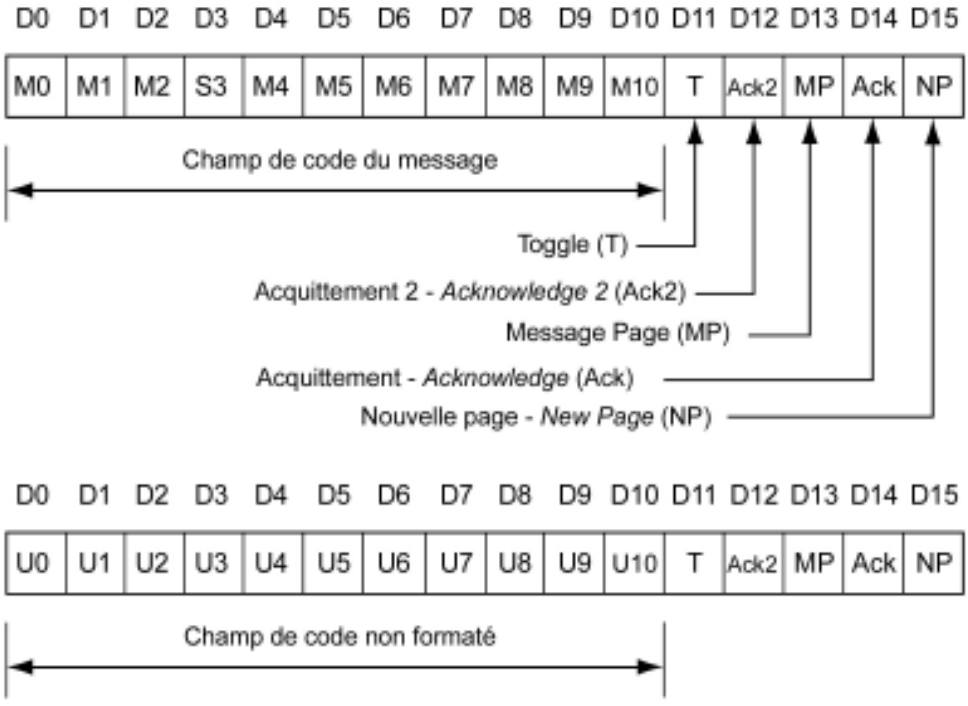


Figure 4.17 Codage des pages additionnelles avec message ou non formatées

La page additionnelle, telle qu'illustrée à la figure 4.17, peut avoir deux formats. Le format message est indiqué lorsqu'un code message prédéfini est utilisé. Il peut signaler le type d'erreur ou le code d'organisation de l'équipement (OUI) ou encore l'expansion du code relatif à l'habileté technologique.

Le deuxième format de la page additionnelle est non formaté. Les 11 bits U0 à U10 renferment des valeurs arbitraires.

Le bit MP décide du format de la page additionnelle. Lorsqu'il est codé à 0, cela signifie que la page est non formatée. Par contre, lorsqu'il est codé à 1, on se trouve en présence du format message.

Le bit clé *Toggle* assure la synchronisation d'échange de pages additionnelles entre les terminaux concernés.

L'accusé de réception 2 (Ack2) démontre que l'équipement se conforme au message lorsqu'il est mis à 1.

Les bits *accusé de réception* et de nouvelle page ont la même définition que dans le cas du LCW traité précédemment.

4.10 Configurations à 100 Mbps

4.10.1 Les codages du 100 Mbps

Un câble de catégorie 5 est utilisé avec le 100BASE-TX. Ce câble se compose de paires torsadées très tressées capables de répondre à un signal de 100 MHz. Les paires torsadées sont aussi connues sous le nom de UTP (*Unshielded Twisted Pairs*/Paire torsadée non blindée).

La paire composée de fils 1 et 2 est utilisée aux fins de transmission alors que la paire de fils 3 et 6 est réservée aux fins de réception et de détection de collision. Les segments du câble ont une longueur maximale de 100 mètres.

Le codage utilisé pour le 100BASE-TX est le 4B/5B NRZI (Non retour à zéro avec inversion des 1 binaires). Tout comme l'illustre la figure 4.18, les 16 symboles de données (*data symbols*) ne pourront jamais avoir plus de trois 0 successifs s'ils sont ordonnés les uns après les autres. De la sorte, ce code permet de conserver la synchronisation lors de la transmission de trois bits de données.

Lorsque le conduit est une paire de fibre optique, le codage est encore le 4B/5B NRZI[1]. Il permet la transmission dans la fibre.

[1] Pour une explication du NRZI, voir l'annexe en fin de chapitre.

CODE Group	Symbol	Assignment
Line state symbols		
00000	Q	Quiet
11111	I	Idle
00100	H	Halt
Starting delimiter		
11000	J	ND
10001	K	ND
Data Symbols		
11110	0	0000
01001	1	0001
10100	2	0010
10101	3	0011
01010	4	0100
01011	5	0101
01110	6	0110
01111	7	0111
10010	8	1000
10011	9	1001
10110	A	1010
10111	B	1011

CODE Group	Symbol	Assignment
11010	C	1100
11011	D	1101
11100	E	1110
11101	F	1111
Ending delimiter		
01101	T	
Control Indicators		
00111	R	logical 0 (reset)
11001	S	logical 1 (set)
Invalid Code Assignments		
00001	V or H	Violation or Halt
00010	V or H	Violation or Halt
00011		
00101		
01110	Violations	
01000		
01100		
10000		

Figure 4.18 Représentation de la table des codes du 4B/5B

Lorsque le conduit d'un 100BASE-TX se compose de paires torsadées, le codage 4B/5B NRZI est inefficace. Il est d'abord converti en NRZ. Le nouveau code est divisé par un polynôme connu (étape de brouillage). Le résultat obtenu est codé par le MLT-3 et la transmission suit.

L'encodage MLT-3 comporte trois paliers : voltage positif V+, pas de voltage V_0 et voltage négatif V-. L'encodage est basé sur le diagramme d'état de la figure 4.19. La transition d'un palier à l'autre (ou d'un état à l'autre) ne se fait que lorsque le 1 binaire est reçu. Dans le cas où le 0 binaire est reçu, il n'y a pas de changement de palier. L'encodage MLT-3 est surtout utilisé en raison d'économie d'énergie.

Un exemple d'encodage du MLT-3 est représenté à la figure 4.20.

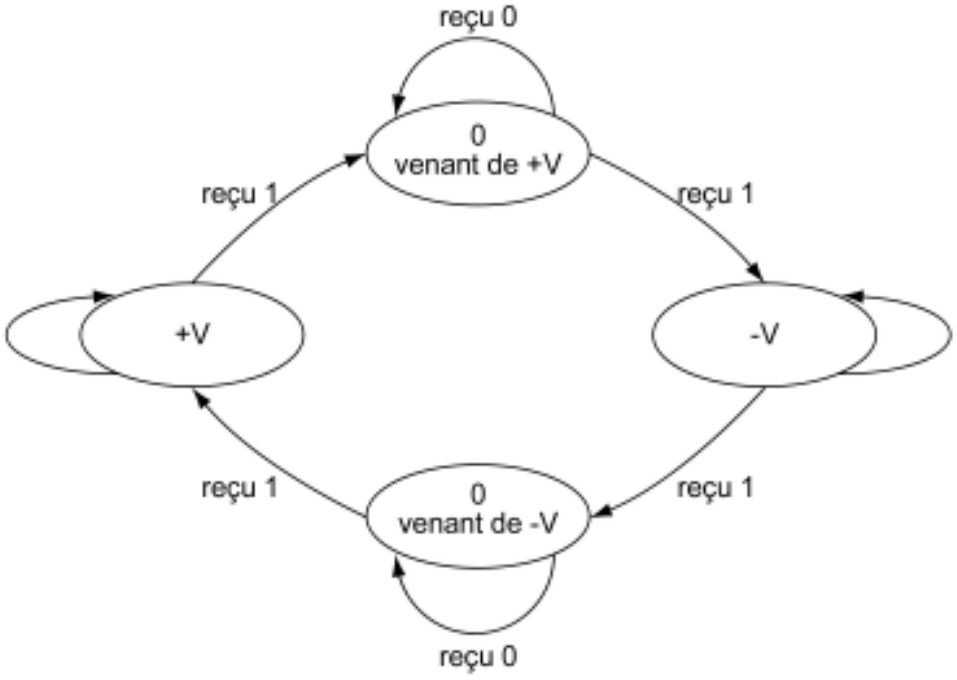


Figure 4.19 Représentation du diagramme d'état de l'encodage MLT-3

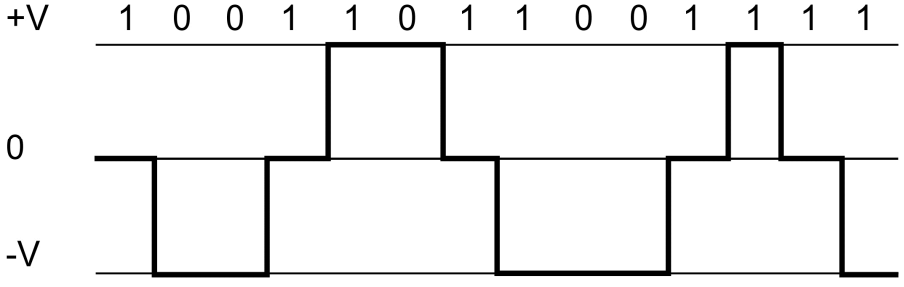


Figure 4.20 Exemple d'encodage MLT-3

Le 100BASE-T4 est utilisé sur du câble de qualité moindre. Cependant, on le retrouve dans la plupart des anciennes installations téléphoniques. Ce câble est fait avec des paires torsadées de catégorie 3 et peuvent répondre à un signal de 16 MHz. Dans ce cas, il faut avoir recours à trois paires pour la transmission et à une pour la détection de collision. Il va de soi que le 100BASE-T4 n'est utilisé qu'en mode semi-duplex. L'encodage utilisé est le 8B/6T (8bits/6 ternary signals à trois paliers, soit : +1, 0 et -1). Le segment a une longueur maximale de 100 mètres.

Le 100BASE-T2 a été conçu en 1997 en vue de réduire le nombre de paires utilisées lors de la transmission par le 100BASE-T4. La norme IEEE802.3 y donne une description du 100BASE-T2. Il peut être utilisé avec des câbles de catégorie 3, 4, et 5. Il ne se sert que d'une paire pour la transmission et réception et d'une pour la détection de collision. Le 100BASE-T2 est fondé sur une technologie avancée du traitement des signaux et de la méthode d'encodage PAM 5x5 (4 bits pour une valeur sur 25 possibilités) indiquée à la figure 4.21. Deux signaux PAM (*Pulse Amplitude Modulation*/Modulation d'impulsions en amplitude) à cinq niveaux sont transmis sur les paires torsadées. Le taux de signalisation est de 12.5 MHz. Pour les deux signaux, il y a donc 25 millions de changements de niveaux par paire et par seconde. Chaque paire de signaux PAM A et B encode 4 bits. Un simple calcul nous donne : 25 MHz x 4 bits = 100 Mbps. Le segment a aussi une longueur maximale de 100 mètres.

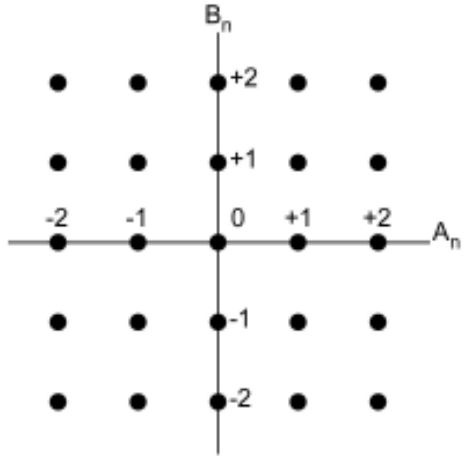


Figure 4.21 Représentation de la constellation de symboles PAM 5x5

Le 100BASE-FX utilise une paire de fibres optiques. Une fibre est réservée pour la transmission alors que la seconde sert à la réception. Le câble peut être en mode simple ou en multimode. Tout comme dans le cas du 100BASE-TX, l'encodage utilisé est le 4B/5B. Cependant, l'avantage du 100BASE-FX repose sur la longueur du segment. En effet, lorsque la fibre multimode est utilisée, le segment peut atteindre une longueur maximale de 2000 mètres en mode duplex intégral et de 400 mètres en mode semi-duplex. Lorsque la fibre est utilisée en mode simple (*single-mode*), la longueur du segment peut atteindre jusqu'à 10 000 mètres.

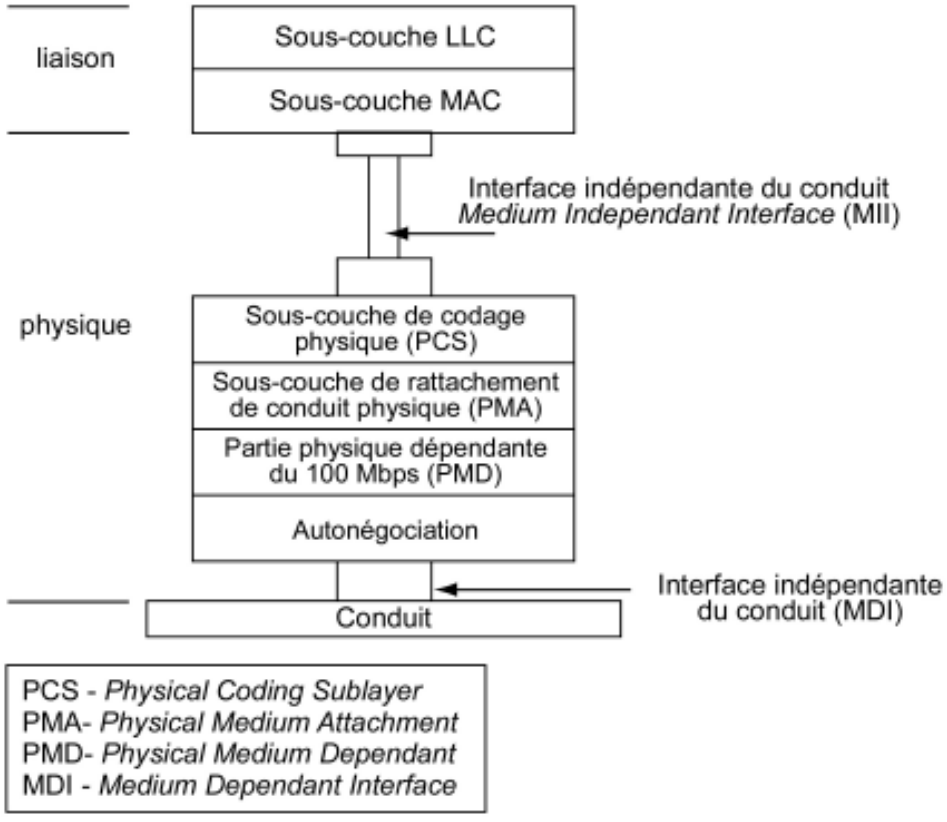


Figure 4.22 Représentation de la couche physique

La figure 4.22 illustre l'interface physique dans une configuration à 100 Mbps. Le MDI (*Medium Dependant Interface*/Interface dépendant du support) est, dans le cas de paires torsadées, le connecteur à 8 broches RJ45.

4.10.2 Media–Independent Interface (MII)

Le MII (*Media-Independent Interface*) Interface indépendante du conduit est un connecteur qui permet d'adapter la carte Fast Ethernet à n'importe laquelle des interfaces de la couche physique mentionnées ci-dessus. Le MII comprend 40 broches (*pins*). Il s'adapte à l'interface de 100 Mbps. Si la connexion se fait sur un 100BASE-T4, un MII spécialement conçu pour cela sera requis. Par contre, si l'on décide de passer à un 100BASE-T2, seul le MII devra être changé. La carte Fast Ethernet demeurera inchangée.

4.11 Étendue du réseau Ethernet

Le réseau Ethernet traditionnel a des restrictions dans l'étendue de son opération. Par exemple, le 10BASE-2 ne peut avoir qu'un segment dont la longueur maximale atteindra 185 mètres. Les répéteurs seront utilisés pour accroître cette étendue, mais avec bien des restrictions. Le 10BASE-T a une limite de 100 mètres de câble. Toutefois, avec un répéteur, le 10BASE-T peut atteindre 100 mètres de plus.

Le problème de l'expansion du segment a été traité dans le cadre de la norme IEEE802.3u. En outre, et pour les systèmes 100BASE-X, deux classes de répéteurs ont été définies. Il s'agit du répéteur de classe I dont la latence est de 0.7 μsec au grand maximum et du répéteur de classe II dont la latence maximale est de 0.46 μsec.

L'avantage du répéteur de classe II repose sur le fait qu'il a une latence inférieure à celle du répéteur de classe I. De plus, dans le cas d'une expansion de segment, il est possible d'avoir recours à deux répéteurs de classe II alors que l'on ne peut employer qu'un seul répéteur de classe I dans les mêmes circonstances. Nous en donnons une illustration à la figure 4.23.

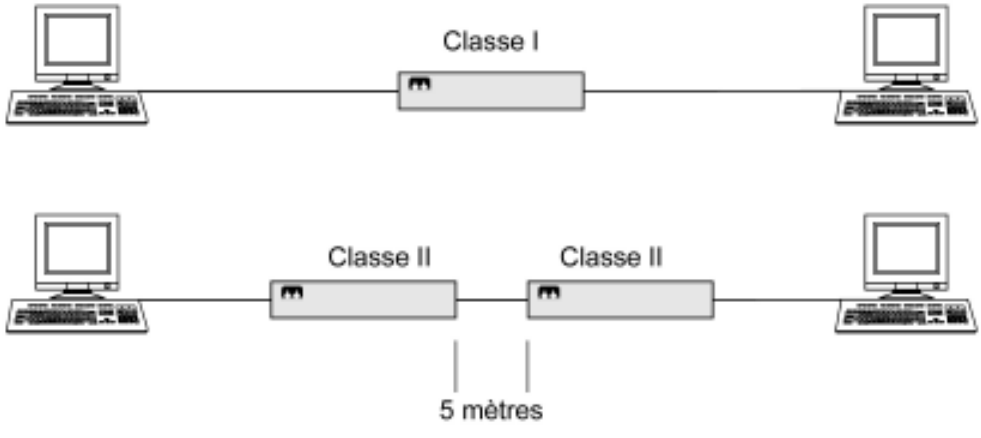


Figure 4.23 Représentation de l'expansion avec répéteurs

Par contre, il faut souligner que les deux répéteurs de classe II utilisés dans le cadre de l'expansion d'un segment ne peuvent pas être éloignés de plus de 5 mètres. Chaque répéteur ayant plusieurs ports, tel qu'illustré à la figure 4.24, on justifie l'utilisation de deux répéteurs de classe II non pour étendre le segment, mais bien plutôt pour permettre l'ajout de ports additionnels.

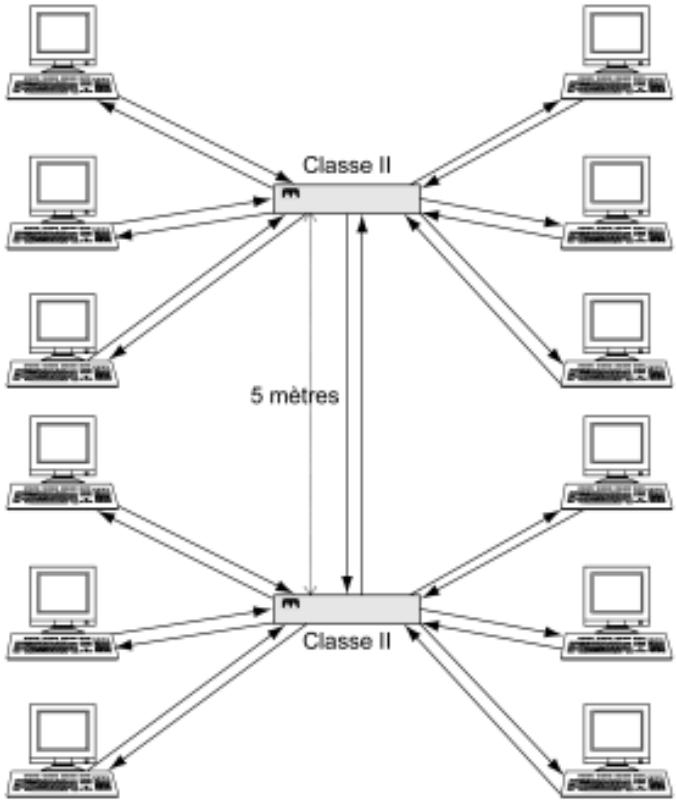


Figure 4.24 Représentation de l'augmentation du nombre de stations

Le répéteur de classe I convertit le signal reçu en un signal numérique. Lorsque ce signal doit être relayé à un port de sortie, le répéteur le reconvertit en un signal analogique selon les spécifications de ce port. Ainsi, le répéteur de classe I peut être utilisé dans les cas où les ports répondent à des configurations distinctes. Un répéteur de classe I pourrait donc avoir un port de 100BASE-T4, un autre port de 100BASE-TX, ou de 100BASE-FX, ou encore et enfin un port de 100BASE-T2.

Le répéteur de classe I peut donc soutenir des ports dont les méthodes d'encodage sont différentes. Il est surtout utilisé dans ce but.

Les répéteurs de classe II doivent se doter de ports dont les méthodes d'encodage sont identiques. De la sorte, on pourrait avoir un 100BASE-TX assigné à un port et un 100BASE-FX assigné à l'autre. Les deux ports se servent de l'encodage 4B/5B. Cependant, si un port relève d'un 100BASE-T4, les autres ports devront l'être aussi.

4.12 Gigabit Ethernet

En juin 1998, une nouvelle norme a été adoptée : il s'agit de la norme IEEE802.3z établissant les aspects techniques du CSMA/CD 1000 Mbps ou gigabit tel que représenté à la figure 4.25. Le Gigabit Ethernet est surtout utilisé en tant que dorsale des réseaux locaux. Or, son application commence à s'étendre aux postes de travail de même qu'à des réseaux de plus grandes portées. Certains PC ne sont pas encore prêts à recevoir ce type d'interface en raison essentiellement des limites des logiciels.

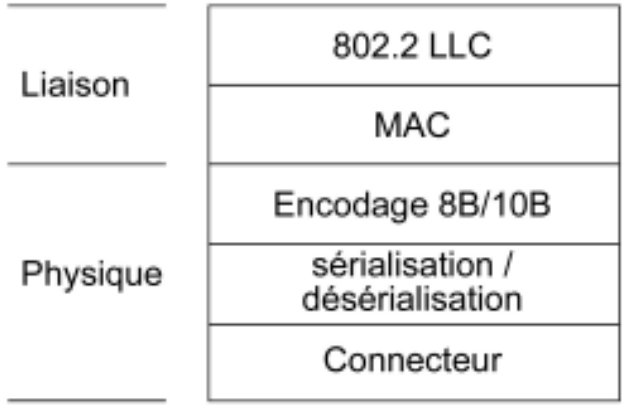


Figure 4.25 Représentation des couches gigabit

Le Gigabit Ethernet répond aux modes duplex intégral et semi-duplex. Tout comme nous l'avons vu précédemment, en mode semi-duplex et dans le cas du 10 Mbps, chaque bit dure 0.1 µsec et la durée de plus petite trame de 64 octets est de 51.2 µsec. Dans le cas du 100 Mbps, la durée de cette trame passe à 5.12 µsec et, pour le gigabit, elle est de 0.512 µsec.

Nous savons aussi que le diamètre du réseau à 10 Mbps est de 2500 mètres (soit la longueur maximale entre stations ou entre une station et un pont). Pour le 100 Mbps, le diamètre réseau passe à 250 mètres. En règle générale, la limite des 200 mètres est appliquée. Si on devait s'en tenir à ces paramètres, le diamètre pour 1 Gbps serait à peine de 20 mètres.

La norme 802.3z a donc établi une prolongation de la trame. Nous en donnons l'illustration à la figure 4.26, après le FCS (*Frame Check Sequence*/Séquence de contrôle de trame), une extension est rajoutée afin que la plus petite trame soit de 4096 bits et que la durée de cette trame soit de 4.096 µsec. L'extension de porteuse, connue sous le nom anglais de *Carrier extension*, n'est nécessaire que

lorsque la trame est plus petite que 4096 bits. Le nombre de ces symboles ajouté dépend de la longueur de la trame.

Après élimination de l'extension, les stations réceptrices s'attendent toujours à recevoir des trames de 64 octets au minimum et n'excédant pas 1518 octets.

Cela nous permet notamment d'avoir un diamètre de 220 mètres sur un 1000BASE-SX.

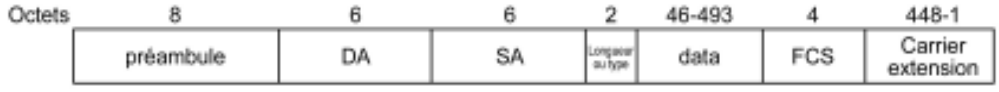


Figure 4.26 Trame gigabit avec extension

Le Gigabit Ethernet fonctionne surtout sur la fibre optique. L'encodage utilisé est le 8B/10B (8 bits codés en 10) tout comme dans le cas du 4B/5B, afin de permettre une alternance d'impulsions. Certains codes ne représentant pas de données sont utilisés pour l'extension.

Le récepteur du 10, 100, ou 1000 Mbps doit présenter à la réception une trame qui ne peut avoir qu'une longueur comprise entre 64 et 1518 octets.

L'autonégociation est aussi utilisée dans le cas d'un 1000BASE-T afin d'établir automatiquement si la vitesse sélectionnée est acceptable par les deux extrémités de la connexion et si le mode semi-duplex ou le mode duplex intégral doit être utilisé.

La vitesse de transmission de 1 gigabit par seconde est rarement nécessaire dans le cas d'une station où l'utilisateur ne requiert que son courriel, le téléchargement des fichiers et la navigation sur Internet. Un simple lien de 10 Mbps, même partagé, suffit à ce genre d'utilisation.

Le lien gigabit par seconde est utile lorsque le réseau comprend un grand nombre de stations et de serveurs. La dorsale utilisée pour procéder au transit des trames peut alors bénéficier des liens gigabits dont l'avantage est de réduire les délais d'attente.

La figure 4.27 nous donne un exemple de configuration comprenant un réseau dorsal et des points d'accès de 10 et 100 Mbps en modes semi-duplex et duplex intégral. La dorsale comprend des Catalyst de Cisco reliés entre eux par des liens gigabits.

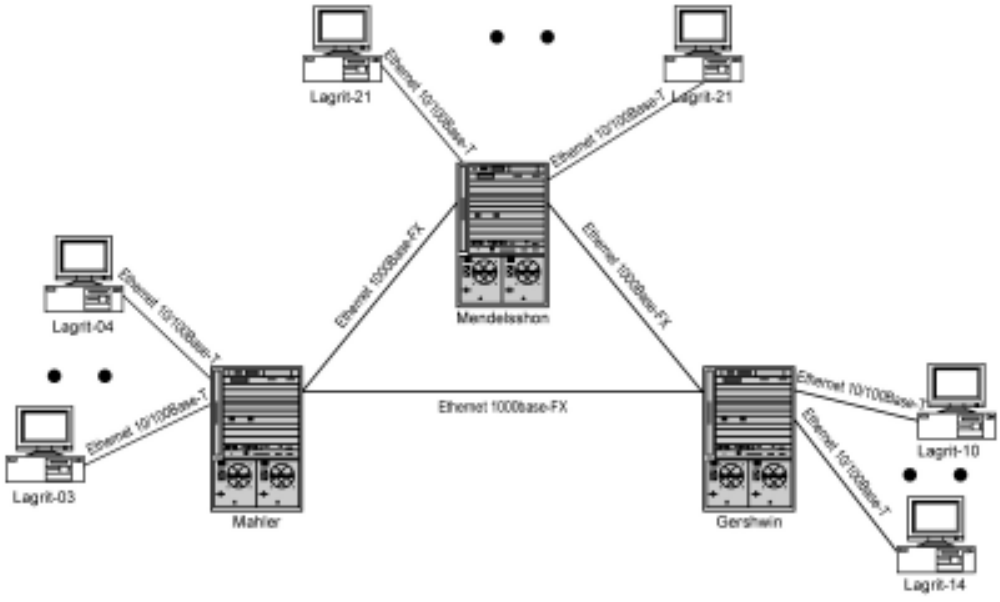


Figure 4.27 Représentation d'un réseau avec dorsale Gigabit Ethernet

4.12.1 Configuration à 1 Gbps

La configuration gigabit 1000BASE-SX transmet, par fibre optique, sur une longueur d'ondes de 850 nm. Malgré la popularité du type 1000BASE-LX qui peut couvrir de plus longues distances, le type 1000BASE-SX, étant plus économique, est réservé pour les courtes distances. Le type 1000BASE-LX émet, par fibre optique, sur une longueur d'ondes de 1300 nm.

Il est possible de transmettre du gigabit sur des câbles de cuivre; le type 1000BASE-CX en est un exemple, mais la longueur du câble peut atteindre, tout au plus, que 25 mètres.

Par contre, il y a eu une évolution dans l'utilisation des câbles de cuivre pour la transmission du gigabit. Le type 1000BASE-T permet de transmettre des données à 1 gigabit sur des paires torsadées de catégorie 5 ou 6. Ces câbles peuvent avoir une longueur maximale de 100 mètres.

Le convertisseur GBIC (*Gigabit Ethernet Interface Converter*/Convertisseur d'interface d'Ethernet gigabit) permet la configuration d'interface avec des composantes externes sans avoir à les changer lorsqu'on passe d'un type de connexion gigabit à l'autre. Ceci est similaire au MII utilisé pour le 100Mbps.

4.13 Exercices

4.1

Quand peut-il y avoir collision dans un réseau CSMA/CD?

1. Lorsqu'une station décide de transmettre même si elle sait qu'une autre station est en train de transmettre.
2. Lorsqu'une station utilise le système ALOHA.
3. Lorsqu'une station n'entend aucun signal sur la ligne pendant un certain temps et décide de transmettre.
4. Lorsque le LAN utilise le processus de déférence.
5. Lorsqu'une station entend le signal de brouillage et décide de transmettre.

4.2

Pourquoi une trame du protocole CSMA/CD doit avoir une longueur minimale?

1. Parce que la station ne doit pas transmettre si elle n'a pas assez d'informations.
2. Pour permettre à la station réceptrice de recevoir les données (temps de transmission de la station émettrice à la station réceptrice).
3. Pour savoir s'il y a eu, au pire des cas, une collision avec une station réceptrice au bout du LAN.
4. Pour minimiser le pourcentage de l'en-tête (*overhead*) dans la trame.

4.3

Une station dans un réseau 10BASE-2 commence à transmettre et détecte immédiatement une collision. Elle applique l'algorithme BEB. Son premier temps d'attente est de un intervalle de temps (*time-slot*). Après la deuxième tentative, suit immédiatement une autre collision. Le deuxième temps d'attente est de deux intervalles de temps. La station transmet par la suite sans détecter de collision.

a) Combien de temps (secondes) la station a-t-elle dû attendre avant d'entamer sa dernière transmission?
b) Si une seule autre station était en conflit avec la station ci-dessus mentionnée, précisez quel était le choix d'intervalles de temps possibles pour cette station lors de chacune des deux tentatives?

4.4

Expliquez pourquoi l'autonégociation n'est possible que dans les cas de 10/100/1000 Base-T.

4.5

a) Pourquoi y a-t-il deux classes de répéteur? Citez les avantages et les inconvénients de chaque classe.

b) Deux répéteurs de classe II ne peuvent être éloignés l'un de l'autre que de 5 mètres dans l'expansion d'un segment. Le coût d'un répéteur additionnel serait-il justifié par un simple gain de 5 mètres? Y a-t-il un autre avantage? Expliquez.

4.6

Dans le cas du CSMA/CD ou de Ethernet, qu'est-ce qui indique que la transmission d'une trame a cessé?

4.7

Comment savez-vous qu'une trame est de type CSMA/CD avec sous-couche LLC, ou CSMA/CD avec sous-couche SNAP ou encore Ethernet?

4.8

En utilisant un programme de surveillance de réseau (exemple Ethereal[2]), trouvez des trames du format SNAP et expliquez où et pourquoi elles sont utilisées dans le contexte de l'application.

4.9

Si la longueur maximale d'un segment est établie à 2500 mètres, quelle est la vitesse de propagation moyenne à travers le segment si la vitesse de transmission est de 10 Mbps et la plus petite trame est hypothétiquement de 100 octets?

4.10

Codez les bits suivants 01100111010111001011
a) en encodage Manchester
b) en encodage 4B/5B

[2] Ethereal est un programme de surveillance gratuit qui peut être téléchargé sur Internet.

4.11

a) Encodez les bits suivants en MLT-3 0101000111011001.
b) Si le code MLT-3 de la question a) est reçu, définissez le diagramme d'état qui va le décoder en binaire.

4.12

Calculez le PPS (*Paquet Per Second*) pour le 100BASE-T pour la plus petite et la plus grande trame.

4.13

Calculez le PPS (*Paquet Per Second*) pour le 1000BASE-T pour la plus petite et la plus grande trame.

4.14

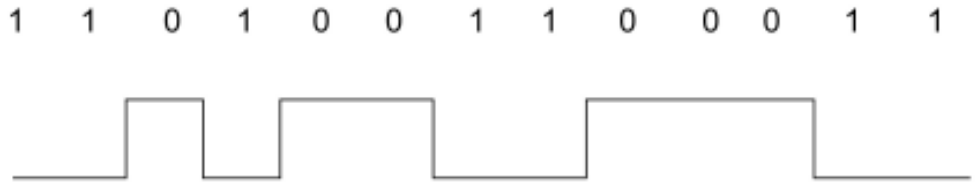
Déterminez la grandeur de la plus petite trame gigabit qui n'a pas besoin d'extension.

Annexe 4A

Codes NRZ et NRZI

Le code NRZ (*Non-return to zero*/Retour à non-zéro) est une représentation du code binaire sous forme d'onde carrée de tension. Le 0 binaire peut être représenté par un niveau de tension positif et le 1 binaire par un niveau de voltage négatif. Ces tensions sont maintenues à leur position jusqu'à qu'il y ait changement de code.

Par exemple, la série de bits suivante est représentée par les ondes carrées correspondantes :

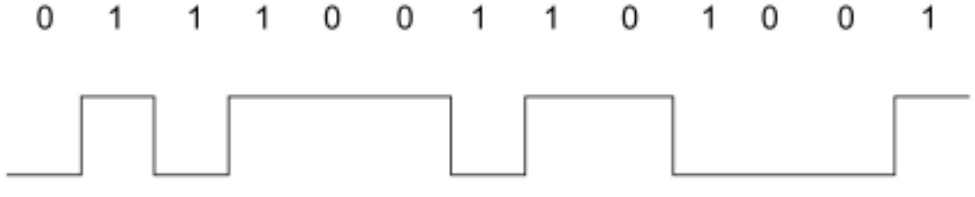


Le NRZI (*non-return-to-zero-change on ones*/non retour à zéro avec changement sur les uns) est un encodage différentiel qui dépend de la polarité du signal adjacent plutôt que de sa propre valeur.

Le 0 binaire n'occasionne aucun changement de niveau de tension. Par contre, le 1 binaire est signalé par un changement ou par une transition de palier. Ce changement peut être positif ou négatif.

C'est la transition et non le palier qui indique la valeur du bit 1 binaire. Cet encodage est plus fiable en présence du bruit. Il est plus facile de détecter une transition que d'établir un palier lorsque le bruit et le code sont indistincts. De plus, la polarité pourrait être inversée accidentellement (par inversion de câbles torsadés). Avec un code différentiel ceci n'a pas d'importance.

L'exemple suivant nous donne une illustration du NRZI.



5 Réseaux locaux sans fil

5.1 Introduction

Le IEEE a adopté en 1997 la première version d'une norme faisant état de la communication sans fil dans un réseau local. Cette norme a été désignée comme IEEE 802.11. Elle fait l'objet d'une reconsidération en 1999. La 802.11, tout comme il est admis de la nommer, définit les services et les protocoles de la sous-couche MAC ainsi que la couche physique pour trois plateformes distinctes.

Les infrastructures physiques décrites par la norme sont : la couche physique à bande basse et infrarouge, le FHSS (*Frequency hopping spread spectrum*/ Étalement du spectre à saut de fréquence) radio dans la bande des 2.4 GHz et le DSSS (*Direct sequence spread spectrum*/Étalement du spectre à séquence directe) radio dans la bande des 2.4 GHz. Ces couches physiques ont été définies initialement pour des vitesses comprises entre 1 et 2 Mbps.

Par la suite, quatre autres infrastructures physiques ont été élaborées et conçues. Il s'agit de la norme IEEE 802.11a (*Orthogonal Frequency Division Multiplexing*/Multiplexage par répartition orthogonale de la fréquence – OFDM radio dans la bande UNII) pour des vitesses allant jusqu'à 54 Mbps sur la bande de 5 à 6 GHz. La seconde est la norme IEEE 802.11b qui utilise l'approche DSSS dans la bande des 2.4 GHz pour des vitesses pouvant atteindre les 11 Mbps. Cette norme (la 802.11b) est incompatible avec la 802.11a. La troisième norme, dénommée la IEEE 802.11g, est tout à fait compatible avec la norme 802.11b. Elle utilise la modulation DSSS-OFDM et opère à des vitesses atteignant 54Mbps. La quatrième est la norme IEEE 802.11n qui utilise les antennes MIMO (Multiple-Input Multiple-Output) et fonctionne dans les bandes 2.4 GHz et 5 GHz. La modulation utilisée est OFDM et peut atteindre des vitesses de 150 Mbps. Son rayon d'opération est supérieur aux autres infrastructure.

Le groupe de travail responsable des normes 802.11 s'est assuré de définir des services et des protocoles de la sous-couche MAC fonctionnant avec la sous-couche LLC (Norme 802.2). Ceci assure une parfaite flexibilité et une

transparence pour toutes les applications des couches supérieures. En effet, ces dernières n'ont pas à se soucier de la nature de l'infrastructure physique. Elles opèrent donc de la même façon que ce soit dans un réseau local avec fil ou dans un réseau local sans fil aussi connu sous le nom de WI-FI (*Wireless Fidelity*).
Malgré leur transparence en regard des couches supérieures, les réseaux locaux sans fil ont des différences marquées par rapport aux réseaux locaux câblés.

Par définition, les réseaux locaux sans fil n'ont pas de câbles qui relient les terminaux aux réseaux. Les liens se font par signaux radios. Les stations sont donc libres de se mouvoir, propriété que ne peuvent avoir les stations câblées. Cette mobilité implique essentiellement des changements en cours de transmission de taux de transmission et de QoS (*Quality of Service*).

Les réseaux locaux sans fil diffusent l'information dans un périmètre défini. Toute station se trouvant dans ce périmètre et dotée de l'interface 802.11 pourra recevoir les données transmises. Celles-ci ne sont pas protégées à moins de les coder par un algorithme de cryptage.

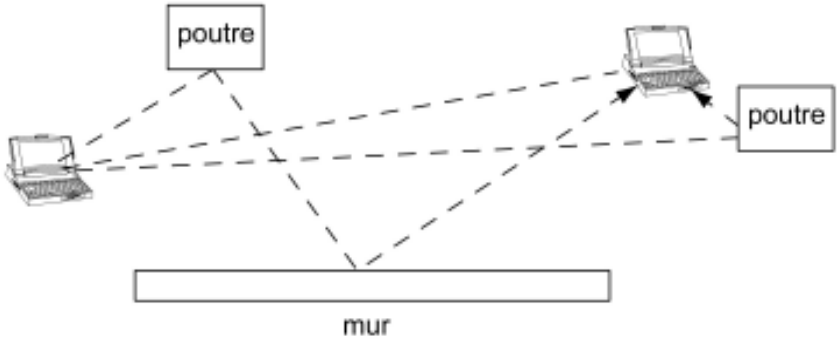


Figure 5.1 Représentation de la réflexion du signal

Étant donné que les réseaux locaux sans fil utilisent l'air comme conduit de transmission, leur environnement les soumet aux effets de la propagation électromagnétique. Tout objet qui s'y trouvera environnant pourra agir comme réflecteur, tel qu'illustré dans la figure 5.1, ou atténuateur de signaux. Une telle situation peut avoir un effet néfaste sur la puissance du signal reçu par les stations. Ce facteur peut devenir d'autant plus imprévisible au moment où les stations se déplacent dans le réseau.

Lorsque la mobilité des stations se limite au réseau local dont le périmètre est celui d'une maison ou même d'un campus, le problème de l'adressage IP ne se pose pas. Par contre, dans le cas où l'accès des stations mobiles au réseau local peut se faire d'un bout à l'autre du pays, alors le problème d'assignation de

l'adresse de la couche réseau se pose. Une des solutions à ce problème est de s'en remettre au DHCP (*Dynamic Host Configuration Protocol*/Protocole DHCP) pour que la station se fasse assigner une adresse flottante. Cette technique sera présentée dans le chapitre 10 consacré au support au réseau IP.
Étant donné que la station mobile fonctionne sur piles, les protocoles de même que l'architecture du réseau local sans fil comprennent des modes d'opérations avec économie d'énergie.

En raison de son architecture décentralisée, le réseau local 802.11 a été pensé et élaboré pour répondre à trois critères :

- la robustesse,
- la flexibilité et
- la tolérance aux erreurs.

L'architecture d'un réseau local sans fil illustrée dans la figure 5.2 et basée sur la norme IEEE 802.11 comprend les éléments suivants :

- Le AP (*Access Point*/Point d'accès) assume l'interface entre le réseau sans fil et le réseau câblé.
- La station comporte une interface avec les protocoles MAC 802.11 et physique selon l'infrastructure et le lien physique du réseau sans fil utilisé.
- Le BSS *(Basic Service Set*/Ensemble du service de base) représente un ensemble de stations 802.11 opérant dans un réseau sans fil fonctionnel.
- Le DS (*Distribution System*/Système de distribution). Cet élément permet au réseau sans fil d'interconnecter de nombreux BSS à travers le AP pour former un ESS (*Extended Service Set*/Ensemble de service étendu). Le ESS est un ensemble de BSS interconnecté par des DS.

Lorsqu'un BSS n'a pas de AP, c'est-à-dire lorsque ce BSS n'a pas d'accès à un autre réseau câblé, il est connu comme étant un BSS indépendant (*Independant* BSS – IBSS). Le IBSS n'est pas très courant et n'est utilisé que pour limiter la communication entre les stations qui s'y trouvent. Le IBSS opére en mode ad-hoc.

Dans un réseau sans fil opérant en mode ad-hoc les stations s'échange directement les informations sans l'intermédiaire d'un point d'accès. Toute la communication se fait point à point entre les deux terminaux.

Un réseau sans fil opére en mode infrastructure lorsque deux ou plusieurs BSSs sont interconnectés à travers des points d'accès (AP). Ces derniers acheminent le

trafic d'un BSS à l'autre possiblement par le biais d'un réseau cablé tel qu'il est illustré dans la figure 5.2.

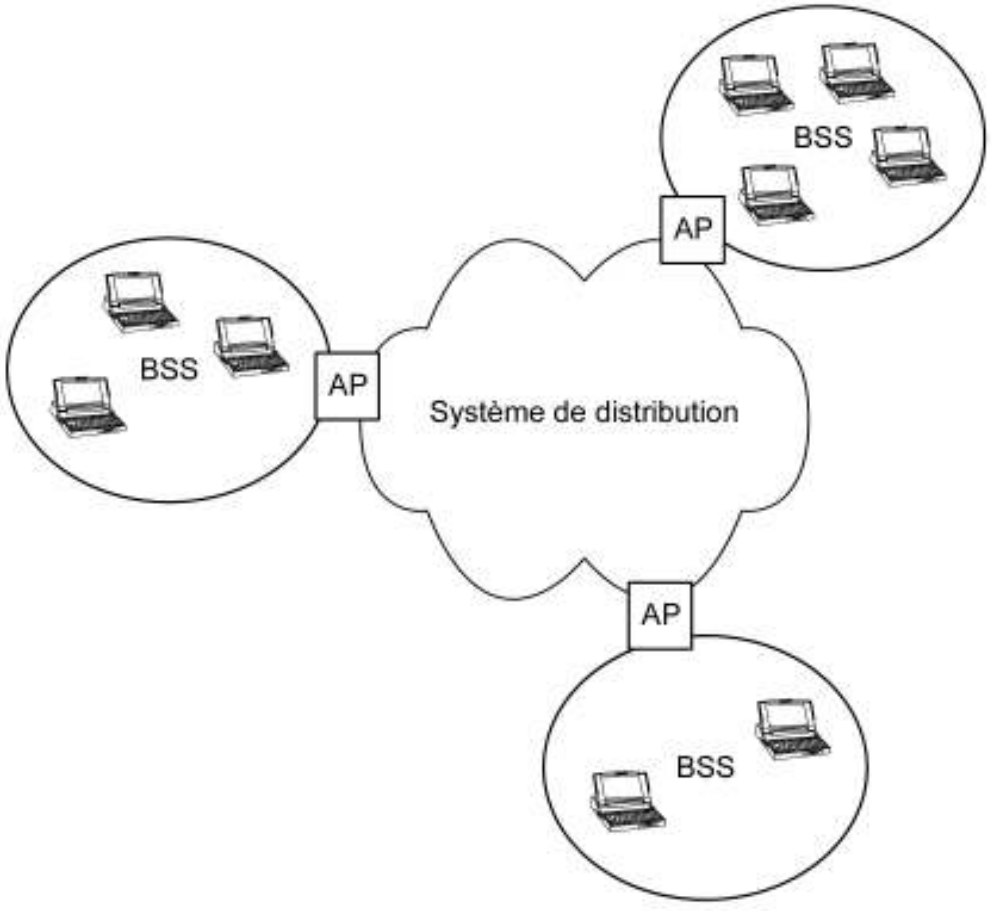


Figure 5.2 Représentation du réseau étendu

5.2 Service IEEE 802.11

La norme IEEE 802.11 définit les services de la sous-couche MAC offerts aux stations et aux systèmes de distribution (DS). Ces services sont également offerts à la sous-couche LLC.

Les services ont été divisés en deux parties : les services de terminaux et les services de systèmes de distribution (DS).

Les services de terminaux, retrouvés aussi dans les points d'accès, sont les suivants :

- L'authentification permet de savoir si le terminal a un droit d'accès en utilisant le même nom de groupe que le point d'accès et les autres terminaux. L'authentification doit se faire avant l'établissement de la connexion.
- La *deauthentification* permet à une station de mettre fin à son rattachement tant au point d'accès qu'au réseau.
- La confidentialité vise à ce que les données ne soient pas à la merci de toute tentative d'écoute. Dans un réseau sans fil, les données sont diffusées et toutes les stations se trouvant dans le périmètre peuvent donc les intercepter. La norme IEEE 802.11 a défini un algorithme de cryptage de messages. Cela a pour effet de réduire les risques d'écoutes non sollicitées. Cet algorithme se nomme WEP (*Wired Equivalent Privacy*/Équivalent de confidentialité par câble).

Les services de stations s'assurent aussi du passage des MAC SDU provenant du LLC. Ceux-ci sont composés de données encapsulées pour devenir des MAC PDU conformément au modèle OSI.

On a recours aux services du système de distribution à partir des points d'accès. Cette procédure permet aux stations de naviguer librement dans le ESS (Ensemble du service étendu) et de communiquer avec d'autres stations au sein d'un autre réseau. Ces services sont les suivants :

- Opération d'association – Afin de pouvoir accéder au système de distribution, la station doit avant tout amorcer un service d'association avec son point d'accès. Chacune des stations n'a qu'un seul point d'accès. Cependant, chacun de ces points d'accès peut accueillir plus d'une station.
- Opération de désassociation – Ce service désassocie la station au système de distribution.
- Opération de distribution – Le service de distribution permet à la station de transmettre des trames MAC par le système de distribution.
- Opération d'intégration – Le service d'intégration permet d'établir la connexion par le système de distribution avec un réseau local distinct de celui du 802.11.
- Opération de réassociation – Le service de réassociation est semblable au service d'association. Par contre, il permet à une station de se réassocier à un nouveau point d'accès au fur et à mesure que la station passe par un ESS.

Il s'agit d'établir l'interaction qui prévaut entre tous ces services. Lorsqu'une station désire communiquer avec une autre station se trouvant au-delà du réseau

local sans fil, elle doit accéder au système de distribution au moyen de son AP. Elle devra alors être authentifiée par les stations avec lesquelles elle désire établir une communication. Ce processus passe par le AP. Une fois authentifiée, la station doit procéder à l'établissement d'une association avec une autre station en passant toujours par un AP. Une fois l'association établie, les stations associées peuvent alors procéder à l'échange des données en utilisant le service de distribution.

Toute station ne peut être associée qu'à une seule à la fois. Par contre, elle peut être authentifiée par plus d'une station. La station peut se dissocier d'avec une station et se réassocier avec une autre station qui l'a déjà authentifiée.

5.3 La sous-couche MAC du IEEE 802.11

La norme IEEE802.11 définit la sous-couche MAC et la couche physique (PHY) pour le réseau sans fil (figure 5.3). Tout comme dans le cas du IEEE802.3 CSMA/CD, la couche MAC est associée à la sous-couche LLC pour former la couche liaison conformément au modèle OSI.

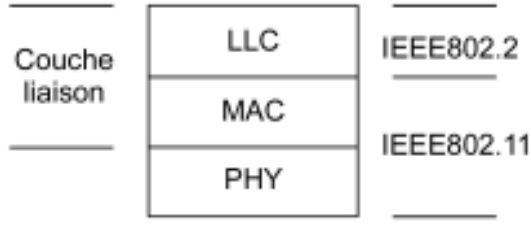


Figure 5.3 Représentation des couches OSI du WI-FI

Le protocole MAC du IEEE802.11 doit assumer la transmission fiable des données, le contrôle de l'accès au réseau sans fil et la protection des données échangées.

Contrairement au protocole CSMA/CD, le protocole MAC du IEEE802.11 transmet ses trames et attend de recevoir un accusé de réception pour chacune d'entre elles.

Toutefois, il est essentiel qu'il n'y ait aucune interruption ou collision au cours de ces échanges. Un certain temps est alloué pour que soit émis tout accusé de réception de trames transmises; temps après lequel si l'accusé de réception n'est pas reçu, un dispositif de retransmission des trames est activé. Cette notion sera expliquée de façon plus détaillée ultérieurement.

Le protocole MAC assume le contrôle de l'accès au réseau sans fil au moyen de deux dispositifs d'accès possibles. Le premier est celui de base, obligatoire pour tous les systèmes conformes à la norme IEEE802.11, et connu sous le nom de DCF (*Distributed Coordination Function*/Fonction de coordination répartie). Le second dispositif en est un de contrôle d'accès centralisé, optionnel dans le cas des réseaux sans fil. Il est connu sous le nom de PCF (*Point Coordination Function*/Fonction de coordination de point d'accès).

Le dispositif d'accès de base, pour tous les systèmes WI-FI, se fonde sur l'approche CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*/Accès multiple avec écoute de porteuse et évitement de collision). Ce dispositif d'accès ressemble à celui du CSMA/CD à quelques différences près. En effet et à l'instar du CSMA/CD, le CSMA/CA écoute à travers le conduit de transmission pour s'assurer qu'il n'y a pas d'activités en cours avant de transmettre. Cette tâche est aussi assumée, en partie, grâce à la couche physique.

Lorsque la station désire transmettre, elle écoute la ligne avant de passer à la transmission. Si la ligne est occupée, la station a recours au système d'attente basé sur le BEB (*Binary Exponential Backup*). Celui-ci permet d'opter pour un temps d'attente aléatoire avant la prochaine tentative de transmission. Ce temps d'attente est défini par la fenêtre de contention composée d'un certain nombre d'intervalles de temps. Pour sa part, la durée de l'intervalle de temps est établie par le réseau.

5.3.1 Vecteur d'allocation réseau

Le protocole MAC de la norme IEEE802.11 comprend le vecteur d'allocation de réseau, paramètre connu sous l'acronyme NAV (*Network Allocation Vector*). Pendant la durée de la transmission, le NAV indique à toutes les stations du BSS le temps prévu et réservé par la station qui occupe la ligne.

Ainsi, toutes les stations vont prendre en compte le temps d'attente assigné par le NAV avant de tenter de transmettre ou de retransmettre à leur tour. Ce système permet aux stations d'éviter toute transmission, et ce, même si la ligne est libre. C'est ce système qui est défini comme évitement de collision (*Collision Avoidance*), c'est-à-dire le CA du CSMA/CA.

5.3.2 Les intervalles

Étant donné qu'il peut y avoir des problèmes entre plusieurs stations au début même de leur transmission, il importe d'être à l'écoute de la ligne pour s'assurer

que la voie est libre avant de transmettre. Le temps d'écoute doit être d'une certaine durée, d'un intervalle spécifié, avant de valider l'état de la ligne.

Le IEEE802.11 a établi les intervalles suivants :

- SIFS – *Short Interframe Space*
- PIFS – *PCF Interframe Space*
- DIFS – *DCF Interframe Space*
- EIFS – *Extended Interframe Space*
- *Time Slot* – Intervalle de temps utilisé par le BEB

Le SIFS est l'intervalle le plus court. Il est suivi d'un intervalle de temps (*Time Slot*). Ces deux intervalles de base sont définis pour la couche physique. Le PIFS est égal à un SIFS plus un intervalle de temps. Le DIFS est égal à un SIFS plus deux intervalles de temps. Le EIFS est le plus long et permet au MAC de procéder à la retransmission lorsqu'il reçoit des trames erronées. Ces intervalles sont illustrés à la figure 5.4.

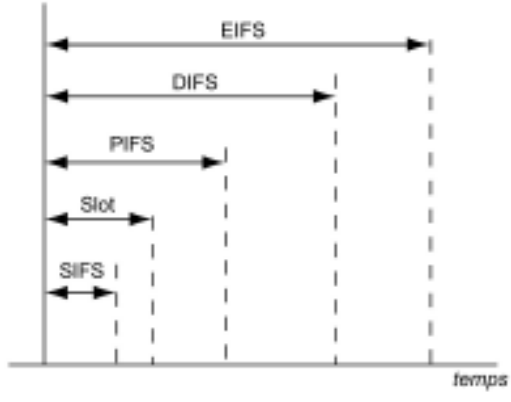


Figure 5.4 Espaces intertrames

5.4 Procédure d'accès DCF

La méthode de contrôle d'accès fondamentale du MAC IEEE802.11, que l'on retrouve dans toutes les stations, est le DCF qui agit conformément au protocole CSMA/CA.

Nous nous penchons à nouveau sur les concepts du CSMA/CA, mais cette fois-ci, de manière plus détaillée en regard du temps d'attente entre les trames.

Une station transmet un MAC PDU après s'être assurée que le conduit est libre pendant au moins une période de temps égale au DIFS. Dès lors, toute station

qui veut et peut transmettre doit donc s'assurer que la ligne est libre pendant au moins une durée correspondant au DIFS.

Après avoir attendu pendant une durée correspondant à la valeur d'un DIFS, si le conduit est libre, la station transmet sa trame de données. Elle attend ensuite l'accusé de réception qui doit lui parvenir dans la limite de temps définie par SIFS. De la sorte, cette procédure évite à toute autre station de transmettre avant que l'accusé de réception de la trame n'ait eu lieu. Nous illustrons ce cas à la figure 5.5.

Si le conduit est occupé ou si une collision survient, la station retient la transmission de sa trame, choisit un chiffre aléatoire selon l'algorithme BEB et attend pendant le temps qui lui est imparti par le nombre d'intervalles de temps assignés.

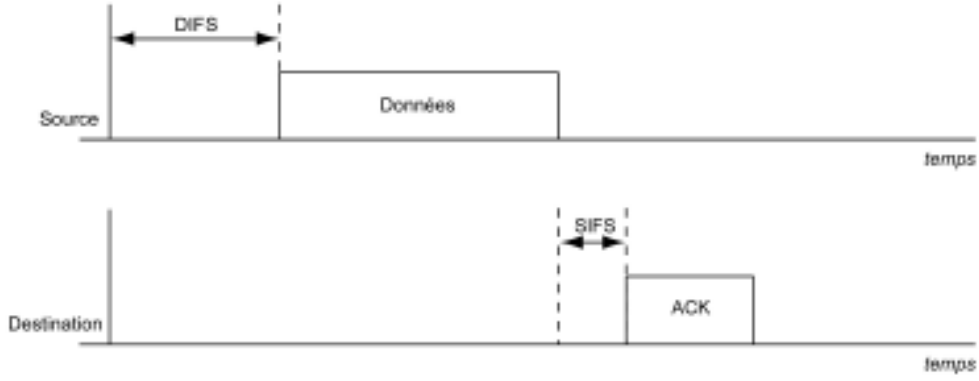


Figure 5.5 Transmission et accusé de réception d'un MAC PDU

Puisque toutes les stations en compétition ne choisissent pas le même chiffre, compte tenu du facteur probabilité, la station qui obtiendra le plus petit nombre attendra moins de temps et sera, par le fait même, la première à transmettre.

Si une autre collision survient, le choix de ce chiffre aléatoire est doublé, ce qui réduit les probabilités de choisir le même chiffre. La possibilité de collision s'en trouve donc diminuée. Par ailleurs, lorsque des stations sont entrées en collision, elles choisissent dès lors un nouveau chiffre parmi ce nouveau nombre de possibilités et attendent pendant une durée correspondant au nombre d'intervalles de temps assignés.

Tout temps d'attente d'intervalle de temps n'est amorcé qu'après s'être assuré que la ligne est libre pendant une durée équivalant au DIFS. Pour qu'une station puisse transmettre, il faut s'assurer que le conduit sera libre pendant tout le

temps de l'attente. Tout comme l'illustre la figure 5.6, le temps sera d'un DIFS auquel vient s'ajouter le temps correspondant au nombre d'intervalles de temps.

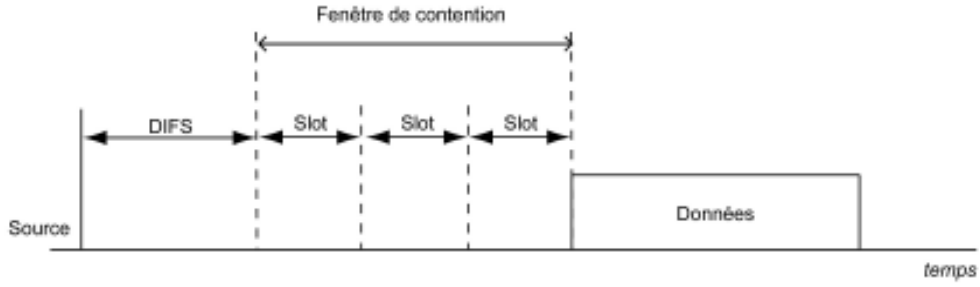


Figure 5.6 Représentation de la transmission du MAC PDU après une collision

Les stations en attente n'activent le chronométrage d'intervalle de temps qu'après un temps d'inoccupation du conduit correspondant au DIFS. Sitôt le conduit occupé, le chronométrage s'interrompt. Il ne reprendra qu'une fois le conduit demeure inoccupé pendant une durée de temps correspondant au DIFS. À un moment donné, alors que tous les intervalles de temps auront été comptés, la station pourra transmettre après le prochain chronométrage d'inoccupation du conduit qui devra correspondre à une durée d'au moins un DIFS.

L'exemple de la figure 5.7 illustre une telle situation. Deux stations prêtes à transmettre des MAC PDU en sont à leur deuxième collision. Le BEB est en vigueur dans les deux stations. La source 1 a choisi le chiffre 3 pour le nombre d'intervalles de temps correspondant à son temps d'attente avant de transmettre. Pour sa part, la source 2 a choisi le chiffre 1 pour le nombre d'intervalles de temps établissant son temps d'attente.

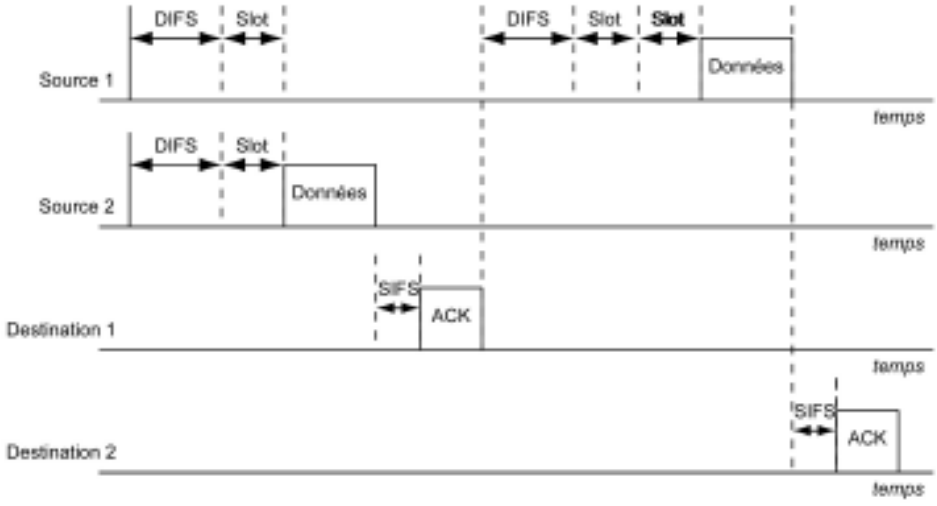


Figure 5.7 Représentation de stations ayant différents temps d'attente

Les deux stations détectent l'inoccupation du conduit pendant un DIFS temps. À ce moment, les deux stations activent le chronométrage du nombre d'intervalles de temps pour couvrir leur fenêtre de contention. La station source 2 transmet des données après une durée équivalant à un intervalle de temps. En attendant, la source 1 qui a laissé passer la valeur d'une durée d'un intervalle de temps arrête son chronométrage lorsque la source 2 commence à transmettre. La source 1 ne reprend son chronométrage que lorsqu'elle détecte un temps d'inoccupation du conduit de DIFS secondes. Elle chronomètre alors la valeur d'une durée correspondant à deux autres intervalles de temps, ce après quoi elle transmet son MAC PDU.

5.4.1 Problèmes de station cachée et exposée : Utilisation de RTS et de CTS

L'opération normale d'échange de données dans un BSS IEEE802.11 peut se faire dans une architecture à infrastructure. Les stations communiquent avec le AP (*Access Point*/Point d'accès) qui relaie les informations jusqu'à leur point de destination. Par contre, toutes les stations sont conscientes de la présence des autres stations par l'écoute des transmissions qui ont lieu dans le conduit. La communication peut donc être réduite à un échange de MAC PDU et d'accusé de réception (ACK) tout comme nous l'avons illustré dans la section précédente.

Il est possible, néanmoins, que le signal d'une station ne soit pas visible pour une autre station (stations cachées) provenant du même BSS tel qu'illustré à la figure 5.8. Les deux stations sont toujours visibles par le AP. Celles-ci pourraient

transmettre simultanément créant ainsi une collision. Toutefois, seul le AP en est conscient. Ceci n'empêcherait pas la collision d'avoir lieu puisque les deux signaux parviendraient au même moment au AP. Ce dernier détecte la collision.

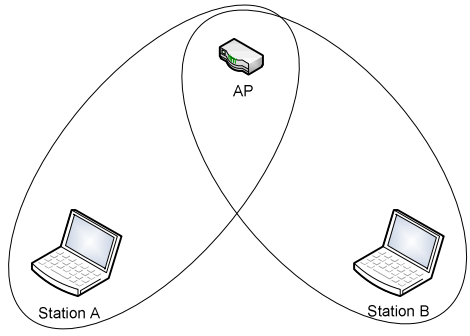


Figure 5.8 Stations cachées : représentation d'une collision non détectée par les stations

Afin de résoudre ce problème, les stations pourraient être configurées de manière à se servir des trames RTS (*Request to Send*) et CTS (*Clear to Send*). De la sorte, la station qui le désire doit, avant de transmettre, envoyer la trame RTS pour demander l'autorisation de transmettre. Le AP lui répond en lui adressant la trame CTS qui comprend l'adresse du demandeur. Toutes les stations détectent le CTS et en déduisent qu'il y a une station invisible qui transmet. La station qui a reçu le CTS transmet sa trame et attend le ACK du AP. Le AP est aussi considéré comme une station lorsqu'il a des trames à relayer. L'utilisation du RTS et CTS est illustrée dans la figure 5.9.

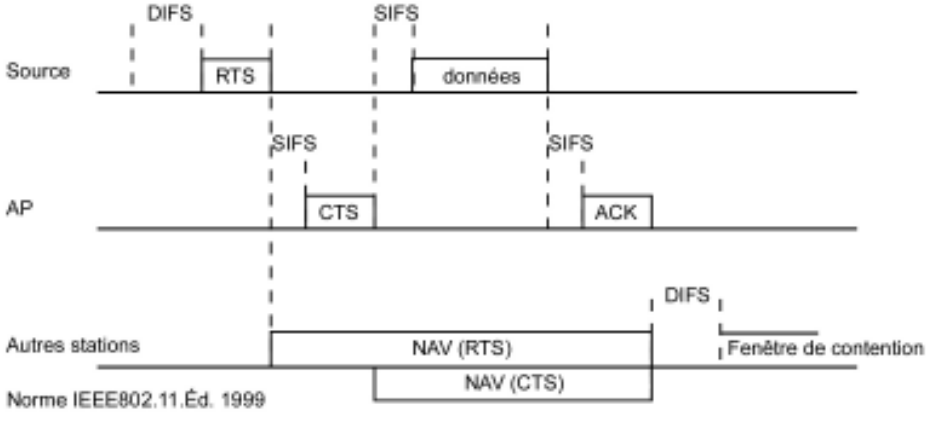


Figure 5.9 Représentation de l'utilisation du RTS et de CTS

Comme nous le verrons plus loin dans ce chapitre, la trame RTS comprend un champ « *duration* » qui nous donne une indication de la durée que prendra la transmission lorsque l'autorisation sera reçue par le CTS. La trame CTS reflète ce temps de transmission des données moins le temps de la transmission du CTS et du SIFS. Tel que mentionné précédemment, toutes les stations détectent le CTS et enregistrent du même coup le contenu du champ « *duration* » même si cette trame ne leur est pas destinée. Les stations mettent à jour leur NAV avec la valeur enregistrée et diffèrent leur accès en attendant ce laps de temps avant de réagir pour transmettre à leur tour.

Le problème de la station exposée se retrouve essentiellement dans les réseaux Ad hoc (IBSS). Il fait état d'une station qui décide ne pas pouvoir transmettre, attribuable au fait qu'une autre station, dans le même champ de transmission, est en train de transmettre. La station exposée doit, pour éviter des collisions, attendre que cette dernière finisse sa transmission. L'illustration de la figure 5.10 en donne un cas de figure. Ainsi, si une communication entre T1 et T2 est en cour, la station T3 ne peut commencer sa communication avec T4 jusqu'à ce que T1 cesse de transmettre. Notons que si T3 transmettait en même temps que T1, les stations cibles T2 et T4 recevraient sans encombre la transmission qui leur est destinée. Toutefois ce fait n'est pas connu des stations émétrices, de la leur retenue.

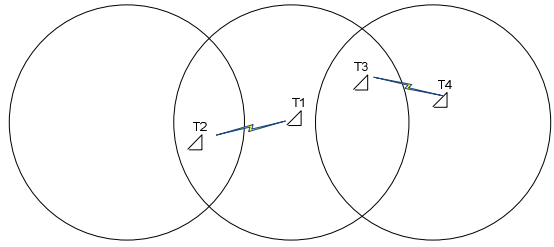


Figure 5.10 Station exposée

L'utilisation des trames RTS et CTS est optionnelle et peut être activée ou désactivée par un attribut de base d'information de gestion. La longueur des trames de données succédant au CTS se voit également assigner un attribut *dot11RTSThreshold* au AP. Cet attribut peut aussi être utilisé pour déterminer si le RTS et le CTS doivent être utilisés.

Si la longueur de la trame de données excède la valeur de l'attribut *dot11RTSThreshold*, on a recours au mécanisme de RTS et au CTS. Dans le cas contraire, on applique le mode simple de transmission de données avec accusé de réception des trames. La valeur par défaut de cet attribut est de 2347.

Les données transmises peuvent faire l'objet de collisions. Elles doivent donc être retransmises. Le nombre de retransmissions est limité. Il existe deux compteurs de retransmissions : le compteur de retransmissions des trames courtes et le compteur de retransmissions des trames longues. Lors de la transmission de trames, le choix de compteur est fixé par l'attribut *dot11RTSThreshold*. Les trames de données plus courtes que la valeur définie par cet attribut font appel au compteur de retransmissions pour trames courtes. Les autres, plus longues que la valeur définie par cet attribut, utilisent le compteur de retransmissions de trames longues. Une fois la transmission de données parachevée, le compteur utilisé est remis à 0. La limite de retransmissions est déterminée par les attributs correspondants :

- *dot11ShortRetryLimit* dans le cas de trames de données courtes avec une valeur par défaut de 7.
- *dot11LongtRetryLimit* dans le cas de trames de données longues avec une valeur par défaut de 4.

Si la limite est atteinte, la transmission est abandonnée.

La définition de ces attributs, tels que définis dans la norme IEEE802.11, se lit comme suit :

```
dot11RTSThreshold OBJECT-TYPE
        SYNTAX INTEGER (0..2347)
        MAX-ACCESS read-write
        STATUS current
::= { dot11OperationEntry 2 }

dot11ShortRtryLimit OBJECT-TYPE
    SYNTAX INTEGER (1..255)
        MAX-ACCESS read-write
        STATUS current
::= { dot11OperationEntry 3 }

dot11LongRtryLimit OBJECT-TYPE
    SYNTAX INTEGER (1..255)
        MAX-ACCESS read-write
        STATUS current
::= { dot11OperationEntry 4 }
```

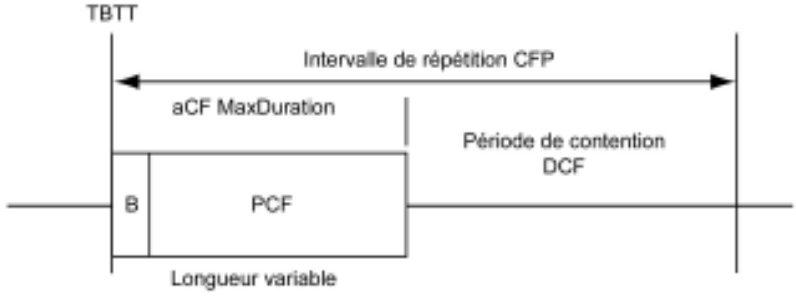
Ces définitions sont particulièrement utiles pour la gestion de réseau (Chapitre 14).

5.5 Procédure d'accès PCF

La procédure d'accès PCF (*Point Coordination Function*/Fonction de coordination de point d'accès) établit par définition une méthode d'accès centralisée qui élimine, durant son fonctionnement, toute possibilité de collision, problème auquel l'on est confronté lors de l'application de la procédure d'accès DCF. Dans le cadre de cette méthode, un point de coordination qui se trouve dans le AP, invite à émettre, de façon séquentielle, les stations qui font partie de ce cycle. Cependant, l'opération du PCF se limite aux réseaux à infrastructure.

Le fonctionnement de la méthode d'accès PCF passe outre la méthode d'accès DCF. Vu que toutes les stations désirent transmettre, le point de coordination se conforme à la méthode d'accès DCF pour obtenir l'autorisation de transmettre. Une fois qu'il a le contrôle du conduit, le point de coordination interroge les stations qui se trouvent sur sa liste de commandes d'invitation à émettre (*polling*) pour leur permettre de transmettre une trame.
Le point de coordination est situé dans le AP. Toutefois, son utilisation est facultative. Les stations doivent pouvoir reconnaître les trames auxquelles l'on a fait appel durant la période PCF, mais elles ne sont pas tenues d'y répondre.

Le point de coordination transmet en premier et périodiquement une trame balisage. La durée pendant laquelle le point de coordination détient le contrôle du conduit est une période sans contention. La collision de trame ne peut guère, en principe, survenir. Tel qu'illustré à la figure 5.11, l'opération périodique PCF a une durée limite, ce après quoi la méthode d'accès DCF prend la relève.



TBTT - *Beacon transmission time* / temps de transmission cible du balisage
B - Trame balise
CFP - *Contention Free Period* / temps libre de contention

Figure 5.11 Représentation des périodes PCF et DCF

Les stations qui ne font pas partie de la liste d'interrogation de commande d'invitation à émettre par le point de coordination doivent signaler quel est leur

état lors de l'établissement d'une association en transmettant la trame demande d'association (*Association Request*) ou demande de réassociation (*Reassociation Request*). Les états possibles lors de l'interrogation des stations sont indiqués dans les paramètres du MAC PDU et sont les suivants :

- STA n'est pas interrogeable (*CF-Pollable*).
- STA est interrogeable (*CF-Pollable*) et demande à ne pas être mise dans la liste d'interrogation.
- STA est interrogeable (*CF-Pollable*) et demande à être mise dans la liste d'interrogation.
- STA est interrogeable (*CF-Pollable*) et demande à ne jamais être interrogée.

Le point de coordination enregistre l'état et agit en conséquence au moment de l'interrogation d'invitation à émettre.

Durant la période d'opération PCF et afin de conserver le contrôle pour toute la durée assignée, le point de coordination utilise le temps intertrame PIFS au lieu de celui du DIFS tout comme dans le cas de l'accès DCF. Vu que le PIFS est plus court que le DIFS, la station qui désire transmettre une trame selon l'accès DCF ne pourra pas le faire et devra attendre la fin de la période PCF pour tenter de transmettre.

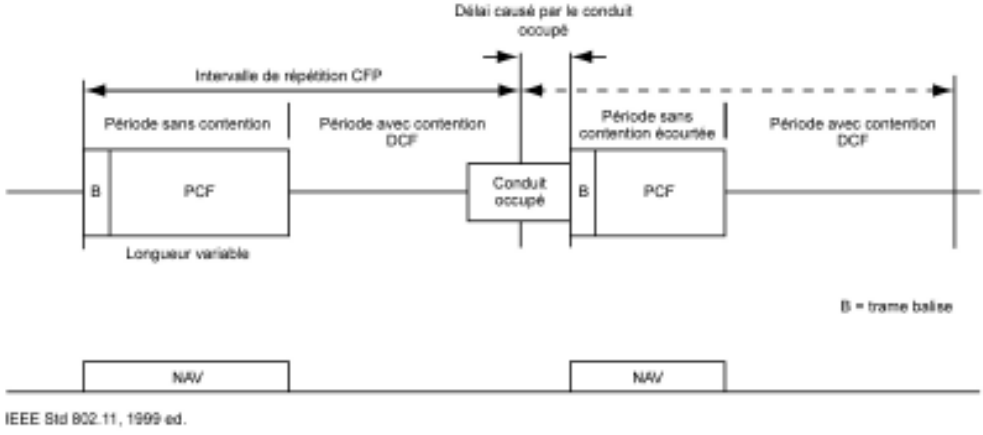


Figure 5.12 Représentation d'une période d'alternance

De plus, toutes les stations, autres que le point de coordination, mettent à jour leur compteur NAV à la valeur *CFPMaxDuration* au début de la période sans compétition de PCF. Cela a pour objet de prévenir toutes les stations que ce

délai d'attente doit être respecté, c'est-à-dire qu'aucune transmission non sollicitée par le point de coordination ne doit avoir lieu durant ce laps de temps.

Le recours au temps intertrame PIFS et au compteur NAV permet d'éviter toute tentative de transmission sans que le point de coordination n'exerce son contrôle durant la période PCF.

Ces périodes étant périodiques, elles offrent un service quasi isochrone. Cela signifie que ce genre de système peut offrir un service de communication sans fil à des stations dont les exigences, en matière de temps pour la transmission de l'information, sont rigides et périodiques.

L'intervalle de répétition de la période CFP commence au point cible défini par l'abréviation TBTT (*Target Beacon Transmission Time*/Temps de transmission cible du balisage). La durée de cet intervalle est indiquée par la trame balisage dans le paramètre intervalle de balisage. Ce dernier est illustré à la figure 5.12.

Il faut noter que la période de contention PCF peut être écourtée de plus d'une façon. Si, durant la période de compétition, une station occupe le conduit pour transmettre et complète sa transmission un peu plus tard en entamant le début de la période sans compétition, cette dernière - qui est la période PCF - devra réduire d'autant sa propre durée (figure 5.13). Une autre situation où la période PCF est écourtée est celle où plusieurs stations sollicitées ne répondent pas et le point de coordination transmet une trame CF-end pour signaler la fin de la période sans compétition.

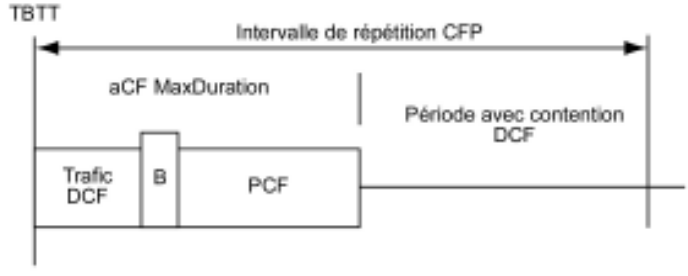


Figure 5.13 Représentation de la durée sans compétition écourtée

Lorsqu'une station est interrogée par le point de coordination, elle peut transmettre un MAC PDU à n'importe laquelle des autres stations et peut y joindre l'accusé de réception correspondant à la trame reçue. Les stations ne peuvent pas retransmettre la trame si le délai d'attente a expiré, à moins d'être réinterrogées par le point de coordination. La station peut effectuer sa

retransmission durant la période de compétition, c'est-à-dire au cours de la période DCF. Le point de coordination peut procéder à la retransmission de ses propres trames d'informations non confirmées.

Avec le début de la période sans compétition, la trame balisage est transmise périodiquement et la transmission en est synchronisée. Lorsque la trame balisage est transmise au début ou durant la période sans compétition, la valeur du paramètre *CFPDurRemaining* est plus élevée que 0. Ce paramètre indique le temps qui reste pour la période sans compétition. Ce paramètre est remis à 0 lorsque la trame balisage est transmise au cours de la période avec compétition. Ce phénomène est illustré à la figure 5.14.

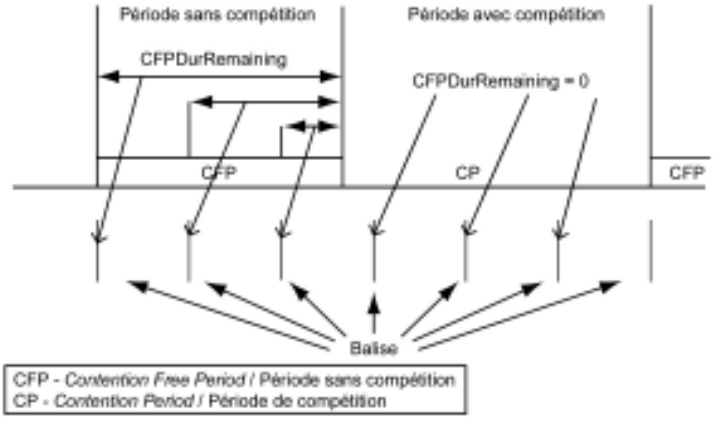


Figure 5.14 Indicateur de durée de période sans compétition

Toutes les trames transmises par le point de coordination vers les stations doivent être confirmées par la station de destination. Cette dernière émet une trame d'accusé de réception (ACK). Au même moment où elle reçoit sa trame d'information, la station pourrait recevoir un avis d'interrogation de la part du point de coordination. Ce processus est indiqué par une trame *Data+CF-Poll*.

D'autre part, la station interrogée pourrait transmettre une trame d'information accompagnée d'un accusé de réception pour la trame d'information reçue du point de coordination. Cette trame est désignée par *Data+CF-ACK*.

Le point de coordination peut transmettre une trame *Data+CF-ACK+CF-Poll* dans laquelle le *data* et l'interrogation (*Poll*) sont destinés à une station alors que l'accusé de réception (*CF-ACK*) l'est pour une toute autre station.

Le point de coordination peut aussi transmettre la trame *CF-Poll* s'il ne désire que permettre à une station de transmettre sans avoir de données pour elle. Il peut

aussi transmettre la trame *CF-ACK+CF-Poll* advenant le cas où il veuille adresser en même temps un accusé de réception.

Lorsque le point de coordination a terminé d'interroger les stations dans sa liste ou lorsque le temps alloué à la période sans compétition a expiré, il met fin à la période sans compétition en transmettant la trame *CF-End* ou *CF-End+ACK*. Aussitôt qu'elles reçoivent cette trame, toutes les stations remettent leur compteur NAV à 0 et la période de compétition commence.

L'exemple donné à la figure 5.15 est tiré de la norme 802.11 et illustre l'utilisation de ces différentes trames.

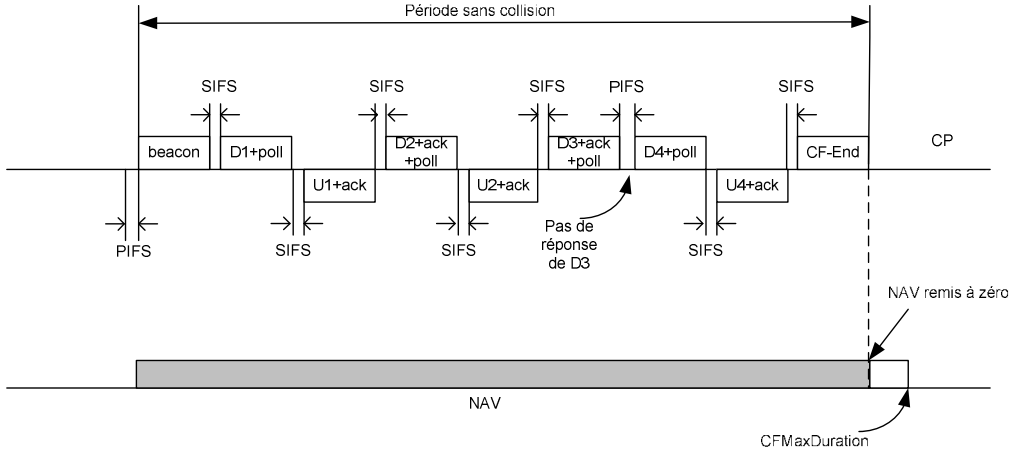


Figure 5.15 Représentation d'un exemple d'activité durant une période sans compétition

5.6 Format des trames MAC

Le transfert de l'information et la gestion de la communication au niveau MAC se font par l'échange de trames MAC PDU. La trame du niveau MAC comprend un en-tête qui permet à toutes les stations du BSS d'être au courant tant des activités et de l'état du système que de la nature de la trame.

L'en-tête de la trame MAC comprend, dans l'ordre, les champs suivants :

- Contrôle de la trame
- Durée
- Adresse1
- Adresse2
- Adresse3

- Contrôle de séquence
- Adresse4
- Contenu ou charge utile
- FCS pour la détection d'erreur

La figure 5.16 en illustre la structure.

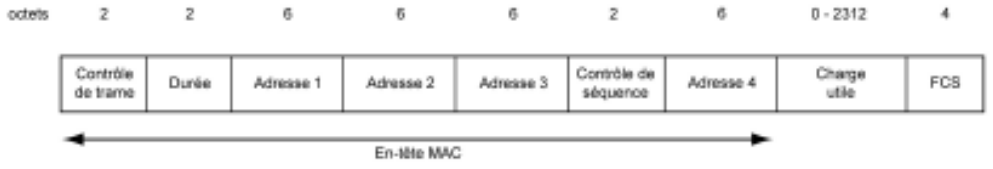


Figure 5.16 MAC PDU

5.6.1 Le champ « contrôle de trame »

Le champ « contrôle de trame » est lui-même composé de sous-champs tels qu'illustrés à la figure 5.17.

Bit 0	Version de protocole	Type	Sous type	To DS	From DS	More frag	Retry	Pwr Mng	More data	WEP	Order
Nombre de Bits	2	2	4	1	1	1	1	1	1	1	1

Figure 5.17 Représentation du champ « contrôle de trame »

Le sous-champ « version de protocole » de 2 bits indique la version du protocole de la trame. Cette valeur est à 0.

Les sous-champs « type » et « sous-type » de 2 et 4 bits respectivement indiquent en commun la nature de la trame. Le sous-champ « type » mentionne si la trame est du type contrôle, donnée ou gestion. La valeur binaire du champ « type » est 00 pour gestion, 01 pour contrôle et 10 pour données. Le tableau 5.1 dresse la liste de tous les types et sous-types de la trame MAC PDU.

La longueur du sous-champ « To_DS » est de 1 bit. Il est mis à 1 lorsque la trame de données est destinée au DS (*Distribution System*/Système de distribution), soit le système de distribution qui se trouve dans le AP. Pour toutes les autres trames, il est mis à 0.

Le sous-champ « From_DS » a une longueur de 1 bit. Il est mis à 1 lorsque la trame de données vient du DS. Pour les autres trames, il est mis à 0. Le tableau 5.2 donne une indication des relations entre ces deux sous-champs.

Tableau 5.1 Représentation des types et des sous-types de la trame MAC PDU

Trame de gestion

Type	Sous-type	Description
00	0000	*Association request* / Demande d'association
	0001	*Association response* / Réponse d'association
	0010	*Reassociation request* / Demande de réassociation
	0011	*Reassociation response* / Réponse de réassociation
	0100	*Probe request* / Demande de sondage
	0101	*Probe response* / Réponse de sondage
	0110-0111	*Reserved* / Réservé
	1000	*Beacon* / Balisage
	1001	*Announcement traffic indication message* / Message d'avis d'indication de trafic
	1010	*Disassociation* / Désassociation
	1011	*Authentification* / Authentification
	1100	*Deauthentification* / Non-authentification
	1101-1111	*Reserved* / Réservé

Trame de contrôle

Type	Sous-type	Description
01	0000-1001	*Reserved* / Réservé
	1010	*Power save (PS)-Poll* / Économie d'énergie invitation à émettre
	1011	*Request to send (RTS)* / Demande de transmission
	1100	*Clear to send (CTS)* / Signal de voie libre
	1101	*Probe request* / Demande de sondage
	1110	*Contention free (CF)-End* / Fin de période sans compétition
	1111	*CF-End + (CF)-End*

Trame de données

Type	Sous-type	Description
10	0000	*Data* / Données
	0001	*Data + CF-Ack*
	0010	*Data + CF-Poll*
	0011	*Data + CF-Ack + CF-Poll*
	0100	*Null* / Nul
	0101	*CF-Ack*
	0110	*CF-Poll*
	0111	*CF-Ack + CF-Poll*
	1000-1111	*Reserved* / Réservé

Tableau 5.2 Relations entre les sous-champs « To_DS » et « From_DS »

To_DS	From_DS	Indication
0	0	Trame de données d'une station à l'autre (system BSS - AdHoc)
1	0	Données destinées au DS
0	1	Données provenant du DS
1	1	Données en transit d'un AP à un autre

La longueur du sous-champ « *More Fragment* » est de 1 bit. Il est mis à 1 lorsque, dans les trames de gestion ou de données, le MAC SDU a été fragmenté et la présente trame n'en est qu'une partie. Ce bit est mis à 0 pour toutes les autres trames.

La longueur du sous-champ « *Retry* » est de 1 bit. Lorsqu'il est à 1, ce sous-champ nous mentionne que la trame procède à la retransmission d'une trame de données ou de gestion.

Le sous-champ « *Power Management* » indique le mode d'économie d'énergie d'une station. Lorsque sa valeur est à 1, la station signale qu'elle se met en mode d'économie d'énergie après la transmission de la présente trame. Ce sous-champ de 1 bit est toujours à 0 dans les trames transmises par le AP.

Le sous-champ « *More Data* » est utilisé pour indiquer à une station en mode d'économie d'énergie que les données, qui lui sont destinées, sont disponibles au AP.

Le sous-champ « WEP » (*Wired Equivalent Privacy*/Équivalent de confidentialité par câble) est mis à 1 lorsque le contenu de la charge utile de la trame a été traité par l'algorithme WEP. Ce cas n'est valable que pour les trames de données et de gestion.

Le sous-champ « *Order* » est mis à 1 et utilisé par des trames de données pour souligner que la classe de services « strictement ordonné » est requise. Ce service garantit l'ordonnancement des trames lors de leur transmission.

5.6.2 Champ « durée »

Le champ « durée » prend sa signification en fonction de la trame et de la période au cours de laquelle l'on y fait appel. Sa longueur est de 16 bits.

Durant la période de compétition, lorsque la trame est utilisée dans le DCF, le champ « durée » donne la durée de l'occupation de la ligne par la station. Cette valeur est prise en compte par les autres stations et copiée dans leur compteur NAV. La valeur va de 0 à 32767.

Durant la période sans compétition, lorsque la trame est utilisée dans le PCF, ce champ prend une valeur fixe de 32768. Ceci est représenté par la valeur binaire suivante : 1000000000000000.

Pour la trame « *Power Save-Poll* », le champ « durée » est remplacé par le champ « AID » qui est utilisé au moment où une station est en mode d'économie d'énergie. Ce champ établit le numéro d'identité décerné à la station par la trame de confirmation d'association et connu sous l'abréviation AID (*Association Identity*). Sa valeur s'étend alors de 1 à 2007. Ce numéro est directement associé à la station et permet au AP d'indiquer à chacune des stations s'il y a des trames stockées en mémoire. Nous verrons le fonctionnement de ce dispositif lorsque nous aborderons le mode d'économie d'énergie.

5.6.3 Les champs « adresse »

Comme illustré à la figure 5.15, il y a quatre champs « adresse » dans la trame du MAC PDU. L'assignation des adresses pour les trames « *Data* » dépend du contenu des sous-champs « To_DS » et « From_DS ».

Les champs « adresse » peuvent désigner le BSSID (*Basic Service Set Identification*) qui est l'identification du BSS, l'adresse source (SA), l'adresse destination (DA), l'adresse de la station qui transmet (TA) et l'adresse de la station qui reçoit (RA). La station qui transmet et celle qui reçoit peuvent être différentes des stations source et destination. Cette situation va se produire dans les deux cas suivants :

- si les trames proviennent d'une source externe au BSS, ou
- si les trames sont adressées à une destination externe au BSS.

Certaines trames de contrôle ne comprennent pas toutes les adresses.

Le BSSID identifie de façon unique le BSS. Cette adresse de 48 bits a le même format que les adresses IEEE MAC. L'adresse BSSID est l'adresse IEEE MAC du AP utilisé dans le BSS en mode infrastructure.

Les adresses source (SA) et destination (DA) sont, respectivement, les adresses IEEE MAC des stations d'origine et de destination du MAC PDU. Dans le cas de l'adresse source, le contenu ne peut être qu'une adresse individuelle, alors que

dans celui de l'adresse destination, le contenu peut être soit une adresse individuelle, soit une adresse de groupe.

Les adresses de la station émettrice (TA) et de la station réceptrice (RA) sont des adresses IEEE MAC identifiant respectivement la station individuelle qui transmet la trame et la station, ou le groupe de stations, recevant directement la trame pour la relayer si besoin est.

Tout comme nous en avons fait mention, les adresses des trames « *data* » dépendent des valeurs des sous-champs « To_DS » et « From_DS ». L'assignation des adresses par rapport à ces sous-champs figure au tableau 5.3.

Tableau 5.3 Représentation de l'assignation des adresses dans la trame donnée

To_DS	From_DS	Adresse 1	Adresse 2	Adresse 3	Adresse 4
0	0	DA	SA	BSSID	N/A
0	1	DA	BSSID	SA	N/A
1	0	BSSID	SA	DA	N/A
1	1	RA	TA	DA	SA

5.6.4 Le champ « contrôle de séquence »

Le champ « contrôle de séquence » a une longueur de 16 bits. Les 12 derniers bits mentionnent le numéro de séquence du MAC SDU. Pour leur part, les quatre premiers bits indiquent le numéro du fragment une fois le MAC SDU fragmenté. Les fragments sont transmis avec le numéro du fragment, ce qui a pour effet d'en mentionner la séquence de transmission par rapport aux autres fragments. Tous les fragments du même MAC SDU conservent le même numéro de séquence. S'il n'y a pas fragmentation, le numéro du fragment reste à 0.

5.6.5 Le champ « FCS »

Le champ « FCS » (*Frame Check Sequence*/Zone de séquence de contrôle) comprend un CRC (*Cyclic Redundancy Code*/Code cyclique de redondance) de 32 bits auxquels l'on a recours pour vérifier la présence d'erreurs dans la trame. Le polynôme alors utilisé est de degré 32.

5.7 Les trames de contrôle

Les trames de contrôle comportent les trois trames RTS, CTS, ACK outre celles mentionnées dans le Tableau 5.1. La figure 5.18 illustre le format de quelques-unes des trames de contrôle. Tous les bits des trames de contrôle du champ « contrôle de trame » se retrouvent après le sous-champ « sous-type » mis à 0, exception faite du « *Pwr Mgt* » qui pourrait être mis à 1.

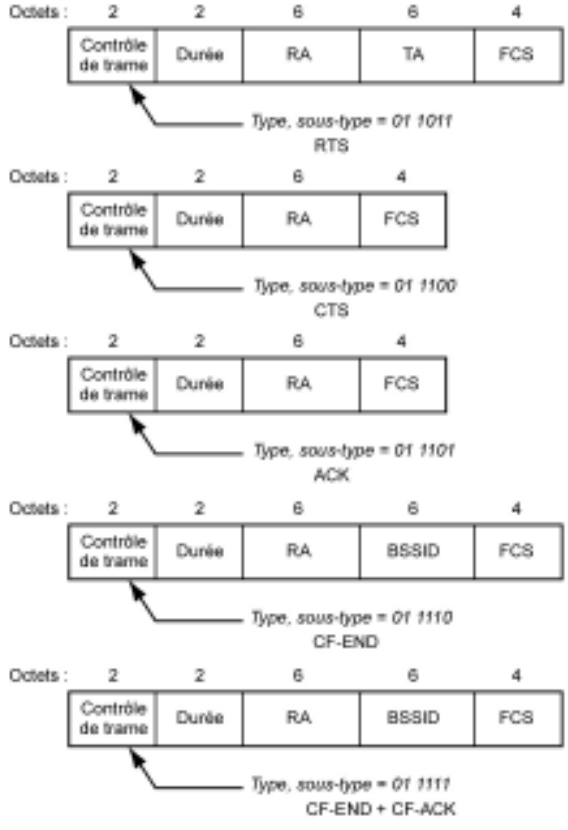


Figure 5.18 Quelques trames de contrôle

5.8 Trames de gestion

Toutes les trames de gestion obéissent au format illustré à la figure 5.19. Le contenu de la trame se compose de champs fixes et d'éléments d'information définis pour chacune des trames de gestion. Les éléments d'information sont obligatoires et doivent être présents dans un ordre prédéterminé.

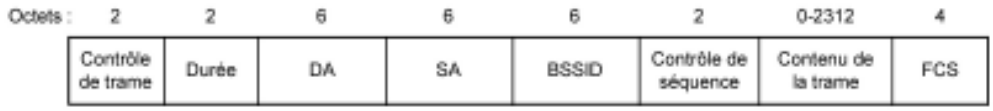


Figure 5.19 Représentation du format de la trame de gestion

La trame balisage, représentée à la figure 5.20, est une trame de gestion dont les codes de type et de sous-type sont respectivement 00 1000. En outre, le contenu de cette trame comprend un certain nombre d'éléments d'information utiles à la gestion du BSS et surtout à la consolidation de la période sans compétition décrite précédemment. Le paramètre SSID joue le rôle d'identificateur de l'éventail de services (*Service Set Identity*) qui n'est autre que le ESS.

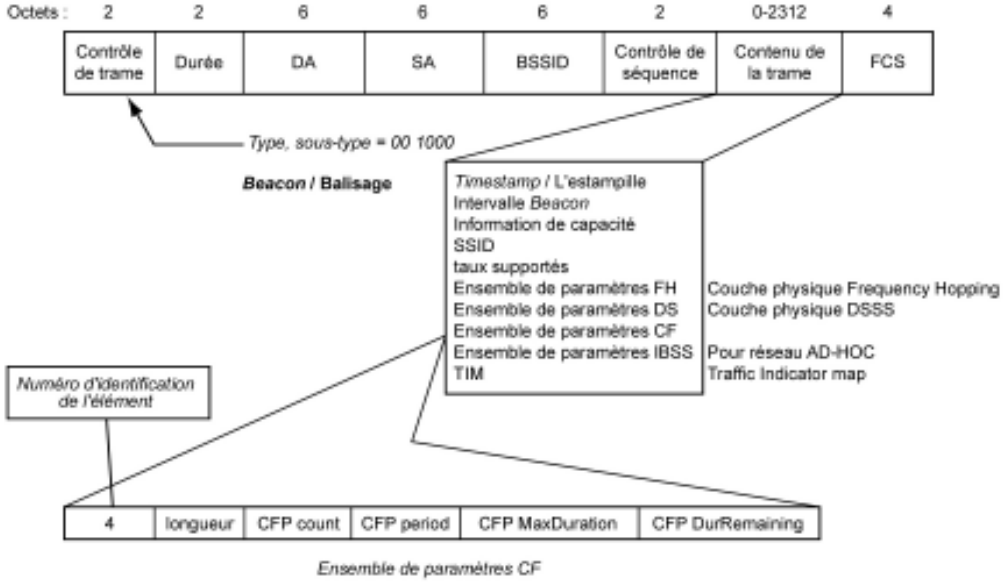


Figure 5.20 Représentation de la trame balisage

Par ailleurs, une autre trame de gestion très utile est la trame demande d'association (*Association Request*). Le contenu de cette trame est représenté à la figure 5.21. C'est par cette trame que la station avise le AP de son entrée dans le BSS et lui indique si elle veut être ou non interrogée. Cette procédure se fait grâce au paramètre « information de capacité » par les valeurs *CF Pollable* et *CF Poll Request*.

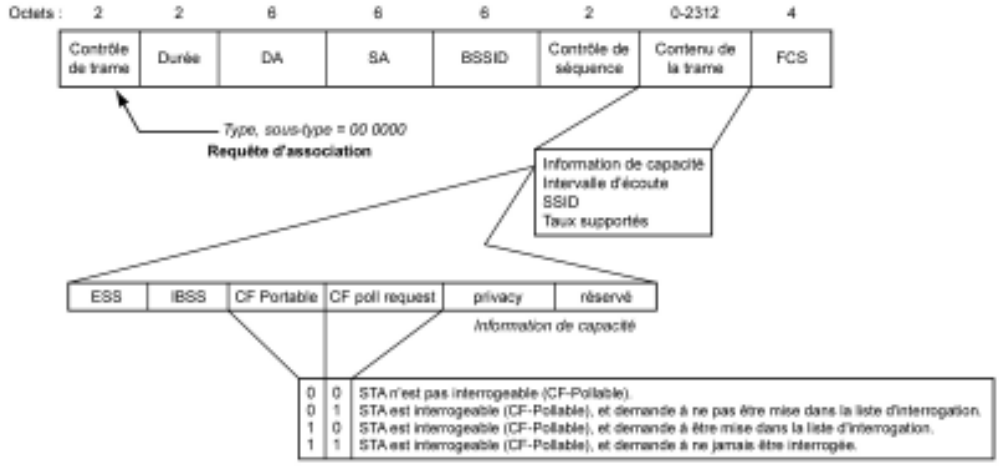


Figure 5.21 Représentation de la trame demande d'association

La confirmation de cette demande se fait par la trame « *Association Response* » ou confirmation d'association dont la fonction est aussi d'assigner une identification d'association par le champ « *Association ID* » (AID). Ce processus est illustré à la figure 5.22.

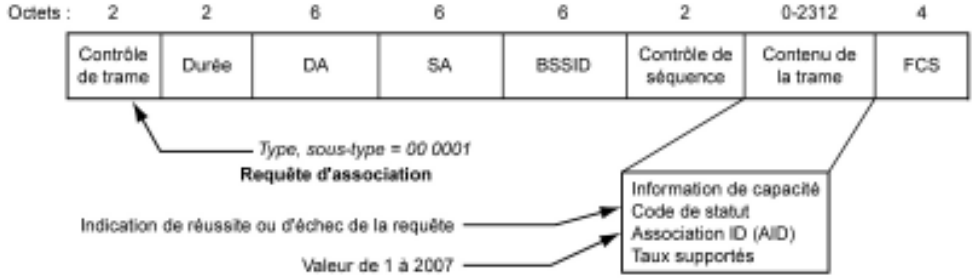


Figure 5.22 Représentation de la trame de confirmation d'association

Les autres trames de gestion diffèrent dans leur code sous-type et dans la composition des éléments d'information retrouvés dans le contenu de la trame.

5.9 Confidentialité et sécurité

Le réseau local sans fil est par définition accessible par tous les terminaux dotés d'une interface d'accès compatible avec les protocoles utilisés. L'accès à un réseau sans fil doit donc être pensé différemment de celui d'un réseau avec fil.

Le comité du IEEE 802.11 a développé des procédures d'accès sécuritaires se rapprochant autant que possible de celle des réseaux avec fil. Le but premier étant d'éviter les écoutes clandestines.

Les procédures d'accès sécuritaires recommandées par la norme comprennent la procédure d'authentification des stations ainsi que l'application de l'algorithme de cryptage WEP (Wired Equivalent Privacy) ou WPA (Wi-Fi Protected Access) en regard des données.

La norme IEEE 802.11 ne prétend pas, par ces procédures, assurer une protection complète à toute intrusion non sollicitée. Toutefois, elle tente d'en limiter l'accès au même niveau que dans le cas de l'accès à des réseaux avec fil.

5.10 Économie d'énergie

En règle générale, les réseaux locaux sans fil sont exploités par des stations mobiles qui fonctionnent sur piles. Il importe donc pour ces stations d'utiliser un système qui leur permettra d'économiser de l'énergie.

Une fois que la station active ce dispositif, elle se met en état de veille et ferme son récepteur et son transmetteur pendant un certain temps donné.

Deux systèmes d'économie d'énergie ont été mis au point. Le premier est réservé aux BSS à infrastructure et le deuxième pour les IBSS ou réseau ad hoc.

La gestion du système d'économie d'énergie dans un BSS à infrastructure est prise en charge par le AP. Ce dernier stocke les trames destinées aux stations en état de veille et ne les leur livre que lorsque ces stations en font la demande.

Chaque station se doit de prévenir le AP de sa mise en veille périodique en transmettant certains paramètres lors de la transmission de la trame de requête d'association. La durée de veille correspond à la durée d'un nombre de trames balisage spécifié dans le paramètre intervalle d'écoute (*Listen Interval*).

La station désireuse de changer de mode d'économie d'énergie le signale en utilisant le bit *Pwr Mng* qui se trouve dans le champ « contrôle de trame » de la trame transmise.

Le AP ne stocke pas les trames des stations qui ne sont pas en mode d'économie d'énergie, mais il les transmet directement.

Le AP transmet périodiquement une trame balisage qui comprend le paramètre TIM (*Traffic Indicator Map*). Un des sous-champs du paramètre TIM est le *Partial Virtual Bitmap*. Celui-ci signale aux stations du BSS les données stockées dans le AP.

Tel qu'illustré à la figure 5.23, le sous-champ « *Partial Virtual Bitmap »* comprend 2008 bits. Chaque bit de 1 à 2007 désigne une station par son AID assigné lors de la confirmation d'association (*Association Response*). Si le bit est à 1, cela indique qu'il y a des données pour cette station dans le AP. Le bit 0 de ce sous-champ est réservé. Il signale qu'il y a des trames diffusion (*Broadcast*) ou multidiffusion (*Multicast*) dans le AP.

Les stations en mode de veille écoutent périodiquement des trames balisage pour savoir si des trames sont en attente dans le AP.

Si une station détecte, dans une trame balisage, l'indication à l'effet que le point d'accès a des trames stockées pour elle et que la période en cours en est une de compétition, elle transmet au AP une trame *PS-Poll*. Le AP répond en transmettant la trame en mémoire. Si le AP a d'autres trames à transmettre à la même station, il met le bit *More Bit* à 1. La station doit transmettre une trame *PS-Poll* pour chaque trame reçue avec *More Bit* à 1. Le AP met le bit *More Bit* à 0 qu'une fois la transmission de la dernière trame stockée pour la station est envoyée.

Durant la période sans compétition, la station qui détecte l'existence de trames stockées dans le AP ne transmet aucune demande, mais reste en éveil durant toute la période et attend que le point de coordination, qui se trouve au AP, lui transmette ses trames.

Les stations faisant partie des stations interrogées par le AP (*Pollable*) doivent être en éveil durant toute la durée de la période sans compétition.

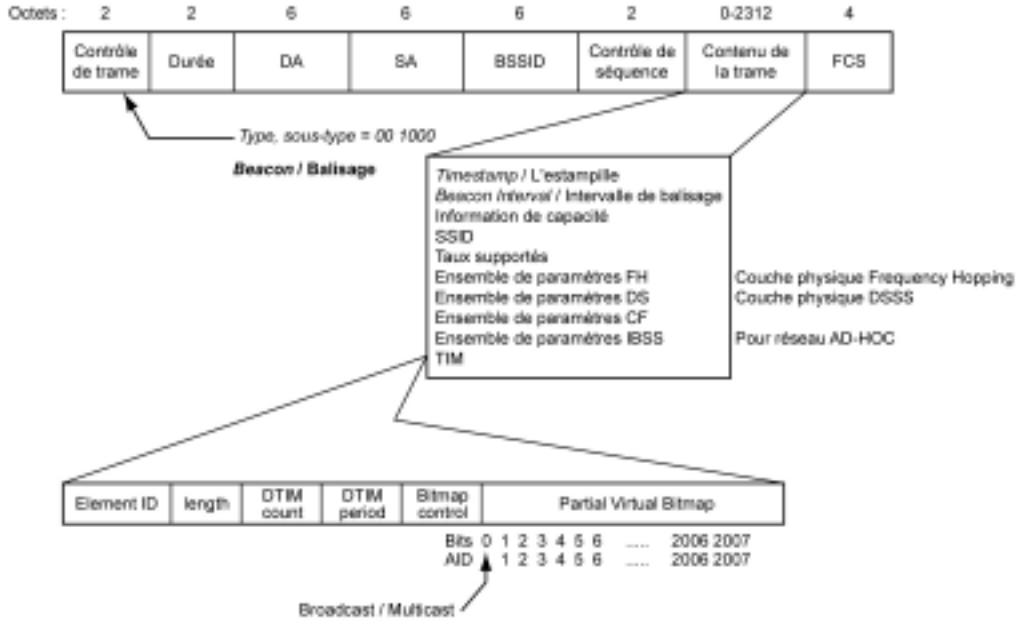


Figure 5.23 Représentation de la trame balisage et du paramètre TIM

Le paramètre TIM est transmis dans toutes les trames balisage. Il existe un autre type de TIM, le DTIM qui est établi lorsque le sous-paramètre DTIM count du paramètre TIM est à 0. Le DTIM (*Delivery Traffic Indication Map*) est transmis périodiquement dans le balisage de tous les DTIMperiod. DTIMperiod est un autre sous-paramètre du paramètre TIM.

Après la transmission d'un DTIM, le AP transmet les trames diffusion/ multidiffusion avant de transmettre les trames diffusion individuelle lorsque le bit 0 du sous-paramètre *Partial Virtual Bitmap* se trouve à 1. Si le AP a d'autres trames diffusion/multidiffusion, il les transmet par la suite après les TIM.

Un exemple extrait de la norme IEEE 802.11 et représenté à la figure 5.24 montre deux stations en mode de veille. La première découvre, en interrogeant le AP avec un « *PS-Poll* », qu'il y a une trame pour elle. Le AP la lui envoie après avoir reçu la trame « *PS-Poll* ». La deuxième station n'ayant qu'une très faible puissance ne surveille que de temps à autre, mais au moment de la transmission d'un TIM, elle détecte qu'une trame a été stockée pendant son état de veille. Elle la demande en envoyant une trame « *PS-Poll* ». Le AP lui envoie la trame sur réception de la trame « *PS-Poll* ».

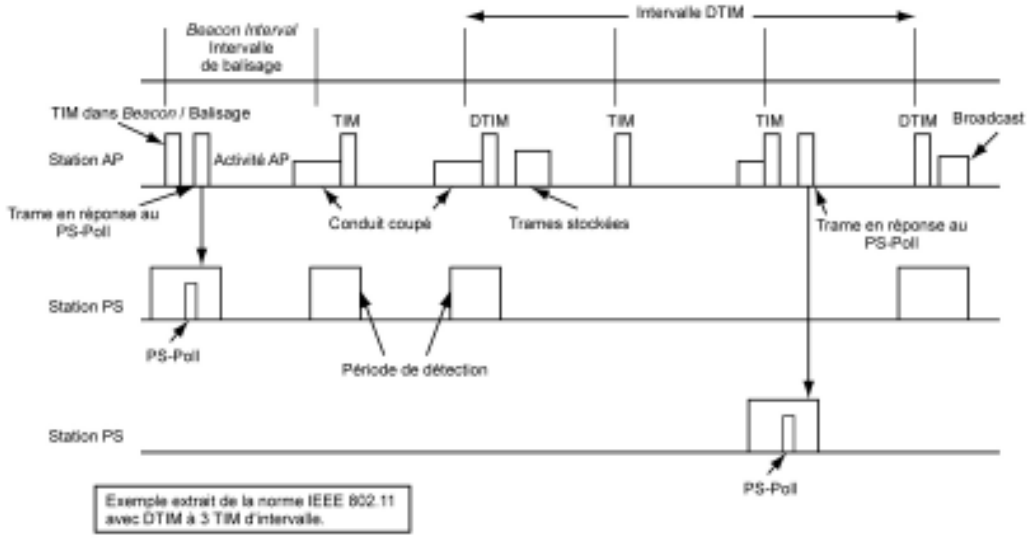


Figure 5.24
Représentation d'une opération de gestion de l'énergie en infrastructure (Opération DCF)

5.11 AP et cellules multiples

La station qui se trouve à l'intérieur d'un BSS à infrastructure pourrait se déplacer et se retrouver dans un autre BSS ou cellule. Si la station est en communication durant sa mobilité, il est essentiel de lui permettre de conserver sa connexion durant sa transition. Cette procédure est possible lorsqu'une station est associée à un AP et, qu'en se déplaçant, elle effectue une réassociation avec le nouvel AP de la nouvelle cellule. Ce processus est connu sous le nom d'itinérance (*roaming*) que l'on retrouve aussi dans les réseaux de téléphonie cellulaire. La figure 5.25 illustre le positionnement des AP dans les cellules.

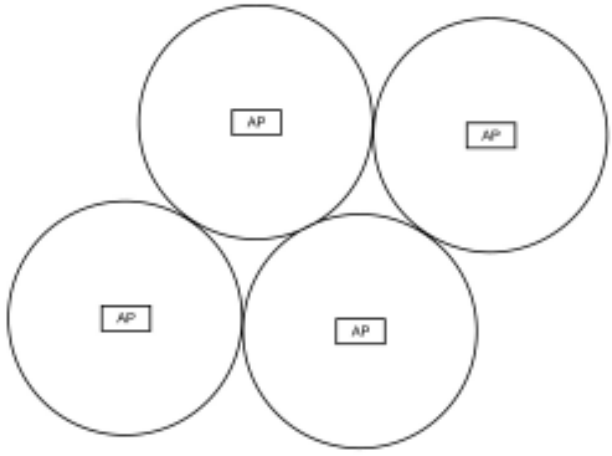


Figure 5.25 Représentation type du AP et des cellules

Le placement des APs dans le territoire que l'on veut couvrir dépend de la qualité de la connexion qui est offerte dans le réseau sans fil. Lorsqu'un terminal s'éloigne d'un point d'accès le signal qu'ils partagent s'atténue. Si la même vitesse de transmission est maintenue à toutes les positions, il risque d'avoir des erreurs de transmission lorsque le terminal est plus éloigné du AP. Afin de maintenir la fiabilité du lien, le AP réduit sa vitesse de transmission. Il effectue ce qu'on nomme un déplacement de vitesse (*Data Rate Shifting*) aussi connu comme (*Multi Rate Shifting*). Ceci veut dire que le changement du taux de transmission ne varie pas de façon uniforme mais plutôt par échelon. Lorsque le AP détecte que le signal a atteint un niveau défini, il passe directement à une autre vitesse de transmission.

Dans le cas d'un point d'accès 802.11b, le déplacement de vitesse se fait selon l'aire courverte par le AP. La figure 5.26 illustre cette couverture. Par exemple, pour une puissance de 100 mW dans le cas du 802.11b, la région 1 qui atteint une distance maximale de 130 pieds, la vitesse de transmission entre le AP et un terminal se trouvant dans cette région est à son maximum, c'est-à-dire 11 Mbps. La région 2 se situant entre 130 pieds et 180 pieds, la vitesse de transmission est réduite à 5.5 Mbps. La région 3 se situe entre 180 pieds et 250 pieds avec une vitesse de transmission de 2 Mbps. Finalement dans la région 4 qui couvre la périphérie entre 250 pieds et 350 pieds, la vitesse est réduite à 1 Mbps.

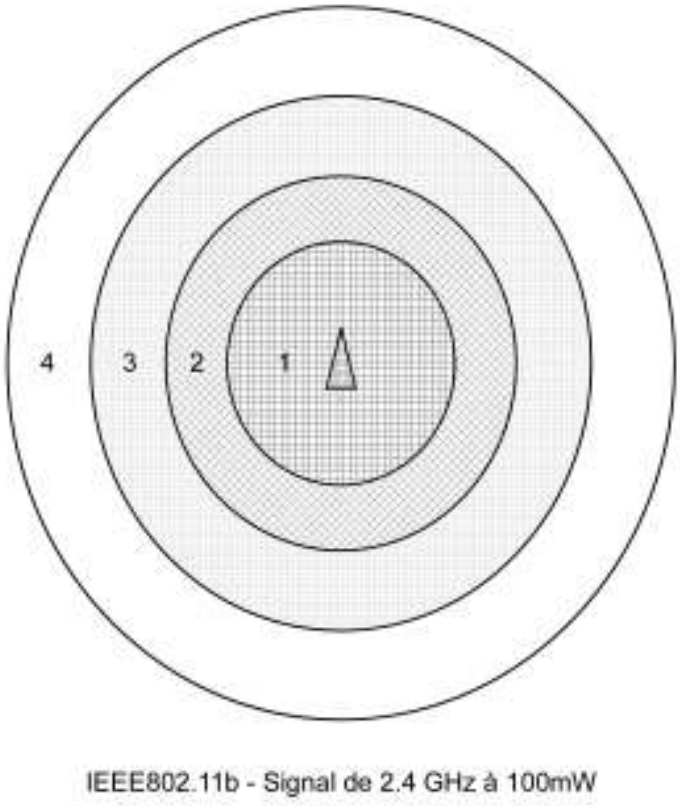


Figure 5.26 Couverture du point d'accès

Pour le IEEE802.11a qui utilise la fréquence de 5GHz le AP à une puissance de transmission de 40 mW et le terminal de 20 mW. Ainsi la vitesse de transmission peut varier de 54 Mbps à 6 Mbps. La table 5.4 donne les détails pour les applications spécifiques des deux types de technologie.

Le niveau de puissance du AP peut être ajusté pour créer de plus petites cellules. Ceci est nécessaire lorsque plusieurs cellules sont adjacentes et très proches l'une de l'autre. La réduction de la puissance évite les interférences de signaux d'un AP à l'autre.

Tableau 5.4 Couverture du point d'accès 802.11a et 802.11b

802.11a AP - 40 mW		802.11b AP - 100 mW	
Rayon en pieds	Vitesse en Mbps	Rayon en pieds	Vitesse en Mbps
60	54	130	11
60-90	48	130-180	5.5
90-110	36	180-250	2
110-125	24	250-350	1
125-135	18		
135-150	12		
150-165	9		
165-300	6		

Nous voyons que la largeur de bande dans le réseau sans fil est en fonction de la couverture des cellules. Les APs peuvent être répartis sur une plus grande superficie comme il est illustré dans la figure 5.27. Par contre, quoique l'itinérance transparente (*seamless roaming*) soit offerte, la largeur de bande varie selon la distance du terminal au AP.

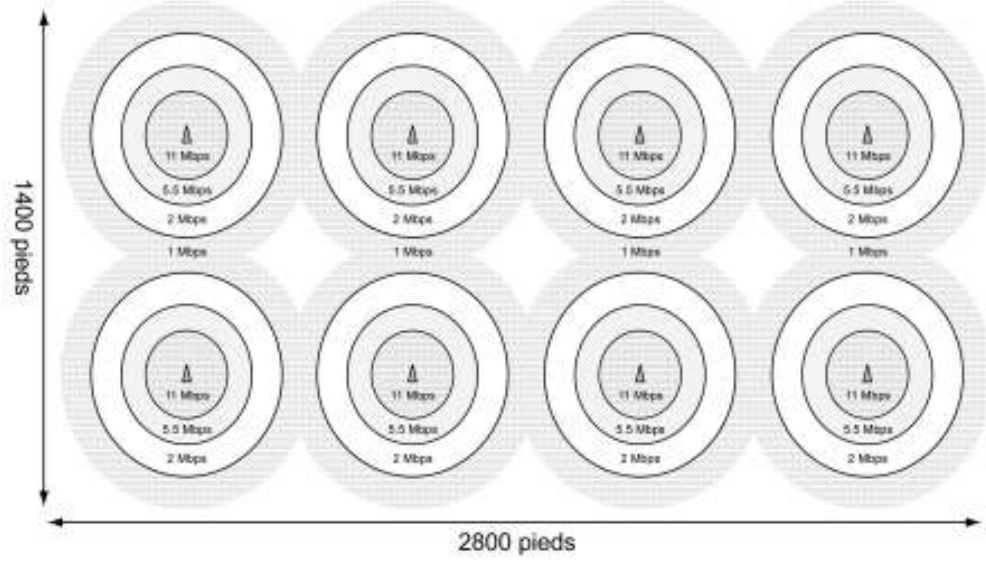


Figure 5.27 Couverture avec vitesses variables pour le 802.11b

Il est toutefois possible de maintenir une vitesse constance à l'intérieur d'un périmètre. Ce dernier est beaucoup plus réduit que dans l'exemple précédent.

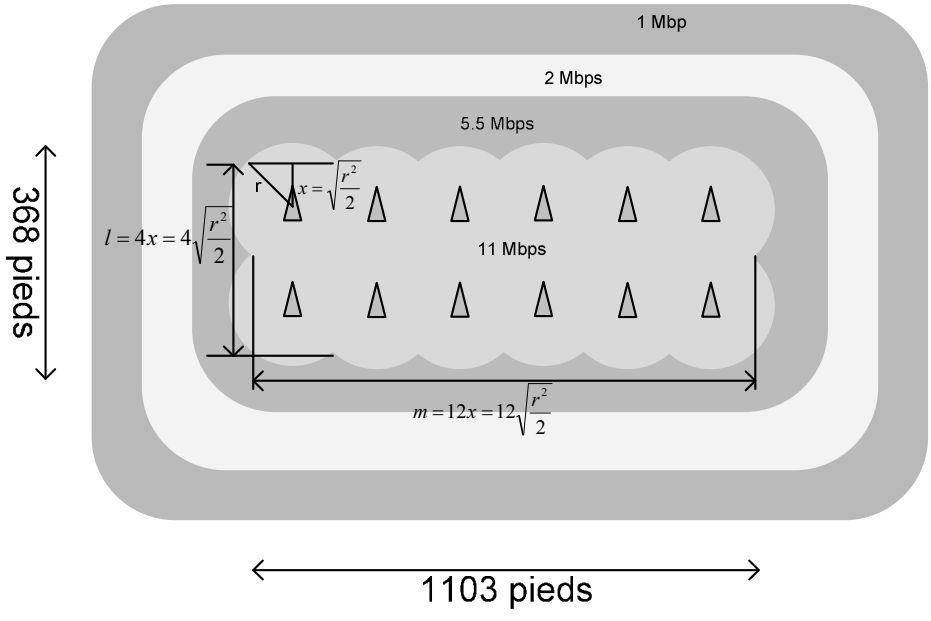


Figure 5.28 Couverture avec vitesses constantes pour le 802.11b

La figure 5.28 illustre cette possibilité. Nous remarquons que les APs sont plus rapprochés de façon à former une seule aire commune où la vitesse est uniforme. De cette façon, l'aire centrale pour tous les APs, maintient un taux de transmission de 11 Mbps alors qu'en périphérie des aires concentriques de 5.5, 2 et 1 Mbps se forment. Le seul désavantage par rapport à l'exemple de la figure 5.26 est que la couverture des cellules est beaucoup plus réduite pour le même nombre de APs.

L'exemple de la figure 5.28 indique une aire large de 368 pieds et longue de 1103 pieds ayant une couverture uniforme de 11 Mbps. Nous supposons ici qu'un quart de la largeur de chaque aire de 11 Mbps de chacun des APs est commun. Puisque le rayon de cette aire est 130 pieds (table 5.4) nous obtenons 390 pieds de largeur. La même supposition est faite sur la longueur.

5.12 Exercices

5.1

Pourquoi l'accusé de réception est transmis le temps d'un SIFS après l'envoi des données et non pas le temps d'un DIFS? Expliquez.

5.2

Si une station désirait prendre le contrôle exclusif du BSS comment aurait-elle pu le faire? Quelles sont les mesures prises par la norme IEEE802.11 pour prévenir ce genre d'agissement?

5.3

Trois stations doivent transmettre chacune un MACPDU. Après trois collisions, le BEB est appliqué par chaque station. La station 1 choisit le chiffre 2 pour le nombre d'intervalles de temps à attendre avant la transmission du MACPDU. La station 2 choisit le chiffre 3 et la station 3 choisit le chiffre 4. La station 1 transmet son PDU à la station 3. La station 2 transmet son PDU à la station 1 et la station 3 transmet son PDU à la station 2. Tracez un diagramme semblable à celui de la figure 5.7 pour illustrer votre réponse.

5.4

Si une station, en réponse au CTS, veut transmettre une trame à une autre station, mais que cette autre station n'est pas visible, comment la station émettrice va-t-elle s'y prendre? Quelle est l'implication de la station réceptrice? L'implication du AP?

5.5

Comment une station peut-elle faire partie de l'invitation à transmettre *polling* par le point de coordination durant la période d'accès PCF et se faire inclure dans sa liste?

5.6

Si deux AP coexistent dans un même BSS et que les deux contiennent des points de coordination, expliquez pourquoi la période sans compétition ne peut pas être entièrement sous le contrôle d'un seul point de coordination. En d'autres termes, il n'est pas possible qu'une fois qu'un point de coordination ait pris le contrôle d'une période sans compétition, qu'un autre point de coordination puisse la lui prendre.

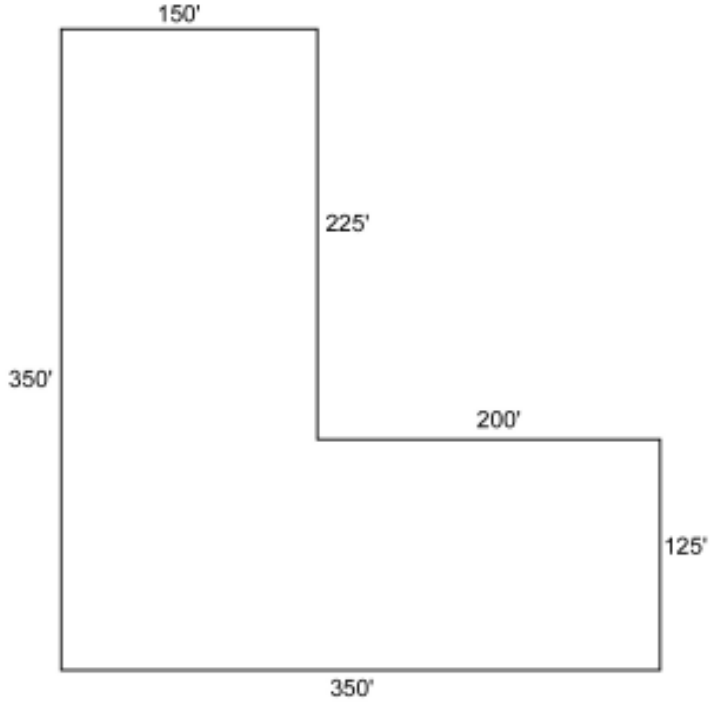
5.7

Quel est le nombre maximum de stations sans fil qu'il pourrait y avoir dans un BSS à infrastructure ? Expliquez pourquoi.

5.8

Un espace de bureau, illustré ci-bas, doit être aménagé avec un réseau 802.11b et avoir une itinérance transparente tout en maintenant partout une vitesse de 11 Mbps.

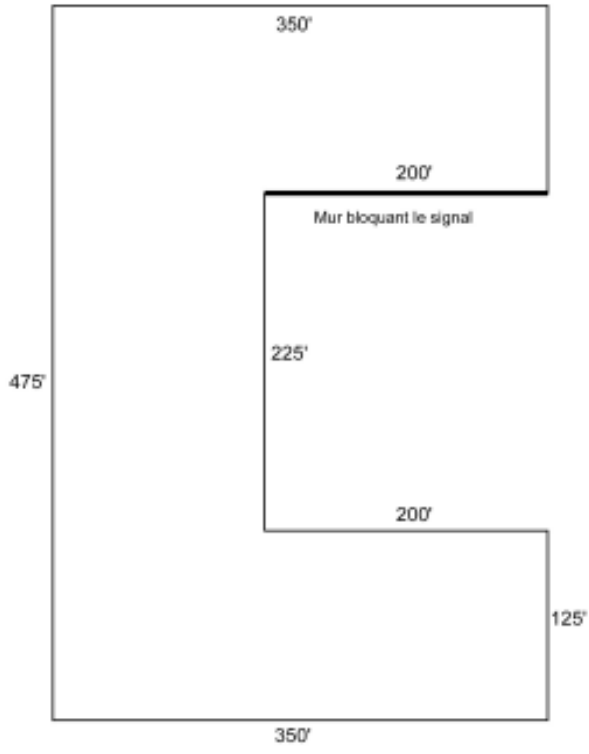
Indiquez où on doit installer les points d'accès si le réseau est à infrastructure.



5.9

Un espace de bureau, illustré ci-bas, doit être aménagé avec un réseau 802.11b et avoir une itinérance transparente tout en maintenant partout une vitesse uniforme. Notez par contre qu'un des murs bloque le signal. Indiquez où on doit installer les points d'accès si le réseau est à infrastructure.

a) Nous désirons obtenir une vitesse uniforme de 11 Mbps partout.
b) Nous désirons obtenir une vitesse uniforme d'au moins 5.5 Mbps partout.



6 Les réseaux à jeton

6.1 Introduction

Une technique qui a évolué en même temps que le réseau Ethernet est celle qui utilise le jeton. Dans ce cas, l'accès est ordonné de telle sorte que chacune des stations sait à quel moment elle peut transmettre et quand elle ne le peut pas. Dans les faits, il ne peut y avoir qu'une seule station qui puisse transmettre à la fois.

Les différents réseaux à jeton sont le bus à jeton (*Token Bus*), l'anneau à jeton (*Token Ring*) et le FDDI (*Fiber Distributed Data Interface*) pour un réseau à débit plus élevé.

Le bus à jeton est un réseau à arborescence. Si son architecture ressemble à celle du réseau Ethernet, son fonctionnement, en diffère totalement.

L'anneau à jeton fait appel à un anneau physique qui lui permet d'interconnecter les stations qu'il dessert. La boucle est simple et unidirectionnelle. Chaque station est reliée aux stations adjacentes par des liens point à point.

Les sous-couches MAC et Physique du bus à jeton sont normalisées dans le texte de la norme IEEE 802.4, tandis que celles de l'anneau à jeton le sont dans celui de la norme IEEE 802.5.

6.2 Le bus à jeton (*Token Bus*)

Le protocole de réseau local qui exploite le bus à jeton tel que sanctionné par la norme IEEE802.4 est utilisé afin de fournir une application appropriée à un environnement manufacturier, industriel, et même parfois en bureautique.

Le MAC du LAN IEEE 802.4 repose sur le bus mais fait appel au jeton pour passer le contrôle d'une station à l'autre. Physiquement, la topologie du bus à jeton est une arborescence de bus sur lesquels les stations sont branchées. Par

contre, celles-ci font partie d'un anneau logique qui les relie dans une séquence ordonnée, de la première station à la dernière et de la dernière à la première. Chaque station est informée de l'identité de la station qui la précède et de celle qui lui succède.

Une trame de contrôle, utilisée comme jeton, se rend d'une station à l'autre en suivant l'anneau logique. Seule la station détentrice du jeton peut transmettre aux autres stations connectées au bus. La station qui transmet peut, pendant un certain laps de temps, conserver le jeton. Si elle n'a plus rien à transmettre ou, si le temps qui lui est alloué a expiré, elle passe le jeton (en transmettant la trame jeton) à la station suivante qui fait partie de l'anneau logique. Il importe de savoir que des stations peuvent être physiquement connectées au bus et recevoir des données sans pour autant faire partie de l'anneau logique.

L'anneau logique ne relie que les stations qui peuvent transmettre. Une station qui ne fait pas partie de l'anneau logique peut recevoir des données, répondre à des demandes précises ou transmettre des acquittements suite à la demande d'une station appartenant à l'anneau. Toutefois, elle ne peut pas amorcer une communication.

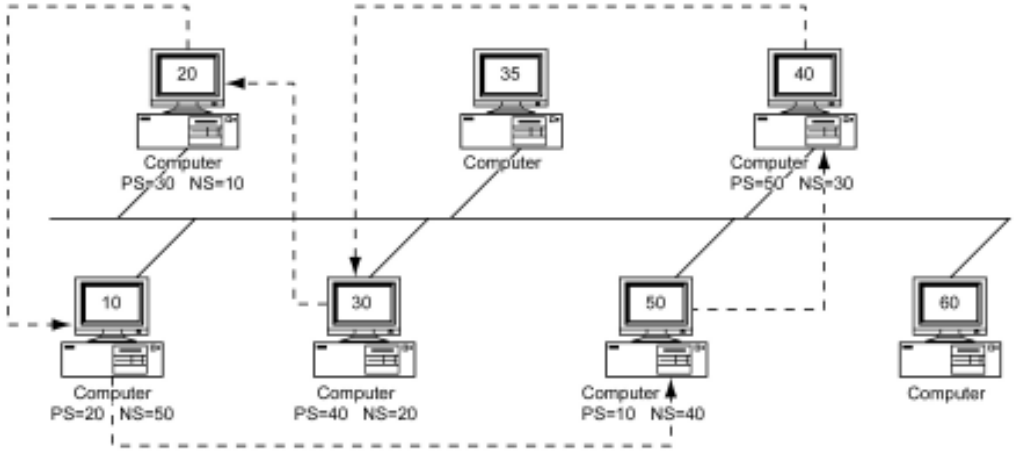


Figure 6.1 Bus à jeton

L'anneau logique se compose des adresses MAC des stations qui en sont membres. La station qui détient le jeton le passe à la prochaine station de l'anneau qui est celle ayant la prochaine plus petite adresse parmi les stations de l'anneau. La figure 6.1 nous donne un aperçu de 5 stations appartenant à l'anneau dont les adresses vont de 10 à 50. Ainsi, si la station 30 a le jeton, elle doit, après avoir fini sa transmission, le passer à la station 20. De la sorte, toutes les stations passent le jeton à la station dont l'adresse qui suit est la plus petite,

sauf dans le cas de la station 10 (la plus petite adresse de tout l'anneau logique) qui transmettra le jeton à la station 50, soit la plus grande de l'anneau logique.

Chaque station comprend deux registres, le PS (*Previous Station*) ou station précédente et le NS (*Next Station*) ou station suivante. Ces registres stockent les adresses des stations adjacentes sur l'anneau logique. Dans notre exemple, la station 30 comprend un PS=40 et un NS=20. De la sorte, chaque station sait d'où vient le jeton et à qui il doit être remis. Par contre, dans notre exemple, la station 10 a son PS=20 et son NS=50. Ceci est nécessaire pour pouvoir boucler l'anneau logique. Les stations 35 et 60 ne sont pas impliquées dans le processus de la transmission puisqu'elles ne font pas partie de l'anneau logique; ce qui, toutefois, ne les empêchera pas de recevoir des données.

Le réseau de bus à jeton est efficace et simple lorsque les stations sont déjà connectées et qu'elles communiquent entre elles. Par contre, certains problèmes peuvent surgir surtout lorsqu'il s'agit d'ajouter une station à l'anneau, d'en retirer une, de réinitialiser l'anneau logique ou de récupérer un jeton perdu. Des fonctions viennent s'ajouter au bus à jeton en vue de trouver une solution à ces problèmes.

6.2.1 Ajout d'une station

Périodiquement, chaque station doit permettre à d'autres ne se trouvant pas sur l'anneau logique, de s'y rattacher. Pour ce faire, la station détentrice du jeton doit être en mesure de disposer de temps pour retenir le jeton et ne plus avoir à transmettre de données. Tout en retenant le jeton, cette station émet alors la trame *solicit-successor*. La raison de l'existence d'une telle trame est de demander aux stations non connectées s'il s'en trouve une qui désire succéder à la présente station. Par contre, il faut que l'adresse de la station à ajouter soit comprise entre l'adresse de la présente station et celle de la station qui lui succède. Les autres stations ne sont pas sollicitées et doivent attendre leur tour. Ainsi, et à titre indicatif si la station 40 permet l'insertion d'une nouvelle station, elle envoie la trame *solicit-successor* pour une station dont l'adresse se trouve entre 40 et 30. Si la station 35, qui a reçu cette trame, décide de se joindre, elle transmet la trame *set-successor* en vue d'indiquer à la station 40 que la station qui va lui succéder sera dorénavant la station 35.

Cependant, il existe quatre choix de réponses possibles pour la trame *solicit-successor* qui sont :

- pas de réponse,

- une réponse par le biais de la trame *set-successor* (tel qu'indiqué dans l'exemple précédent),
- plusieurs réponses (plusieurs stations répondent à la requête) et,
- une réponse invalide.

Lorsqu'il n'y a pas de réponse, la station passe le jeton à la station qui suit immédiatement.

Si la réponse nous parvient sous la forme d'une trame *set-successor*, cela signifie qu'une station unique désire être ajoutée à l'anneau logique. La station avec le jeton met à jour son NS avec la nouvelle adresse MAC après avoir vérifié sa pertinence et lui remet le jeton. La nouvelle station sait qui est son prédécesseur et son successeur puisqu'elle en a été informée par la dernière trame reçue, c'est-à-dire la *solicit-successor* où ces informations figurent.

Si plusieurs stations répondent à la demande, la réponse est une collision. La station avec le jeton va devoir transmettre une nouvelle trame *resolve-contention* afin d'amorcer la résolution de la collision. Cette trame demande aux stations qui ont répondu à la trame *solicit-successor*, de bien vouloir attendre un certain laps de temps en se fondant sur leur adresse, avant de retransmettre la trame *set-successor*.

Le temps d'attente avant la retransmission d'une trame *set-successor* est calculé sur les deux premiers bits de son adresse, dans le cas d'une première retransmission. Ainsi, dans le cas où seules deux stations sont en collision et que les deux premiers bits de l'une sont à 10, et que les deux premiers bits de l'autre sont à 00, la première station (c.-à-d. 10) devra attendre 2 laps de temps avant de retransmettre cependant que la seconde pourra procéder à une retransmission immédiate. De cette manière, la collision sera évitée. Le laps de temps est égal à deux fois le temps de propagation à travers un bus.

Si les deux stations avaient les mêmes deux premiers bits, une nouvelle collision surviendrait puisque les deux stations doivent attendre pendant le même laps de temps pour retransmettre au même moment. Après une nouvelle collision, la station ayant le jeton retransmet un nouveau *resolve-contention*. Les stations qui ont déjà répondu au premier comprennent que les deux premiers bits sont identiques. Ces stations prennent alors les deux bits suivants de l'adresse et reprennent la même procédure. Il arrivera qu'une station parvienne enfin à être la seule à transmettre étant donné que les adresses sont uniques. Il est bon de rappeler qu'une seule station à la fois peut être ajoutée.

Si la trame reçue n'en est pas une du type *set-successor*, on se trouve face à la possibilité de jeton double et la station qui en fait la sollicitation ne passe pas son jeton mais se trouve à l'état d'écoute.

6.2.2 Retrait d'une station

Lorsqu'une station désire se retirer de l'anneau, elle attend tout d'abord de recevoir le jeton. Une fois ce dernier reçu, elle transmet la trame *set-successor* à la station précédente pour l'aviser de changer de nouveau successeur. Ce dernier est en fait la station qui succède à celle qui se retire. Le prédécesseur met à jour sa variable NS afin de savoir quand il recevra le jeton et à qui l'adresser une fois son utilisation faite.

La station qui lui succède n'a pas besoin d'être prévenue du changement parce que toute station qui reçoit un jeton met à jour sa variable PS avec l'adresse de la station qui a transmis le jeton. Lorsque cette station recevra le jeton elle inscrira l'adresse de sa nouvelle station précédente dans la variable PS.

Lorsqu'une station qui, à cause d'une panne, est brusquement retirée, la station qui la précède, soit celle qui lui a transmis le jeton, attendra pendant un espace temps équivalent à deux fenêtres pour s'apercevoir que la station qui lui a succédé n'est plus en service. Passé ce temps d'attente, elle transmet la trame *who-follows*. Cette dernière demande aux stations faisant partie de l'anneau : «*quelle est la station qui précédait la station x*» (la station x étant celle qui n'a pas pris le jeton et qui s'est retirée). La station qui succède à la station x va alors reconnaître l'adresse appartenant à la station précédente. Aussi va-t-elle transmettre la trame *set successor* par laquelle elle indiquera son adresse. Ainsi, la station qui se trouve en possession du jeton met à jour sa variable NS et lui remet le jeton.

6.2.3 Présence de plusieurs jetons

Dans un réseau bus à jeton, il ne peut y avoir qu'un seul jeton. Chaque station sait qu'en détenant le jeton, la permission de transmettre lui est accordée. Toutefois, il peut arriver que plusieurs jetons circulent dans le réseau. Cette situation est inacceptable; raison pour laquelle toute station laisse tomber son jeton lorsqu'elle s'aperçoit qu'il y a un autre jeton. Dans certains cas, une telle situation pourrait générer la perte de tous les jetons du réseau et amorcer une procédure de récupération du jeton ou une initialisation de l'anneau.

6.2.4 Initialisation de l'anneau

La phase initiale du démarrage de l'anneau logique est son initialisation. Lorsque le système est activé, aucune station ne détient de jeton; en fait, il n'y en a tout simplement pas. Chaque station a une horloge qui lui permet de détecter le manque d'activités sur l'anneau après une certaine période de temps. La station qui s'aperçoit la première qu'il n'y a pas de jeton va transmettre la trame *claim-token*. Si plusieurs stations transmettent la même trame en même temps, il y aura collision. La résolution d'une telle collision est en tous points identique à celle traitée dans l'ajout de station; c'est-à-dire qu'elle se fonde sur l'adresse de chacune des stations pour établir le temps d'attente avant la retransmission d'une nouvelle trame *claim-token*. La station qui attend et entend une activité sur la ligne se retire et ne réclame pas le jeton. La station qui réussit à transmettre sans collision va obtenir ce nouveau jeton.

6.2.5 Format de trame du IEEE 802.4

La trame du protocole de bus à jeton est illustrée à la figure 6.2. Elle ressemble partiellement à celle d'un CSMA/CD, mais sa fonctionnalité en est fort différente.

La trame comprend un préambule pour permettre la synchronisation à la réception. Il y a aussi un champ permettant de savoir où débute la trame, le SD (*start delimiter*) codé NN0NN0. N est le code Non data, c'est-à-dire qui ne représente ni 0 ni 1.

préambule	SD	FC	DA	SA	Données	FCS	ED

Figure 6.2 Trame du bus à jeton

Le champ FC (*Frame Control*) détermine le type de trame. Le FC peut indiquer que la trame transporte dans les données une trame LLC telle qu'indiquée à la figure 6.3, lorsque les deux premiers bits du FC sont à 01. Dans ce cas, le champ FC comporte d'autres informations dont entre autres la priorité de la trame.

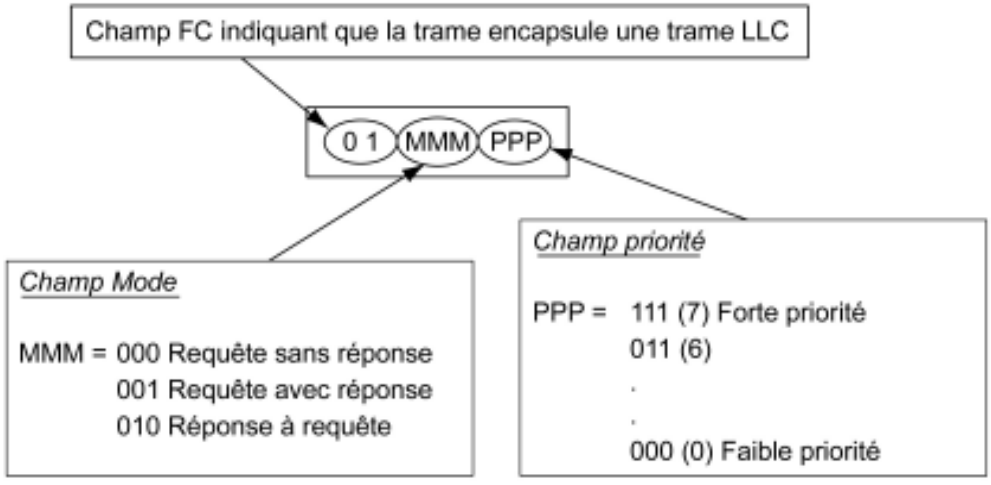


Figure 6.3 Champs FC pour données LLC

Lorsque les deux premiers bits de FC sont codés 00, c'est une trame de contrôle, codée conformément au format illustré à la figure 6.4.

Il a été auparavant fait mention des trames de contrôle dans les différentes procédures du bus à jeton (ajout de station, retrait de station etc.). Le champ *adresse* nous fournit les adresses MAC tant source que destination. Le FCS (*Frame Check Sequence*) est utilisé pour la détection d'erreurs au niveau MAC. Il s'agit encore là d'un CRC ayant recours à un polynôme de 32 bits.

Le dernier champ, ED (*end delimiter*) signale la fin de la trame. Ce champ n'est pas vérifié par le FCS lors d'erreur de transmission, raison pour lequel son dernier bit E peut indiquer s'il y a eu erreur lors de sa transmission.

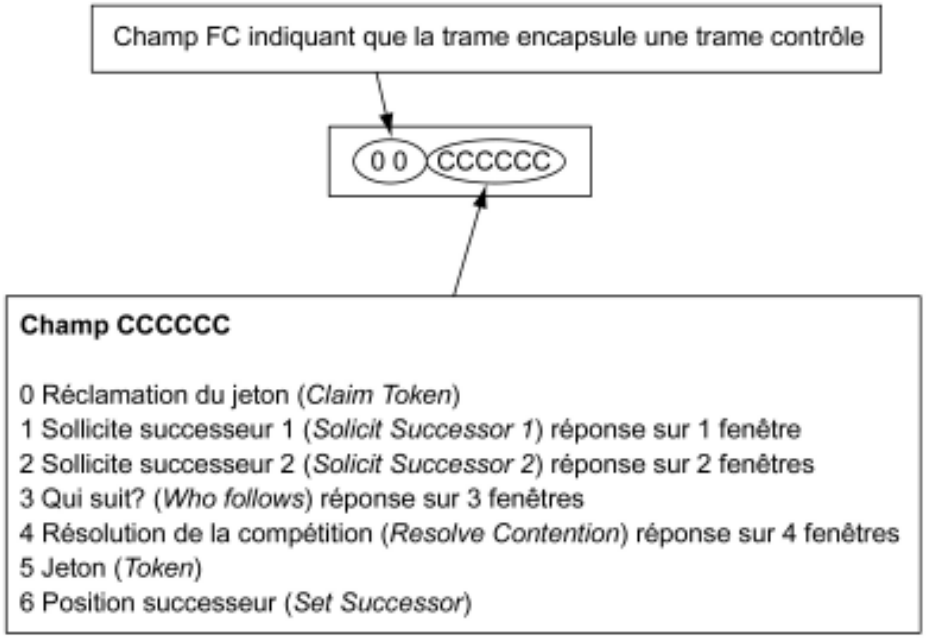


Figure 6.4 Champ FC du bus à jeton pour contrôle

6.2.6 Assignation des priorités

Les priorités des trames sont assignées au champ FC lorsque celui-ci indique le transport de données. Trois bits (PPP) sont utilisés pour faire état de cette priorité. Ces dernières sont assignées selon la classe de service des PDUs transmis. Celles-ci sont assignées par le LLC. Le LLC assigne une classe de service se trouvant entre 0 et 7 à chacun des PDUs. Le MAC reçoit ledit PDU et lui assigne une valeur prioritaire connue sous le nom de classe d'accès dont la valeur peut être de 0, 2, 4, ou 6, selon la classe de service du LLC. L'assignation se fait comme suit :

Classe de service	Classe d'accès
0 1	0
2 3	2
4 5	4
6 7	6 priorité la plus élevée.

Le système de priorité est optionnel. Dans le cas où l'on y a recours, le MAC met les PDUs dans quatre files d'attentes en se fondant sur leur classe d'accès. Les premiers PDUs à être traités sont ceux de la classe d'accès 6. Ils sont transmis par la station qui a le jeton jusqu'à ce qu'il n'y en ait plus ou jusqu'à expiration du

temporisateur ***High priority token hold time.*** Les PDUs de la classe d'accès 4 suivent dans la mesure où il reste encore du temps à la station. Ces PDUs sont transmis jusqu'à ce qu'il n'y en ait plus ou jusqu'à expiration du temporisateur ***Target token rotation time 4.*** Le même raisonnement est appliqué aux autres classes d'accès.

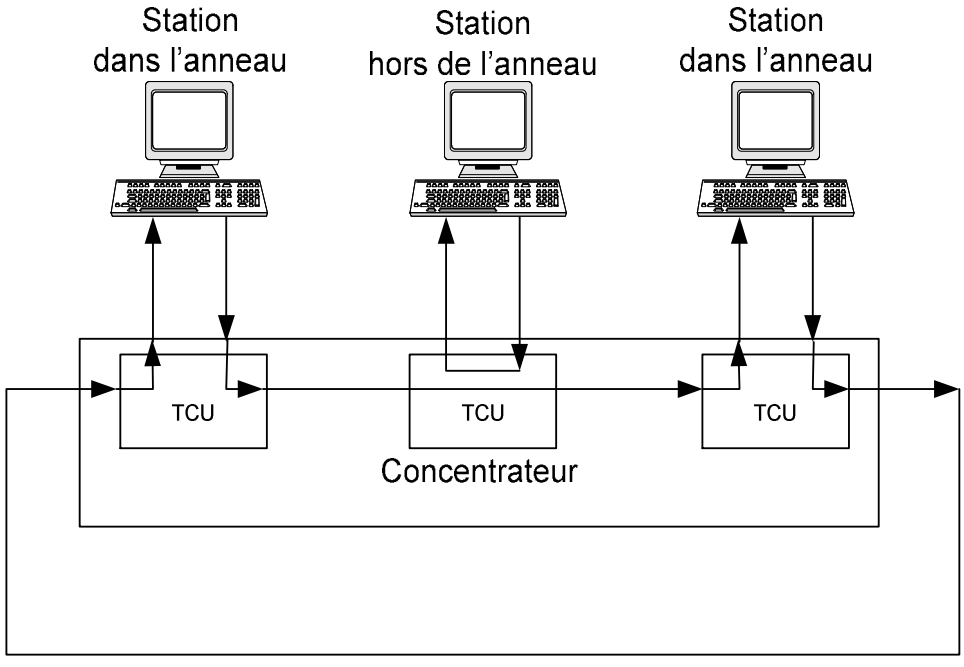
L'encodage utilisé dans le bus à jeton est l'encodage Manchester déjà traité au chapitre 4. Dans l'encodage Manchester le Non Data correspond à l'impulsion qui n'a pas de transition au milieu du bit.

6.3 L'anneau à jeton

Contrairement au bus à jeton, l'anneau à jeton a un anneau physique pour le passage du jeton. Mais comme le bus à jeton, l'anneau à jeton est basé sur le passage d'un jeton d'une station à l'autre. Une station qui veut transmettre doit attendre de recevoir le jeton. La station qui reçoit le jeton le retient et transmet ses trames jusqu'à expiration du temps qui lui est alloué ou jusqu'à ce qu'il ne lui reste plus rien à transmettre. Elle transmet alors le jeton, mais seulement après avoir reçu le début de la dernière trame transmise (la trame circule dans l'anneau et revient à la station transmetteur).

Chaque station qui reçoit une trame en vérifie l'adresse de destination. Si c'est la sienne, elle copie la trame. Dans tous les cas, elle retransmet la trame dans l'anneau.

Le protocole de la couche MAC et de la couche Physique de l'anneau à jeton est normalisé dans le texte de la norme IEEE 802.5.



TCU - Coupleur d'interface (*Trunk Coupling Unit*)

Figure 6.5 Exemple de câblage d'anneau à jeton

Grâce à l'anneau physique, l'anneau à jeton n'éprouve pas les problèmes du bus à jeton pour l'ajout ou le retrait de station comme l'indique la figure 6.5. Une station peut se retirer de l'anneau sans avoir à suivre une procédure particulière. Quoiqu'il en soit, le jeton va continuer de circuler le long de l'anneau et chacune des stations qui y sont reliées pourra le lire et le copier.

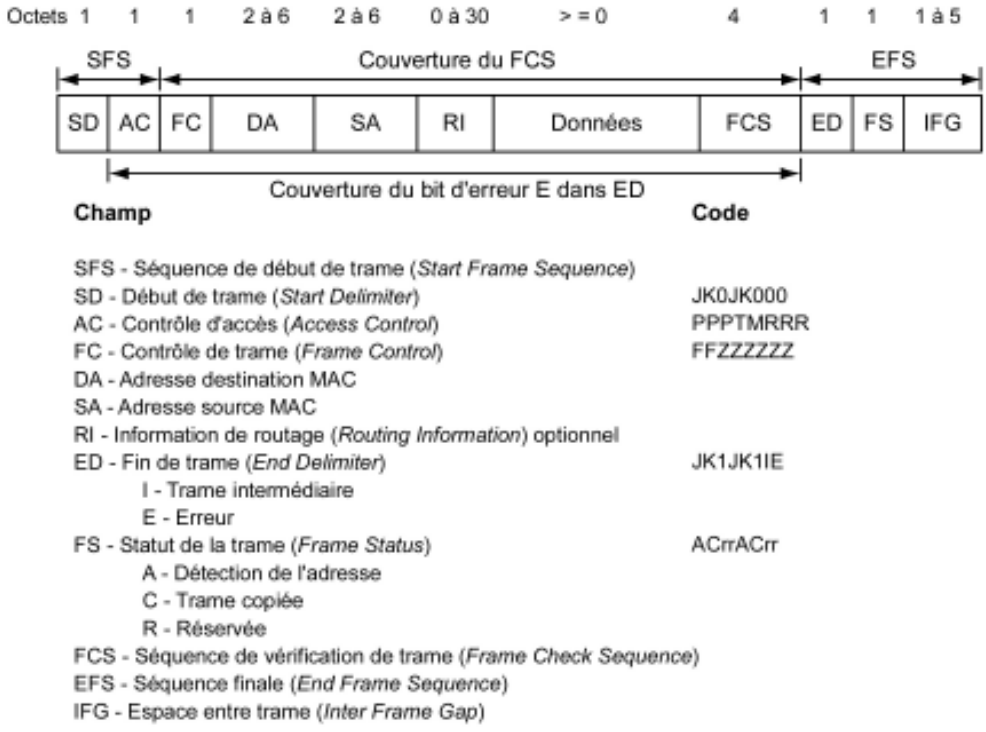


Figure 6.6 Trame de l'anneau à jeton

La trame de l'anneau à jeton indiquée à la figure 6.6 comporte plusieurs champs expliqués ci-dessous :

Le champ SD (*Start Delimiter*) permet d'établir le début de la trame. Il est codé comme suit : JK0JK000. Les codes J et K sont des Non Data, opposé l'un de l'autre. Ces codes seront traités ultérieurement.

Le champ AC (*Access Control*) est un champ de contrôle d'accès. Il est composé de 8 bits. Les trois premiers sont les indicateurs de priorité du jeton. Le quatrième bit nous signale si la trame est un jeton ou non. Le cinquième bit est utilisé par la station moniteur que l'on verra un peu plus loin dans le cadre de ce chapitre. Enfin, les trois derniers bits sont utilisés pour procéder à la réservation du jeton ce dont nous reparlerons plus loin également. Si la trame est un jeton, elle se limite aux champs indiqués dans la figure 6.7.

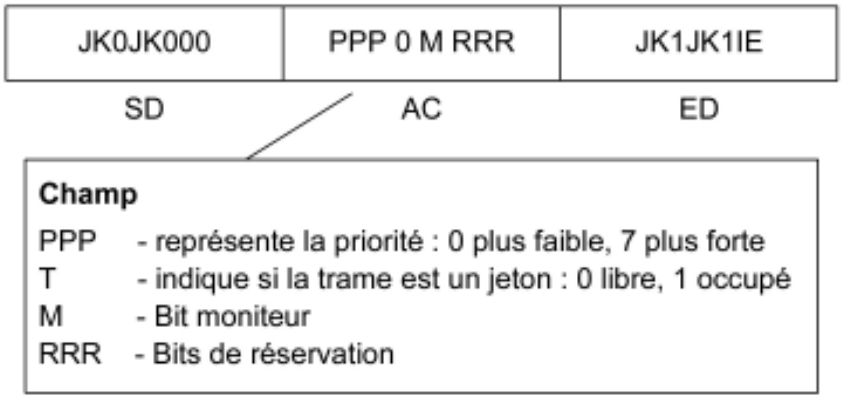


Figure 6.7 Format du champ contrôle d'accès (Access Control) pour le jeton

Le champ Contrôle de trame FC (*Frame Control*) indique la nature de la trame. Ainsi, lorsqu'ils sont codés à 01, les deux bits signifient que la trame transporte des données comprenant le PDU LLC. Dans ce cas, les trois derniers bits de ce champ établissent la priorité de la trame.

Lorsque les deux premiers bits sont codés à 00, cela signifie que la trame en est une de contrôle MAC. Les fonctionnalités de ces différents types de trames MAC, sont illustrées à la figure 6.8. Nous en traiterons plus loin dans ce chapitre.

Type de trame	FF ZZZZZZ	Définition
LLC	01 000 PPP	Priorité de la trame
MAC	00 00 0000	Test adresse double
	0001	Express buffer
	0010	Beacon
	0011	Réclamation du jeton (*Claim Token*)
	0100	Purge de l'anneau (*Ring Purge*)
	0101	Monitor actif (*Active Monitor Present*)
	0110	Monitor reserve (*Standby Monitor Present*)
	1X XXXXXX	Réservés

Figure 6.8 Types de trames de l'anneau à jeton

Les champs suivants de la trame de l'anneau à jeton comprennent les adresses MAC source et destination. L'adresse destination, telle qu'indiquée à la figure 6.9, comprend 6 octets ou 48 bits. Le premier bit de cette adresse indique si c'est une

adresse individuelle ou de groupe. Le deuxième bit indique si l'adresse est locale ou universelle.

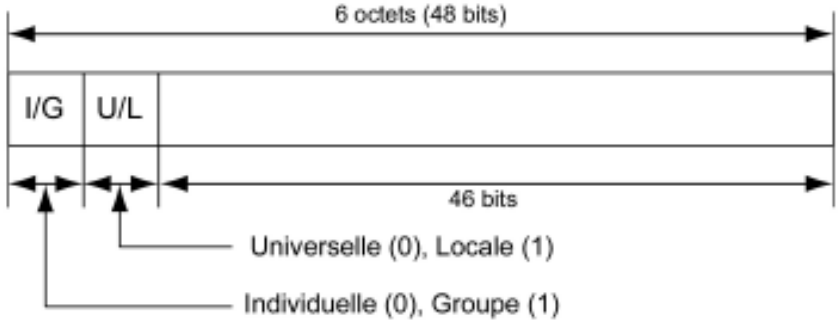


Figure 6.9 Champ d'adresse destination

L'adresse source illustrée à la figure 6.10 identifie la station d'origine. Le premier bit du champ de l'adresse source est l'indicateur RII (*Routing Information Indicator*). Lorsqu'il est à 1, ce bit indique qu'il y a un champ d'information de routage qui suit. Le deuxième bit fait mention de la nature de l'adresse source : locale (1) ou universelle (0).

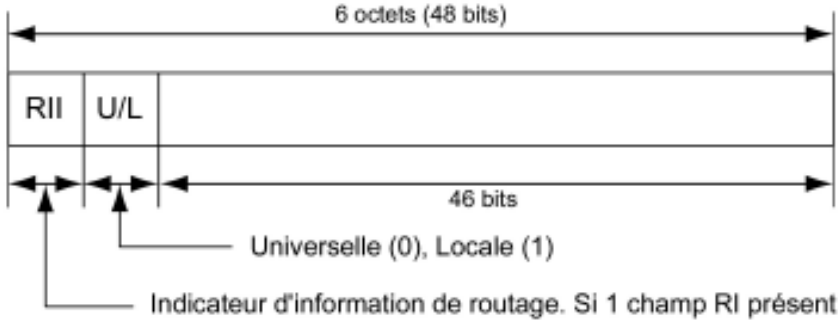


Figure 6.10 Champ d'adresse source

Le champ information de routage (*Routing Information – RI*) qui pourrait suivre si le bit RII est à 1, peut avoir une longueur de 2 à 30 octets. Ce champ est utilisé par l'algorithme de routage de source (*Source Routing*).

Ces champs sont suivis d'un champ de données lorsque ce dernier est requis. Le FCS est annexé à la trame afin de la protéger contre des erreurs de transmission. Quant aux deux derniers champs. Le ED (*end delimiter*), est utilisé pour signaler la fin de la trame ainsi que la segmentation des données (figure 6.11), et le FS

(*Frame Status*) nous fournit des informations à propos des actions exercées par le récepteur.

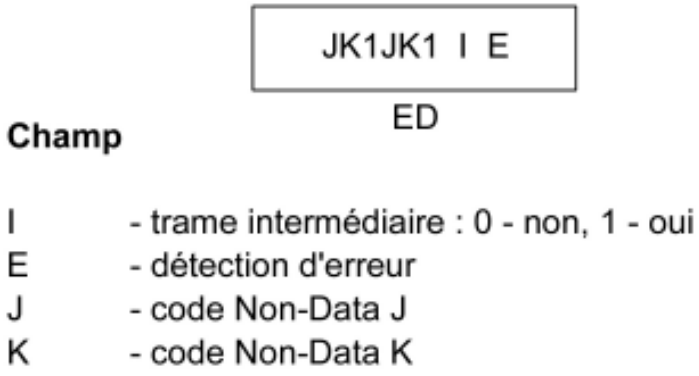


Figure 6.11 END DELIMITER

Il est bon de souligner que le sixième bit du ED nous signale la segmentation, c'est-à-dire s'il y a d'autres trames qui composent ces données, ou si c'est le dernier segment transmis ou encore le seul. Le dernier bit de ce champ permet de savoir s'il y a eu détection d'erreurs de la trame par une autre station. Lorsque ce cas se présente, les stations suivantes n'ont plus besoin de perdre du temps à la vérifier.

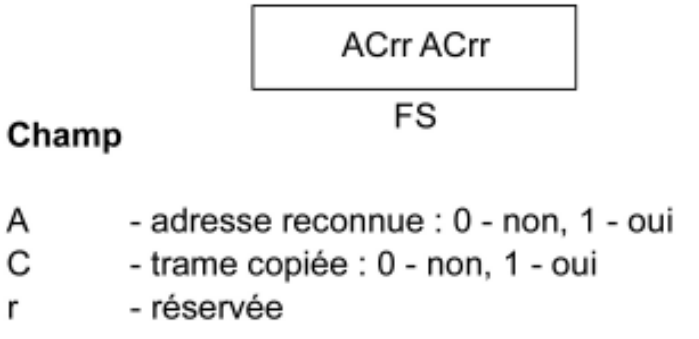


Figure 6.12 FRAME STATUS

Les bits A font état de ce que l'adresse a été reconnue par la destination et les bits C, que la trame a été copiée par la destination (Cf. figure 6.12).

Nous remarquons que le champ FS comprend deux copies de **A** et de **C.** Ce cas est pratiqué parce que l'octet *Frame Status* est transmis après le FCS et il n'y a pas de vérification d'erreurs sauf pour la retransmission de ces informations dans le même octet.

6.3.1 Le moniteur actif

Afin d'éviter que des trames puissent circuler indéfiniment dans l'anneau, une station se voit assigner le rôle de moniteur actif «pour *Active Monitor*». Toutes les trames transmises font le tour complet de l'anneau. Toute nouvelle trame passant par le moniteur actif verra son bit M du champ «contrôle d'accès» mis à 1 par ce dernier.

Si le moniteur actif détecte une trame dont le bit M est à 1 avant de la marquer, cela signifie que cette trame a déjà circulé autour de l'anneau. Elle est donc détruite sans plus attendre.

Le moniteur actif est choisi grâce à une procédure du protocole. Pour signaler sa présence, le moniteur actif transmet des trames AMP «pour *Active Monitor Present*». Si ces trames cessent d'être transmises après un certain temps déterminé par le temporisateur TAM (*Timer Active Monitor*) dont la valeur varie de 6,8 à 7 secondes, les autres stations, connectées à l'anneau et qui sont également des moniteurs potentiels, prennent part à l'élection du nouveau moniteur actif de l'une d'entre elles. La priorité est octroyée par l'adresse MAC. La station ayant la plus grande adresse MAC jouit d'une plus grande priorité. Cette station assume alors le rôle de moniteur actif.

Une fois choisi, le moniteur actif purge l'anneau en transmettant la trame RP « pour *Ring Purge* ». Les stations qui la reçoivent réinitialisent leurs temporisateurs et attendent le jeton. Il incombe au moniteur actif de transmettre le premier jeton.

Lorsque le moniteur actif détecte l'absence du jeton pendant la période de son temporisateur TRP (*Timer ring purge*) dont la valeur varie de 1 à 1,2 secondes, il purge l'anneau.

6.3.2 Perte de jeton

Lorsqu'une une station s'aperçoit de la perte du jeton en raison d'un délai trop long (2,6 secondes en moyenne) sans qu'il y ait mention de réception du jeton par le temporisateur TNT (*Timer, no token*) et donc du mal fonctionnement du moniteur actif, la station transmet alors une trame *claim-token*. Cette trame

connaît l'adresse de la station qui en fait la demande et elle circule dans l'anneau de station en station. Si, au cours de son parcours, une autre station qui reçoit le jeton décide qu'elle a une plus grande priorité que celle qui en a fait la demande, elle remplace l'adresse du *claim-token* par la sienne.

La station qui a émis ou modifié un *claim-token*, et qui revoit ce *claim-token* avec son adresse, détruit cette trame et crée le jeton pour le transmettre sans plus attendre.

6.3.3 La procédure balisage (Beacon)

La station qui décèle la perte du signal et qui ne détecte aucune trame se met alors à transmettre des trames balisage «*beacon*». Elle transmet ces mêmes trames lorsque des trames «*claim token*» sont transmises avec une fréquence plus élevée. Ce phénomène est le signal à l'effet qu'il y a détection d'un problème majeur.

La trame balisage « *beacon* » comprend, dans ses champs, la cause du problème ainsi que l'adresse MAC de la station voisine de la station qui l'a créée (figure 6.13). Notons que l'adresse MAC de la station voisine est connue grâce à la procédure « *neighbor notification* » amorcée par le moniteur actif.

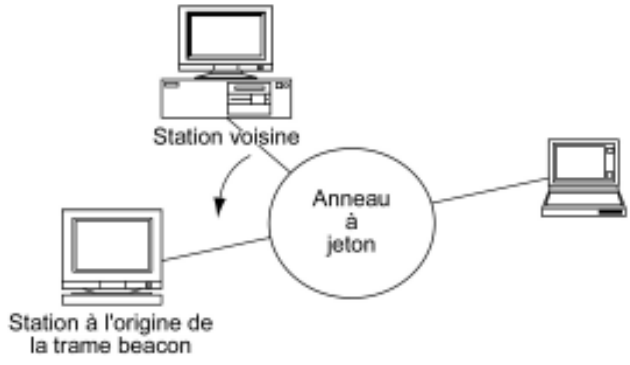


Figure 6.13 Procédure balisage (*Beacon*)

La station qui reçoit une trame balisage indiquant son adresse MAC est donc voisine de la station d'origine. Elle se retire de l'anneau et entame des procédures de tests pour savoir s'il y a problème.

Si le problème est résolu, la station achemine la trame balisage. Si le transmetteur reçoit sa propre trame balisage, cela signifie que le problème est réglé. La station entame alors la procédure « *claim token*. »

Par contre, si le problème persiste et que le temps alloué à la procédure a expiré après l'écoulement de 18 secondes en moyenne accordées par le temporisateur TBT (*Timer beacon transmit*), alors, la station qui a entamé la procédure se retire de l'anneau et à son tour, elle exécute des tests pour savoir si elle a un problème.

6.3.4 Priorité

Le protocole propre à l'anneau à jeton peut fonctionner avec ou sans priorité. **Sans priorité** le jeton passe de station en station conformément à leur ordre dans l'anneau. **Avec priorité**, la station qui reçoit le jeton ne peut transmettre que les trames ayant une priorité plus élevée ou égale à celle du jeton.

La station qui désire faire prévaloir sa priorité en fait la demande. Elle utilise le champ Request (RRR indiquant les trois bits) dans le champ contrôle d'accès de toutes les trames qui passent et accorde la priorité aux trames à transmettre. Si une autre station désire également faire sa réservation, elle ne peut mettre à jour le champ Request (RRR) du jeton que si sa priorité est plus élevée que celle déjà contenue dans le champ RRR.

Une fois libéré, le jeton est transmis en mettant dans son champ PPP la dernière valeur reçue du champ RRR. Seule la station suivante qui a au moins cette priorité en attente, peut prendre le jeton ; autrement, le jeton est relayé.

6.3.5 Couche physique de l'anneau à jeton

La couche physique de l'anneau à jeton telle que mentionnée dans la norme IEEE 802.5 fait appel à l'encodage « *Differential Manchester* » tel qu'illustré à la figure 6.14.

Le « *Differential Manchester* » se fonde non pas sur des transitions fixes pour indiquer le 0 ou le 1, mais bien plutôt sur le nombre de transitions. Le 1 est indiqué par une seule transition au milieu du bit. Ceci peut être fait aussi bien en partant du niveau positif au négatif, qu'en procédant à l'inverse. Par contre, le 0 est indiqué par une transition double à l'intérieur de l'espace du bit.

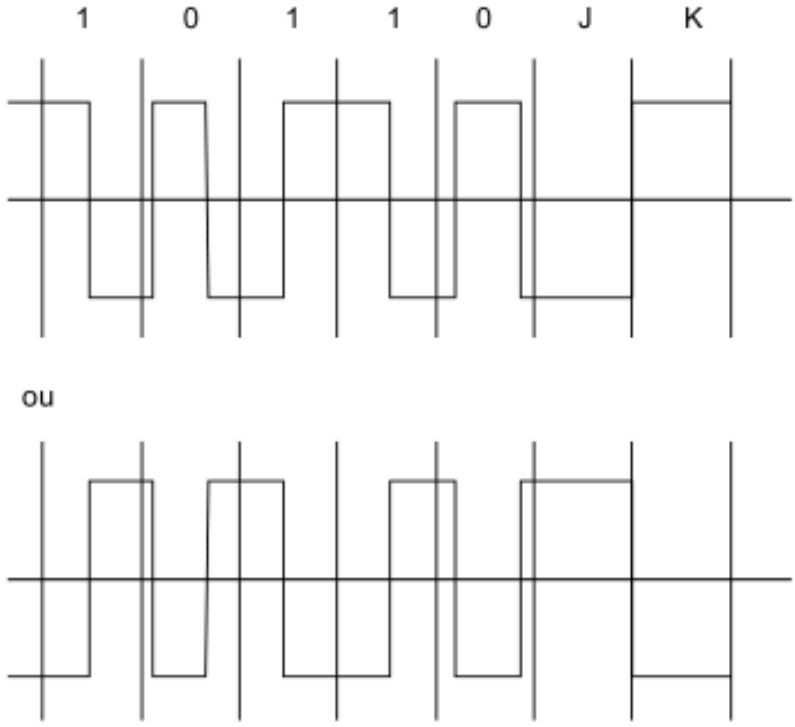


Figure 6.14 Differential Manchester

J et K sont des codes Non Data parce qu'il n'y a aucune transition à l'intérieur du bit.

6.4 Le réseau FDDI

L'anneau à jeton a certaines vertus en matière de fiabilité et de robustesse. Cependant, à cause du traitement des trames par chacune des stations et en raison de la complexité du protocole, le débit peut s'en trouver ralenti. Ainsi et en vue d'améliorer le débit, certaines modifications sont indispensables. Ces changements sont assez importants pour devoir concevoir un nouveau protocole. Le FDDI (*Fiber Distributed Data Interface*) a été conçu afin de remédier à ces lacunes et offrir du même coup un réseau à haut débit.

Le FDDI a été considéré par le IEEE comme norme potentielle pour le développement du réseau MAN (*Metropolitan Area Network*). Plusieurs compagnies y ont contribué. Toutefois, lorsque le IEEE s'est penché sur un autre protocole pour son réseau MAN (le DQDB pour *Distributed Queue Dual*

Bus), ANSI (*American National Standards Institute*) l'a pris en main et en a fait une norme.

La différence avec les réseaux à jeton traditionnels réside dans le fait que dans le cas du FDDI, le temps pris par le jeton pour faire le tour de l'anneau est évalué par chacune des stations de manière précise, et cette mesure est utilisée pour déterminer le temps d'utilisation du jeton.

Dans un réseau FDDI l'organisation des stations peut être faite d'après une topologie en anneau de fibre optique double ou simple. Un réseau FDDI peut répondre à un maximum de 500 stations. La longueur maximale de fibre entre chacune de ces stations est de deux kilomètres. Le type de fibre utilisée est la fibre *multi mode*, ce qui en rend le câblage plus économique. Cependant les fibres *monomodes* peuvent être utilisées. Dans ce dernier cas, la distance entre les stations peut atteindre 60 kilomètres. Toutefois, le réseau FDDI peut être déployé sur une distance maximale de 200 kilomètres. Si la distance entre les stations est de moins de 500 mètres, il est possible d'avoir recours à des composantes optiques plus économiques.

La vitesse normalisée du FDDI est de 100 Mbps par anneau. Les trames du FDDI peuvent avoir une grandeur maximale de 4500 octets. Cette longueur inclut l'en-tête ajouté au moment de l'encapsulation.

6.4.1 Composantes du FDDI

La norme FDDI se compose de quatre parties telles qu'illustrées à la figure 6.15 :

- Medium Access Control (MAC)
- Physical protocol (PHY)
- Physical Medium Dependent (PMD)
- Layer Management (LMT)

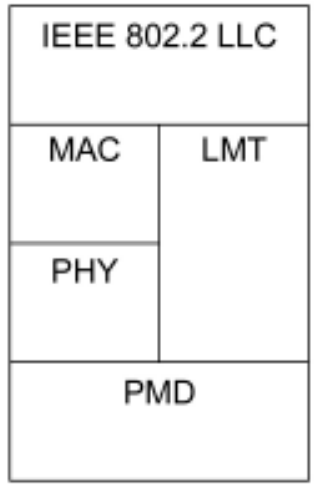


Figure 6.15 Composants de la norme FDDI

Le MAC précise les règles que toutes les stations du réseau utiliseront et partageront en regard de la largeur de bande. Il définit les trames ainsi que les procédures qui composent ce protocole.

Le PHY est la partie de la couche Physique qui ne dépend pas du médium utilisé. Il en précise les algorithmes qui détectent les erreurs, prennent en considération les différences d'horloge, encodent les bits transmis et décodent ceux qui sont reçus.

Le PMD s'occupe de la fibre, du connecteur, du transmetteur et du récepteur utilisés. Le PMD est propre au médium utilisé. C'est la seule partie de la norme qui varie selon le type de câble utilisé.

Le LMT précise les protocoles et formats de trames utilisés en regard de la localisation automatique des fautes, de la configuration du réseau et de son initialisation.

6.4.2 Le protocole MAC du FDDI

Le protocole de FDDI est un protocole à jeton. Ce dernier est transmis de station en station et, seules celles détenant le jeton peuvent transmettre en tout temps. La station émettrice est responsable de la destruction non seulement de ses trames mais aussi de toutes les autres trames qu'elle reçoit après leur parcours autour de l'anneau.

Lorsqu'une station a cessé de transmettre, elle fait suivre le jeton directement après la dernière trame, même si elle n'a pas reçu le début de sa dernière trame comme dans l'anneau à jeton.

Chacune des stations qui reçoivent une trame est tenue d'en vérifier l'adresse de destination. S'il s'agit de la sienne, elle copie la trame. Dans tous les cas elle retransmet la trame dans l'anneau.

Tout comme l'anneau à jeton, grâce à l'anneau physique, une station FDDI peut se retirer de l'anneau sans devoir suivre une procédure particulière. Quoiqu'il en soit, le jeton va circuler le long de l'anneau et chacune des stations qui y est reliée, pourra le lire et le copier.

Il n'existe pas de station monitrice dans le FDDI. La responsabilité de toutes les stations qui détiennent le jeton est de détruire toute trame reçue.

SFS - Séquence de début de trame (*Start Frame Sequence*)
PA - Préambule*Synchronise la trame avec l'horloge de chaque station. Composé de 16 symboles idle (11111)*
SD - Délimiteur de début de la trame (*Start Delimiter*)
FC - Contrôle de trame (*Frame Control*) : FFZZZZZZ
DA - Adresse destination (*Destination Address*)
SA - Adresse source (*Source Address*)
ED - Délimiteur de fin de trame (*Ending Delimiter*) : JK1JK1IE, I - trame intermédiaire, E - erreur
FS - Statut de la trame (*Frame Status*) : ACrrACrr, A - adresse détectée, C - trame copiée, rr - réservé
FCS - Séquence de vérification de la trame (*Frame Check Sequence*)
EFS - Séquence de fin de trame (*End Frame Sequence*)

Figure 6.16 Trame FDDI

La trame du FDDI (Figure 6.16) comprend les champs suivants :

Le **préambule**, permet de procéder à la synchronisation de la trame. Le **SD** (*Start Delimiter*), permet de savoir où débute la trame. Il est codé par les symboles J et K. Ces codes sont qualifiés de Non Data et ne représentent ni 0 ni 1. Ces symboles font partie d'un codage de bits qui est décrit dans la section 6.4.5 et illustrés à la figure 6.17.

Le champ suivant de la trame ou contrôle de trame **FC** (*Frame Control*), est un champ de contrôle d'accès. Il est composé de 8 bits et fait état du genre de trame. Les deux premiers bits soit C et L sont normalement associés aux trames « data ». Le bit C désigne la classe de service de la trame. Si C est à 0, la trame est du type asynchrone mais si C est à 1, la trame est qualifiée de synchrone. Le bit

qui suit nous précise la longueur des adresses. Ainsi, lorsque L est égal à 0 cela signifie que les adresses ont une longueur de 16 bits, par contre, si L est égal à 1, la longueur de l'adresse est de 48 bits. Les deux bits suivants signalent que la trame transporte des données comprenant le PDU LLC lorsqu'ils sont codés à 01. Dans ce cas, le quatrième bit n'est pas utilisé et les trois derniers bits font état de la priorité de la trame.

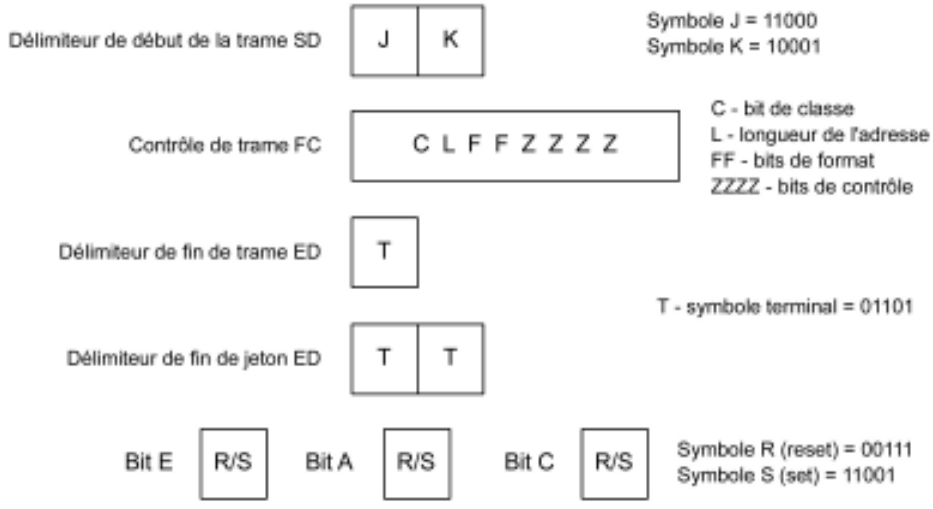


Figure 6.17 Définition des champs

Lorsque les deux bits suivant CL sont codés à 00, cela signifie que c'est une trame de contrôle MAC. Les fonctionnalités de ces différents types de trames MAC sont illustrées à la figure 6.18.

Type de trame	CL FF ZZZZ	Description
Void	0X 00 0000	logiquement ce n'est pas une trame, à ignorer
Nonrestricted token	10 00 0000	pour synchrone et non restreint asynchrone
Restricted token	11 00 0000	pour synchrone et restreint asynchrone
Trames LLC		***Trames data***
Asynchrone	0L **01** rPPP	avec priorité PPP
Synchrone	1L **01** rrrr	r est réservé pour version future
Trames MAC		***Trames de contrôle***
MAC	1L**00** 0001 à 1111	utilisée par le protocole MAC
MAC Beacon	1L**00** 0010	indique une panne sérieuse
MAC Claim	1L**00** 0011	pour déterminer la station qui crée le jeton
Trames réservées	CL **10** r000 à r111 CL **11** rrrr	réservées pour version future

Figure 6.18 Types de trames de contrôle FDDI

Contrairement à l'anneau à jeton, le jeton FDDI comprend son propre champ FC (Cf. figure 6.19). Le jeton ou la trame qui le représente n'est composé que de trois octets (en dehors du préambule) représentés par six symboles.

PA	SD	FC	ED

Figure 6.19 Format du jeton

La trame FDDI comprend les adresses MAC source et destination. Celles-ci sont suivies d'un champ de données si nécessaire. Le FCS est annexé à la trame afin de la protéger des erreurs de transmission. Quant aux deux derniers champs, le délimiteur de fin de trame ED (*End Delimiter*) est utilisé pour indiquer la fin de la trame, et le statut de la trame FS (*Frame Status*) nous donne certaines informations sur les actions du récepteur. Ainsi dans A, le symbole SET indique que l'adresse a été reconnue par la destination. Dans C, le symbole SET signale que la trame a été copiée par la destination (Cf. figure 6.17).

6.4.3 Allocation de capacité

Le FDDI permet la transmission de trafic asynchrone en rafale tout comme la transmission de données synchrones. Les deux types de trafics définis sont: synchrone et asynchrone. Afin de contrôler le partage des ressources dans le réseau, une partie de la capacité est alloué à chaque station. Chacune d'entre elles contrôle son temps de transmission ou le temps durant lequel elle garde le jeton. Chaque station peut avoir un temps fixe réservé à la transmission de trames synchrones. Si le temps alloué le permet, il servira à la transmission de trames asynchrones.

Afin de pouvoir établir le temps de transmission, une valeur est assignée pour représenter le *TARGET TOKEN ROTATION TIME* (**TTRT**). En effet le TTRT est la valeur théorique maximale que le jeton peut prendre pour parcourir l'anneau et revenir à la station de départ. D'autres paramètres sont définis en vue de calculer cette valeur. Tout comme nous l'avons déjà mentionné, à chaque station correspond un temps fixe pour la transmission des trames synchrones. Ce temps fixe coïncide avec l'allocation synchrone pour la station i ($\mathbf{SA_i}$). De plus, lors de la transmission, il y a un temps de propagation **P** pour un circuit de l'anneau**.** Il s'agit de permettre aussi la transmission d'au moins une trame asynchrone lors de la rotation complète du jeton. C'est le temps nécessaire pour pouvoir transmettre une longueur de trame maximale de 4500 octets **Tmax.** Le jeton doit aussi être transmis afin de passer le contrôle de l'anneau à une autre station pour fins de transmission. Il nous faut alors prendre en considération le temps de transmission **TT** du jeton autour de l'anneau.

Le TTRT peut donc être calculé comme suit :

$$\mathbf{P + Tmax + TT + \sum_i SA_i \leq TTRT}$$

Toutes les stations ont le même **TTRT,** mais des valeurs $\mathbf{SA_i}$ diffèrent. Plusieurs variables sont indispensables pour procéder à l'opération de l'algorithme d'allocation de capacité. Ces variables sont les suivantes :

- Temporisateur de rotation du jeton (*Token Rotation Timer*) **TRT**
- Temps de détention du jeton (*Token Holding Time*) **THT**
- Compteur de retard (*Late Counter*) **LC**

Le temps réel à compter du moment de la réception d'un jeton jusqu'à celui de sa réception suivante est calculé par chaque station. Le TRT se compose du TTRT moins la mesure du temps calculé par chacune des stations.

Le THT est le temps qui reste à une station donnée pour transmettre des trames asynchrones. Ceci est en fait, à l'origine, une copie du TRT. La raison pour laquelle le THT est nécessaire vient de ce que, après réception du jeton, le THT prend la valeur de TRT, et le TRT est remis à la valeur du TTRT pour recommencer immédiatement le décompte.

Le LC est un indicateur incrémenté à chaque fois que le jeton accuse du retard.

Nous allons examiner des situations diverses qui nous permettront de mieux saisir le mécanisme de l'allocation de la capacité.

Lorsque le jeton arrive tôt à une station, le THT est mis à la valeur du TRT et le TRT est mis à la valeur du TTRT. Le LC est mis à 0 et pour le TRT le décompte commence.

Toujours dans la même station qui a le jeton :

- la station peut transmettre des trames synchrones pendant le temps $\mathbf{SA_i}$ indépendamment de la valeur du compteur THT.
- Le THT est alors activé et le décompte commence lorsque les trames asynchrones entament leur transmission (soit après la transmission des trames synchrones),
- la station transmet des trames asynchrones aussi longtemps que THT > 0.

Nous remarquons que le temps de transmission des trames synchrones est indépendant de la valeur du THT.

Si, dans une station, le TRT atteint 0 avant de recevoir le jeton, alors le TRT est remis à la valeur du TTRT et le LC est mis à 1. Cela mentionne que le jeton accuse un retard.

Dans ce cas, lorsque le jeton est reçu :

- Le LC est mis à 0,
- Le TRT continue son décompte sans réinitialisation,
- s'il y a des trames synchrones, elles sont transmises jusqu'à $\mathbf{SA_i}$ sans se soucier du retard accusé,
- le THT n'est pas pris en considération,
- aucune transmission asynchrone n'est autorisée.

L'exemple de la figure 6.20 illustre le comportement d'une station qui reçoit le jeton en retard. Le TTRT est égal à 120 ms. La capacité synchrone de la station ou son SA est de 20 ms.

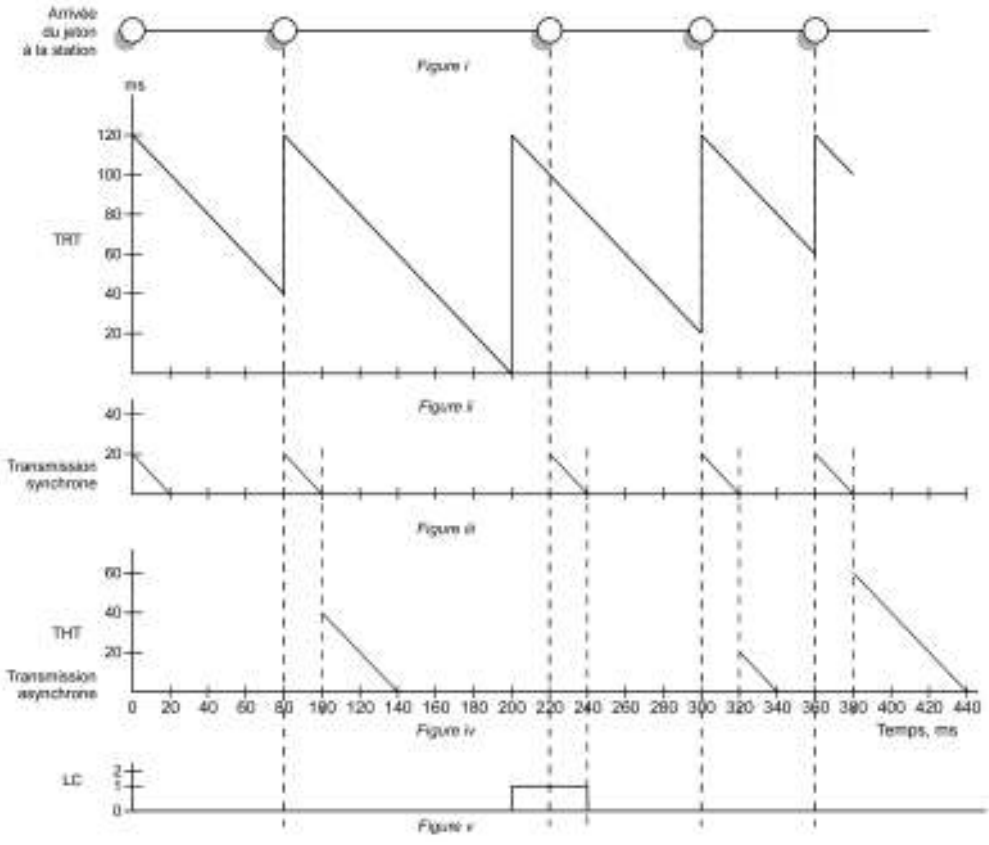


Figure 6.20 Exemple de représentation d'une allocation de capacité

En premier lieu, le jeton est arrivé tôt. La station n'a pas de trames asynchrones à transmettre. Le TRT prend la valeur de 120 ms. Le décompte du TRT commence. La station transmet des trames synchrones pendant 20 ms après quoi le jeton est transmis.

Le jeton revient à la même station 80 ms plus tard (*figure i* de la figure 6.20). Ceci implique que le TRT a fait un décompte jusqu'à 40 ms (*Figure ii*). La station met le THT à 40 ms, c'est-à-dire à la valeur du TRT, puis remet le TRT=TTRT=120ms. La station transmet ses trames synchrones pendant 20 ms (*Figure iii*). Par la suite le THT entame le décompte, et la station se met à transmettre des trames asynchrones pendant 40 ms (*Figure iv*), c'est-à-dire le

temps qui reste dans le THT. Le jeton est alors transmis. Signalons que le TRT fait son décompte depuis le temps où il a été initialisé à 120ms.

Le TRT de la station dans notre exemple arrive à 0 alors que le jeton n'est pas encore de retour à la station (*figure ii*). La station incrémente son compteur LC (LC=1) (*figure v*) et remet son TRT à 120 ms. 20 ms plus tard le jeton arrive à la station. Puisque LC=1, la station n'a pas le droit de transmettre des trames asynchrones. Par contre elle peut transmettre des trames synchrones pendant 20 ms (SA). Elle met par la suite LC=0 et sans toucher au TRT qui continu à décrémenter, elle transmet le jeton.

La suite de la figure 6.20 illustre un comportement normal de la station, vu que le jeton n'est pas en retard dans les prochains tours.

6.4.4 Surveillance de l'anneau

Trois processus sont impliqués dans la détection d'erreurs et de correction :

- Le Processus de réclamation du jeton
- Le Processus d'initialisation et
- Le Processus de vérification ou processus de balisage (*Beacon process*)

6.4.4.1 Réclamation du jeton

Lorsqu'une station s'aperçoit de la perte du jeton ou lorsque LC=2 (délai trop long sans réception de jeton), elle transmet des trames *claim*. Ce *claim* a l'adresse de la station qui en fait la demande ainsi qu'une valeur du TTRT proposée. Cette trame circule dans l'anneau allant d'une station à l'autre. Si, pendant son cheminement une autre station qui la reçoit décide qu'elle a une meilleure proposition de TTRT que celle qui a fait la demande, elle remplace l'adresse du *claim* par la sienne et met sa nouvelle valeur de TTRT. Les claims circulent dans l'anneau. Lorsqu'une station détecte ses claims, elle initialise le jeton.

6.4.4.2 Initialisation

Lorsque les stations du réseau voient passer des trames *claim*, elles se préparent au processus de réinitialisation. La station qui a reçu le droit de créer le jeton transmet un *nonrestricted token*. En voyant cela, chacune des stations passe de l'état initialisation à l'état opérationnel et met son TRT à la valeur acceptée du TTRT.

6.4.4.3 Vérification ou processus de balisage (Beacon Process)

Lorsqu'une station ne reçoit pas ses propres trames ou présume qu'il y a un mal fonctionnement de l'anneau, elle transmet des trames *beacon*. Toute station qui voit des trames *beacon* doit les laisser passer. Dans certaines circonstances les trames *beacon* sont transmises avant la transmission des trames *claim*.

6.4.5 Couche physique du FDDI

La couche Physique s'occupe entre autres de l'encodage des données. Pour le FDDI, le taux de transmission des données par anneau est de 100 Mbps. Afin d'obtenir une meilleure efficacité, le code 4B/5B est appliqué. Ce code prend 4 bits de données et l'encode n'en faisant qu'un symbole de 5 cellules (bits). Chacune de ces cellules représente un élément de signal unique (1 pour la présence de lumière, 0 pour l'absence de lumière). Chaque cellule de séquence 4B/5B est traitée en valeur binaire et est encodée conformément au codage NRZI (*nonreturn to zero inverted*).

Puisque 4 bits sont encodés, seuls 16 des 32 codes possibles sont requis pour l'encodage des données. Les données encodées ont ainsi une transition présente au moins deux fois pour chaque symbole. On ne peut avoir plus de trois zéros consécutifs. La figure 6.21 (qui a déjà été présenté dans le chapitre 4) adresse la table des codes correspondants.

CODE Group	Symbol	Assignment
Line state symbols		
00000	Q	Quiet
11111	I	Idle
00100	H	Halt
Starting delimiter		
11000	J	ND
10001	K	ND
Data Symbols		
11110	0	0000
01001	1	0001
10100	2	0010
10101	3	0011
01010	4	0100
01011	5	0101
01110	6	0110
01111	7	0111
10010	8	1000
10011	9	1001
10110	A	1010
10111	B	1011

CODE Group	Symbol	Assignment
11010	C	1100
11011	D	1101
11100	E	1110
11101	F	1111
Ending delimiter		
01101	T	
Control Indicators		
00111	R	logical 0 (reset)
11001	S	logical 1 (set)
Invalid Code Assignments		
00001	V or H	Violation or Halt
00010	V or H	Violation or Halt
00011		
00101		
01110	Violations	
01000		
01100		
10000		

Figure 6.21 Table du code 4B/5B

Soulignons qu'au lieu de la fibre, il existe une version du protocole sur fil de cuivre. Ce protocole est désigné par le sigle CDDI pour *Copper Distributed Bus Data Interface*. Il fonctionne de la même façon que le FDDI. La différence se trouve dans l'immunité électromagnétique et dans les distances de la station au concentrateur.

Le CDDI est pratique parce qu'un câble de paires torsadées de catégorie 5 peut être utilisé et parce qu'il est plus économique. Par contre, le CDDI est susceptible aux interférences électromagnétiques. De plus alors qu'il fonctionne aussi à 100 Mbps le CDDI n'accorde qu'un intervalle de 100 mètres entre le terminal et le concentrateur.

Le CCDI a été normalisé par le ANSI et est aussi connu sous le nom de TP-PMD (*Twisted-pair Physical Medium-Dependent*).

6.4.6 Reconfiguration de l'anneau après panne

Le FDDI requiert deux anneaux de fibre. Par contre un seul des anneaux est utilisé pour transporter l'information. L'autre anneau passif est mis en état de veille. L'anneau en opération est contrôlé par l'entité MAC. Deux types de

pannes peuvent survenir : la panne causée par le bris du lien, et celle causée par le bris d'une station.

Lorsque le lien tombe en panne, le tout premier processus amorcé est celui de la vérification. Les trames balisage (*beacon*) évaluent la panne et l'anneau est reconfiguré en isolant le segment de fibre en panne. Deux stations qui se trouvent de part et d'autre du lien défectueux permettent de clore la boucle avec l'anneau secondaire (Cf. figure 6.22).

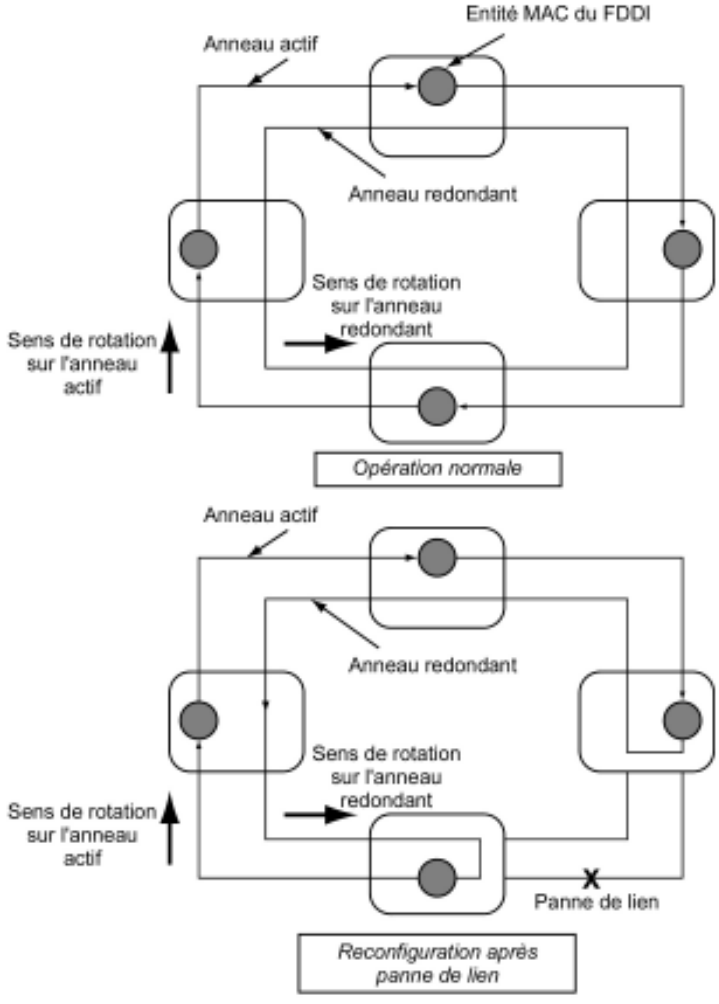


Figure 6.22 Reconfiguration après panne du lien

L'autre possibilité de panne survient lorsqu'une station ne fonctionne plus et ne relaie plus l'information sur l'anneau actif. Tel qu'illustré à la figure 6.23, les

stations adjacentes permettent alors de clore les boucles avec l'anneau passif et de former un nouvel anneau.

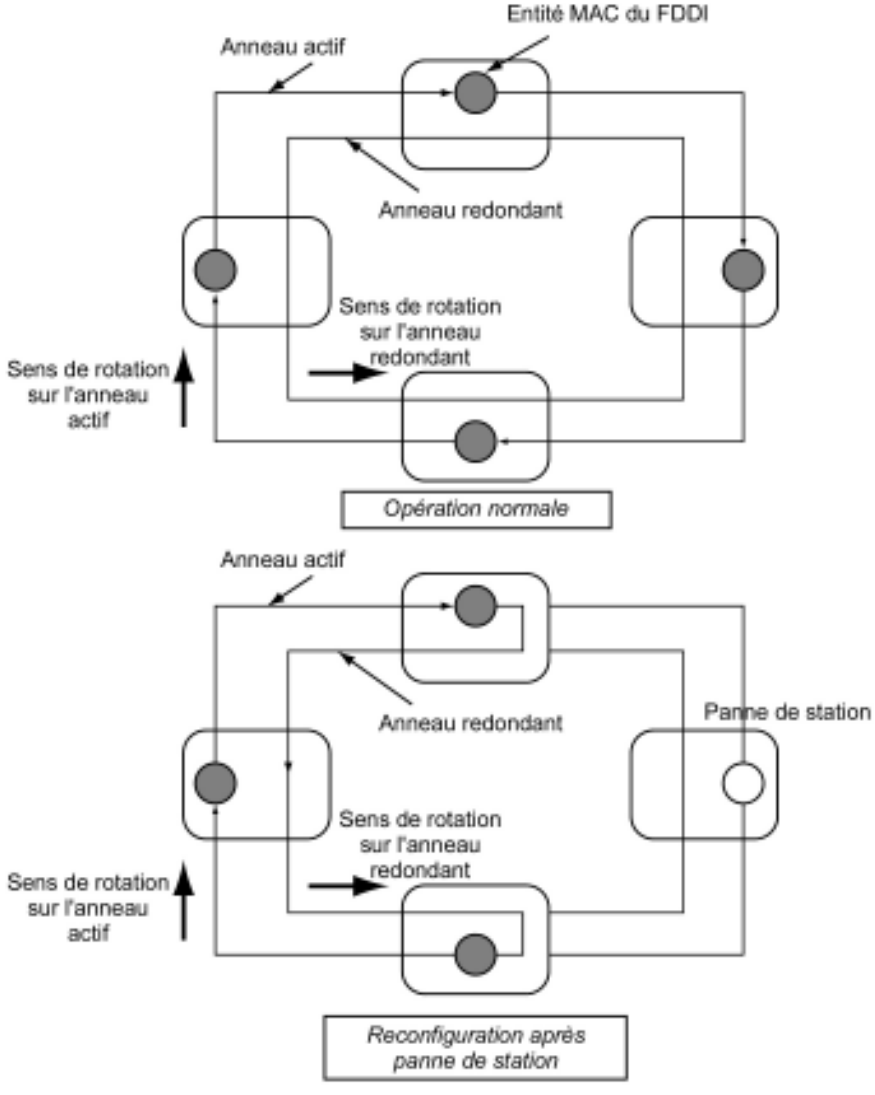


Figure 6.23 Reconfiguration après panne de la station

6.4.7 Types de stations

Il existe différents types de stations FDDI. On peut en identifier quatre illustrées à la figure 6.24. La première est la *Dual Attachment Station* (DAS) reliée aux deux

anneaux. C'est à cette station que l'on fera appel pour procéder au dépannage de l'anneau dans le cas des deux types de pannes.

La deuxième catégorie de station est la *Dual Attachment Concentrator* (DAC) (Cf. figure 6.25) qui, outre le fait d'assumer les fonctionnalités du DAS, permet de connecter des stations secondaires.

La troisième catégorie est la *Single Attachment Station* (SAS), rattachée au DAC. Cependant, elle ne fait appel qu'à un seul anneau. La quatrième catégorie est la *Single Attachment Concentrator* (SAC) qui permet le branchement de plusieurs SAS tout en n'ayant recours qu'à un anneau.

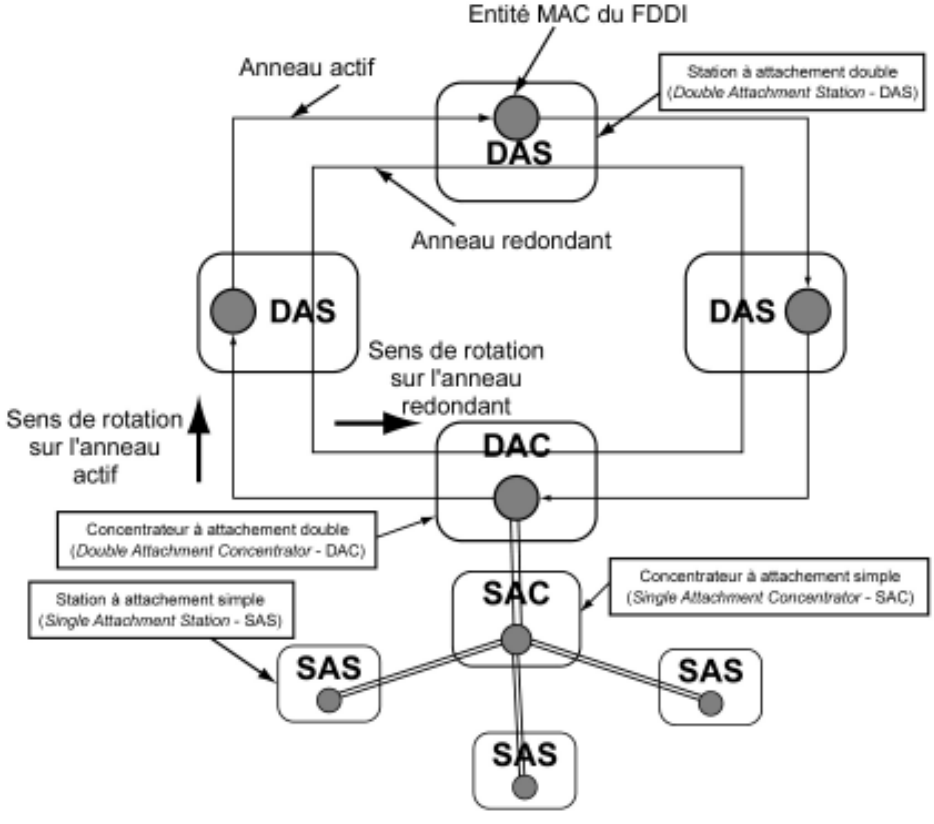


Figure 6.24 Raccordement au FDDI

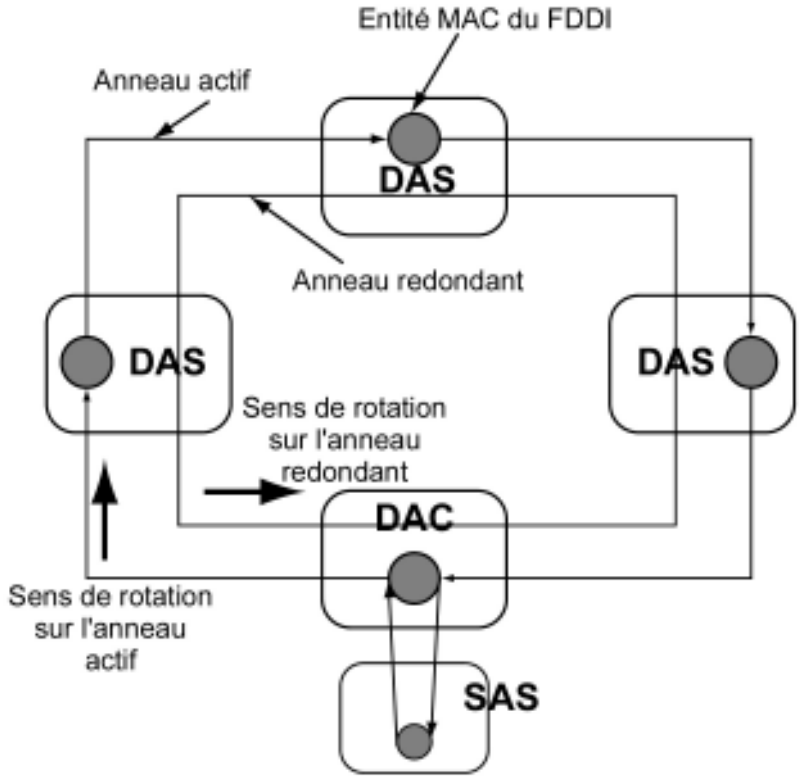


Figure 6.25 Description de l'accord du DAS avec le SAS

Chacun de ces types de station a des ports qui leur permettent de se connecter et qui sont les suivants :

- **Port A** : *Il* se c*onnecte à l'anneau primaire entrant et à l'anneau secondaire sortant du double anneau.*
- **Port B** : *Il* se c*onnecte à l'anneau primaire sortant et à l'anneau secondaire entrant du double anneau.*
- **Port M** : *Il connecte un concentrateur au SAS, au DAS ou à tout autre concentrateur.*
- **Port S** : *Il* c*onnecte un SAS ou SAC à un concentrateur.*

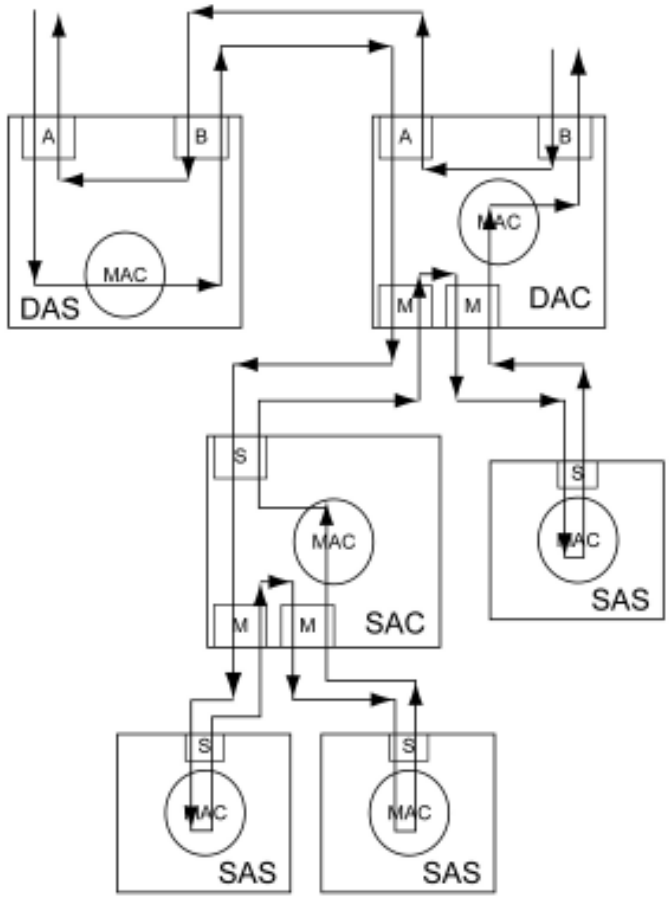


Figure 6.26 Utilisation des ports

La figure 6.26 illustre l'interconnexion des différents types de ports. Notons que l'anneau actif est celui qui passe par le MAC. Lorsqu'il y a une panne de lien de l'anneau actif ou d'une station, le port du DAS ou DAC qui la détecte et qui y est directement rattaché, crée un court circuit avec l'anneau passif. Ainsi le problème est isolé et un nouvel anneau est créé. Ce nouvel anneau utilise aussi bien l'anneau actif que passif.

La circulation de l'information et du jeton se fait clairement à travers chaque station connectée à l'anneau. Ce dernier doit toujours exister pour que le FDDI fonctionne. Si un SAC ou un de ses liens venait à tomber en panne, c'est le SAC et tous les SAS qui y sont rattachés qui seront isolés et déconnectés de l'anneau actif.

6.5 Exercices

BUS À JETON

6.1

L'ajout d'une station dans le Bus à jeton est un processus par lequel :

1- La station qui désire être ajoutée fait la demande à un moment approprié.
2- Le successeur immédiat accorde la permission.
3- Le détenteur du jeton décide de faire la demande lorsque le temps assigné à son jeton vient à expiration.
4- Le détenteur du jeton fait la demande lorsqu'il en a encore le temps.
5- Le détenteur du jeton attend qu'un serveur central lui en accorde le droit.

6.2

Dans le Bus à jeton, lorsqu'une station s'apprête à être ajoutée :

1. Elle demande leur adresse aux stations précédente et suivante.
2. La station précédente lui remet toutes les adresses.
3. La station suivante lui remet les adresses.
4. Elle reçoit les adresses avec le jeton.
5. Elle demande les adresses seulement à la station précédente.

6.3

Dans le Bus à jeton, le jeton circule toujours :

1. D'une station dont l'adresse est la plus élevée vers une station dont l'adresse est inférieure à la sienne.
2. Dans l'ordre décroissant des adresses pour toutes les stations connectées au bus.
3. Dans l'ordre décroissant des adresses en allant de la plus basse à la plus élevée pour toutes les stations connectées au bus.
4. Dans l'ordre décroissant des adresses en allant de la plus basse à la plus élevée pour toutes les stations connectées à l'anneau.
5. Dans l'ordre décroissant des adresses en allant de la plus faible à la plus élevée pour toutes les stations connectées actives au sein même du réseau.

6.4

Dans le Bus à jeton, la station qui est sur le bus se retire :

1- En transmettant un *set-successor* à la station précédente en tout temps et lorsqu'elle le désire.
2- En transmettant un *set-successor* à la station précédente même si elle-même ne se trouve pas sur l'anneau logique.
3- En transmettant un *set-successor* à la station précédente lorsqu'elle reçoit le jeton.
4- En se déconnectant.

6.5

Dans le Bus à jeton, la commande *who-follows* est utilisée par :

1. la station qui se retire.
2. la station qui connaît la station suivante de celle qui se retire sans préavis.
3. la station qui était la station suivante de celle qui se retire sans préavis.
4. la station qui était la station précédente de celle qui se retire sans préavis.

6.6

La fenêtre utilisée dans le Bus à jeton correspond :

1. Aux deux premiers bits de l'adresse MAC.
2. Au temps de propagation aller retour multiplié par deux bits de l'adresse MAC.
3. Au temps de propagation aller retour multiplié par la distance du bus.
4. Au temps de propagation aller retour multiplié par la distance entre les deux dernières stations.

6.7

Sur un bus fonctionnant conforme au protocole IEEE 802.4, quatre stations deviennent actives en même temps (par exemple, suite au démarrage des équipements d'une chaîne de montage). Que se passe-t-il? Citez l'ordre dans lequel les stations s'insèrent dans l'anneau logique (On suppose que les stations portent les adresses 9, 4, 2 et 1).

ANNEAU À JETON

6.8

L'anneau à jeton est basé sur :

1. La circulation logique du jeton dans tous les types de réseaux.
2. La circulation physique du jeton dans tous les types de réseaux.
3. La circulation physique du jeton dans un réseau à anneau.
4. La circulation logique du jeton dans un réseau à bus.

6.9

Le monitor qui se trouve dans l'anneau à jeton est une station qui :

1. Surveille le comportement de chaque station.
2. Distribue les statistiques d'utilisation du réseau.
3. Permet la gestion des réseaux.
4. Assure la circulation des trames autour de l'anneau à un seul tour complet.

6.10

Dans l'anneau à jeton une station qui se retire doit :

1. Envoyer un *beacon* à toutes les stations.
2. Prévenir les stations adjacentes.
3. Se retirer sans aviser aucune station.
4. Prévenir le monitor.

6.11

Lorsque le jeton est perdu dans un anneau à jeton :

1. La première station qui le réclame l'obtient automatiquement.
2. La station qui voit le *claim-token* avec son adresse, crée le jeton.
3. La station avec l'adresse la plus élevée, obtient le jeton.
4. Le monitor décide de l'assignation du jeton.
5. Le monitor crée le jeton.

FDDI

6.12

a) Dans un anneau FDDI, à quel moment une station pourra transmettre le jeton?
b) Peut-il y avoir plusieurs trames FDDI en circulation dans l'anneau. Expliquez le processus d'élimination des trames
c) Pourquoi un anneau optique FDDI utilise le protocole 4B/5B-NRZI ? Expliquez.

6.13

Un réseau FDDI est composé de 4 stations DAS et de 2 stations DAC. Un regroupement comprend 3 SAS et 1 SAC avec 2 SAS. L'autre regroupement se compose de 2 SAS et d'1 SAC avec 3 SAS. Dessinez le réseau en question en prenant en considération les connexions internes de chacun des éléments.

6.14

Dans le contexte d'un réseau FDDI :

a) Expliquez ce que les trames synchrones et asynchrones représentent et à quelles fins elles sont utilisées.

b) Expliquez quelles sont les contraintes de chacun de ces services (synchrone et asynchrone).

c) Expliquez ce qui arrive lorsqu'une station retient le jeton pour transmettre toutes ses trames et que le compteur TRT arrive à zéro pendant ce temps.

6.15

Un réseau FDDI a le TTRT égal à 150 ms. La station Gamma s'est vu assigner une allocation synchrone (SA) de 25 ms. Cette station est composée de trames synchrones qu'il faut transmettre lors de chacun des cycles. Les trames synchrones ont toutes une longueur de 1500 octets. Les trames asynchrones que cette station pourrait transmettre ont une longueur de 4500 octets. Le cycle de vie du jeton est en moyenne de 100 ms.

a) Établissez le nombre de trames synchrones transmises à chaque cycle.

b) Si le jeton est reçu sans retard et que la station peut transmettre autant de trames asynchrones que possible, dessinez les diagrammes de temps pour la transmission des trames synchrones et asynchrones pendant 4 cycles du jeton.

c) Donnez la durée de temps du moment ou le jeton est transmis par la station au moment ou il retourne à cette même station. Faites ce calcul pour chaque itération du jeton.

7 L'interfonctionnement

7.1 Introduction

Les réseaux locaux sont très utiles lorsqu'il s'agit de desservir un ensemble de postes de travail surtout dans le cas où les échanges d'information se limitent au réseau. Ce processus est simple lorsqu'il s'agit d'un nombre restreint de postes de travail se trouvant à de très courtes distances les uns par rapport aux autres. Toutefois, lorsque le réseau local doit être étendu au-delà des capacités électriques du segment, que le nombre de stations devient important, ou encore qu'il soit impérieux de s'interconnecter à un autre réseau, on a alors recours à des équipements de réseaux tels les concentrateurs (*repeater)*, les ponts (*bridge)* ou les routeurs (*router)*. Chacun de ces éléments de réseaux s'avère être d'une utilité qui lui est propre. Nous allons essayer d'en donner une description fidèle et de comprendre les contraintes qui y sont reliées.

7.2 Les concentrateurs

Dans le cas où il faut agrandir la distance d'un réseau local du type Ethernet, l'usage d'un concentrateur est nécessaire. Le concentrateur est un équipement d'interfonctionnement agissant au niveau de la couche 1 du modèle de référence OSI tel qu'illustré à la figure 7.1. Sa fonction vise à prolonger le segment existant. Les postes de travail ignorent son existence sur le réseau.

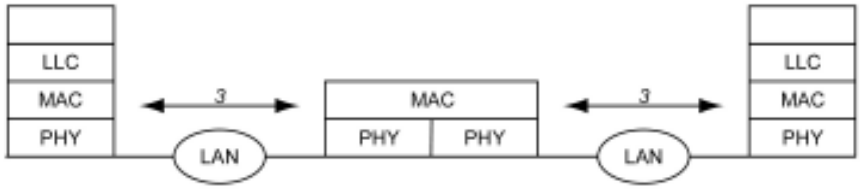


Figure 7.1 Représentation des fonctions du concentrateur

Le concentrateur régénère le signal qu'il reçoit d'un segment câblé et le transmet à l'autre segment. Il génère aussi un nouveau préambule pour chacune des trames reçues avec indication du début de la trame SFD (*Start Delimiter*/Délimiteur de début de la trame). Ses fonctions sont très simples et sans aucune programmation autre que la régénération du signal émis.

Lorsqu'il y a collision sur l'un des segments, le concentrateur va tout simplement propager la collision à l'autre segment. À la figure 7.2 illustrant des segments interconnectés par des concentrateurs, toutes les transmissions de la station 10, par exemple, seront propagées à toutes les autres stations à travers les deux concentrateurs.

Étant donné que la collision est également propagée, le domaine de collision couvre le réseau au complet. Cela signifie que s'il y a collision, toutes les stations sont concernées et immobilisées jusqu'à ce que la transmission revienne à la normale.

Malgré la régénération du signal, le concentrateur ne peut pas résoudre les problèmes liés au temps de propagation (effet sur la longueur de la trame). C'est pourquoi un réseau local a une étendue maximale même avec les concentrateurs. De plus, la largeur de bande est partagée par tous les terminaux qui communiquent.

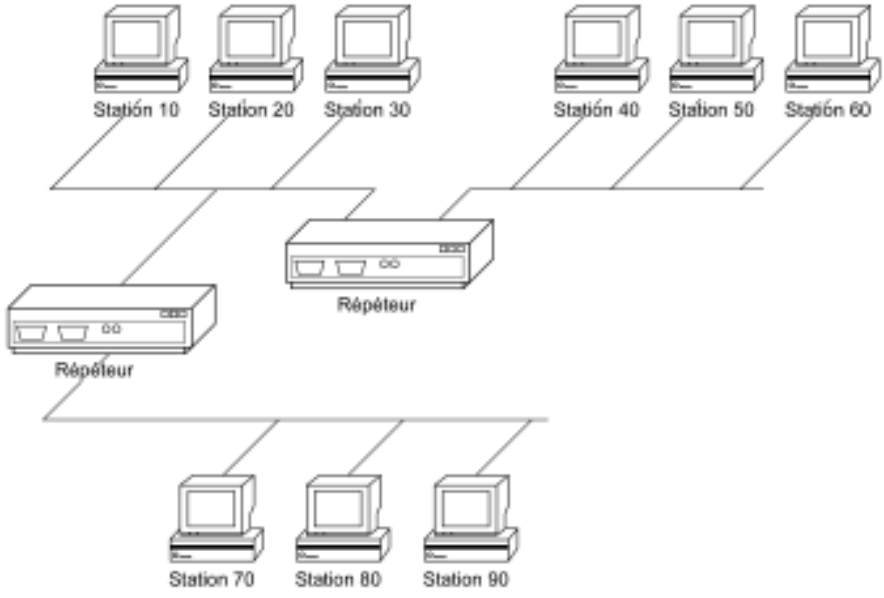


Figure 7.2 Représentation de concentrateurs sur réseau local

Une transmission de diffusion (*broadcast*) ou de multidistribution (*multicast*), comme toute autre transmission, est transmise sur tout le réseau avec un concentrateur. Le domaine de diffusion (*broadcast*) couvre donc le réseau au complet. Il ne reste donc qu'un seul domaine de diffusion d'un réseau local équipé uniquement de concentrateurs.

7.3 Les ponts (*Bridges*)

Contrairement au concentrateur, le pont fait appel à un processus de filtrage pour savoir s'il doit faire passer la trame à l'autre segment. Le pont est un simple équipement d'interfonctionnement utilisé entre les segments ou les réseaux locaux dont le protocole MAC est identique.

Un réseau local du type Ethernet exploite un pont qui comprend idéalement les fonctionnalités de services de transfert d'une fonction de pontage transparent (*Transparent Bridging*) sur lesquelles nous nous pencherons ultérieurement dans ce chapitre.

Idéalement, le pont des réseaux locaux avec anneau à jeton utilise la technique du pontage d'acheminement par la source (S*ource Routing Bridging*) dont nous traiterons brièvement plus loin.

Tout comme le concentrateur, le pont offre une prolongation du réseau local de type Ethernet. Il couvre seulement les couches, physique et liaison du modèle OSI (figure 7.3). La couche réseau des deux systèmes interconnectés doit être identique.

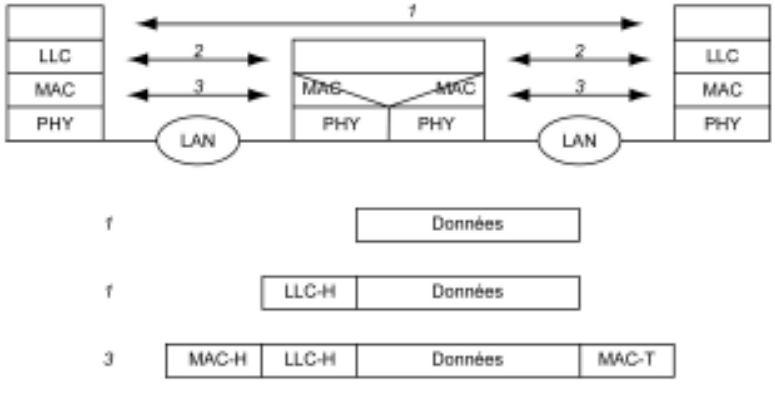


Figure 7.3 Représentation type de réseaux locaux interconnectés avec des ponts

Le pont ne retransmet pas les collisions détectées sur un segment. Il isole les segments en créant des domaines de collision indépendants et séparés les uns des autres. Malgré ces collisions, les autres segments peuvent continuer à opérer indépendamment en amont du pont.

Par contre, lorsqu'il y a une transmission de diffusion ou de multidistribution, le pont doit propager les trames qu'il reçoit sur tous ses ports, c'est-à-dire vers tous les segments du réseau tout comme dans le cas du concentrateur. Ici aussi un seul domaine de diffusion couvre tout le réseau si des ponts ou concentrateurs sont uniquement utilisés. Toutes les stations du réseau composé de ponts ont la même adresse réseau parce qu'elles relèvent du même réseau.

Un pont contrôle le flux du trafic, mais ne change pas le contenu de la trame. Ce contrôle de flux se fait en limitant le passage des trames, qui ne sont pas destinées à d'autres ports de sortie, et en bloquant les collisions et les trames erronées.

Une configuration particulière de pont nous est donnée par le commutateur. Ce dernier est un pont doté de plusieurs ports, chacun d'entre eux étant assigné à une station. De la sorte, la fonction du commutateur vise à filtrer les trames et à les acheminer vers leur port de destination conformément à l'adresse de destination de la trame.

Les ponts peuvent rendre le réseau local fiable lorsque la distribution des données se fait sans congestion. Il a été suggéré[1] que, dans un réseau local traditionnel tel qu'illustré à la figure 7.4, une répartition de trafic, dont près de 80 % est destiné à l'intérieur du segment et près de 20 % au-delà du pont, permet au réseau de rester stable et fiable. Si le trafic externe excède les 20 %, le pont commence à se comporter comme un concentrateur.

[1] Kennedy Clarck, Cisco LAN Switching, Cisco Press 1999

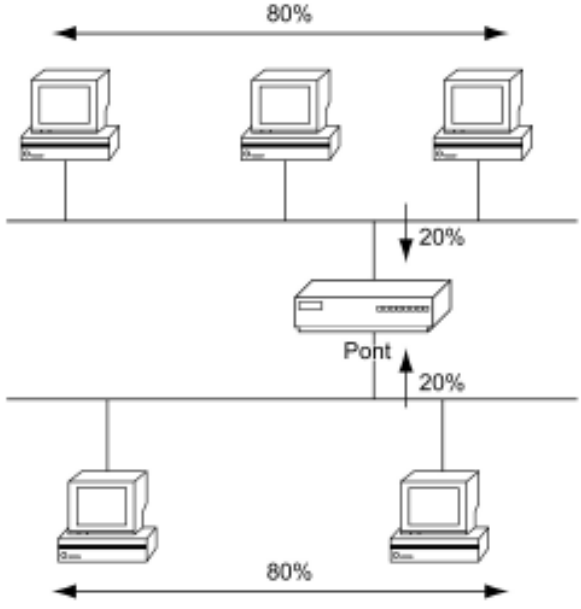


Figure 7.4 Représentation de la répartition du trafic

Prenons l'exemple de la figure 7.5. Les stations 10, 20 et 30 sont dans le même segment et tout échange entre ces stations se limite à ce segment. Les ponts s'assurent de ne pas faire passer les trames, sachant que les stations de destination se trouvent du même côté que celui de l'origine.

Par contre, lorsque la station 20, par exemple, transmet une trame destinée à la station 50, le pont A laisse passer la trame, mais le pont B la bloque sachant que la destination ne se trouve pas au-delà.

Les ponts connaissent les adresses de destination qu'ils mémorisent dans des tables. Celles-ci indiquent les adresses MAC des stations connues et le port par lequel leurs trames arrivent.

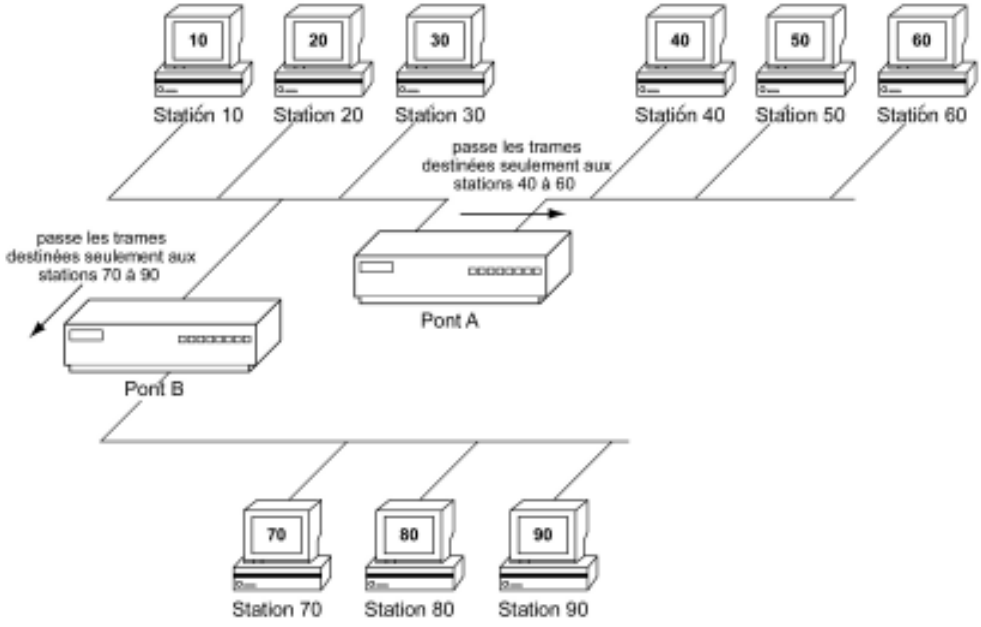


Figure 7.5 Représentation type de la fonction d'un pont

7.4 Tables d'acheminement

Afin d'acheminer les trames d'un segment à l'autre, les ponts ont recours aux tables d'acheminement.

La méthode la plus simple et la plus banale de la mise à jour de ces tables est qualifiée de méthode fixe. Dans cette méthode, les entrées se font manuellement.

Considérons un réseau local composé de plusieurs segments comme celui de la figure 7.6. Chaque pont comprendrait une table d'acheminement qui lui permettrait de filtrer les trames parvenant de chacun des ports.

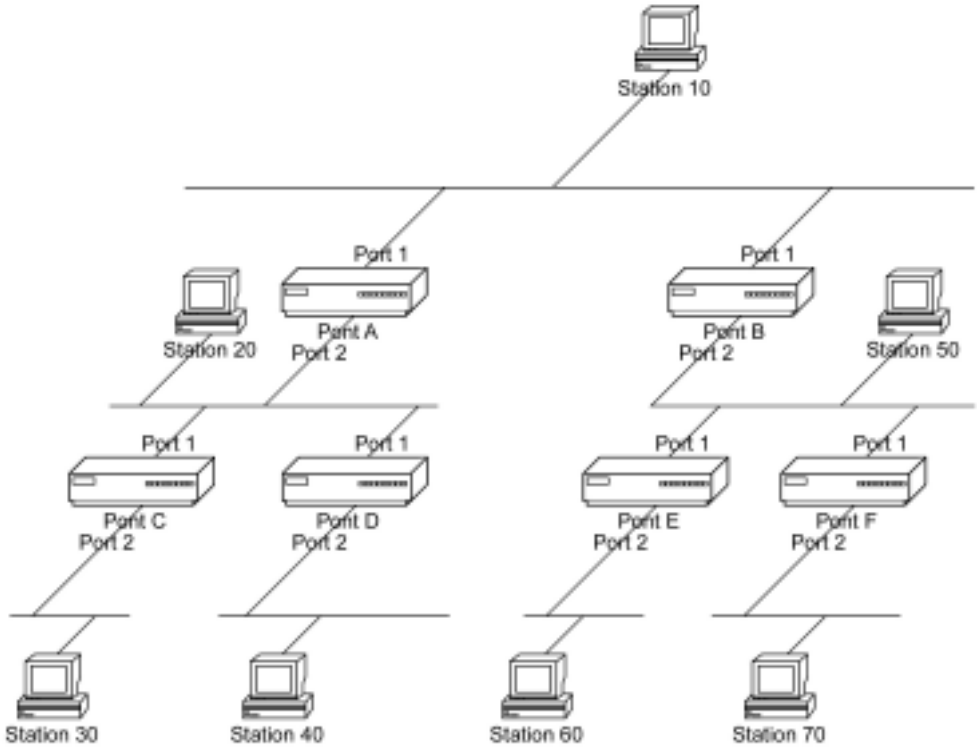


Figure 7.6 Représentation type de l'acheminement à travers des ponts

Il est indispensable de pouvoir visionner le réseau local de façon globale afin d'éviter de créer la possibilité de boucle. Une table d'acheminement globale établit le pont utilisé par des trames provenant de stations d'un segment et destinées à une autre station sur un autre segment. Dans le cas de notre exemple, la table globale serait celle qui est représentée à la figure 7.7.

Cette table assigne à chaque pont tant les trames qui doivent être acheminées que celles qui doivent être bloquées.

Station source \ Station destination	10	20	30	40	50	60	70
10	-	A	A	A	B	B	B
20	A	-	C	D	A	A	A
30	C	C	-	C	C	C	C
40	D	D	D	-	D	D	D
50	B	B	B	B	-	E	F
60	E	E	E	E	E	-	E
70	F	F	F	F	F	F	-

Figure 7.7 Représentation d'une table globale d'acheminement fixe

Les tables d'acheminement des ponts donnent le port d'origine de la trame, l'adresse de destination MAC ainsi que le port de sortie. Dans notre exemple, la table d'acheminement du pont A est illustrée à la figure 7.8.

Port d'entrée	Adresse MAC Destination	Port de sortie
1	10	-
1	20	2
1	30	2
1	40	2
1	50	-
1	60	-
1	70	-
2	10	1
2	20	-
2	30	-
2	40	-
2	50	1
2	60	1
2	70	1

Figure 7.8 Représentation type de la table d'acheminement pour le pont A

Par exemple, la trame qui arrive par le port 1 avec l'adresse de destination 30 sera acheminée vers le port de sortie 2.

Le problème que l'on peut rencontrer dans un réseau à arborescence, comme celui représenté à la figure 7.6, survient lorsqu'il y a des routes distinctes menant à la même destination et formant une boucle. Ce cas est illustré par la figure 7.9.

On peut encore avoir recours à l'acheminement fixe. Par contre, lorsqu'une panne survient dans l'un des ponts, s'il y a une route, les entrées doivent être changées manuellement aux autres ponts pour refléter le nouveau chemin qui sera emprunté.

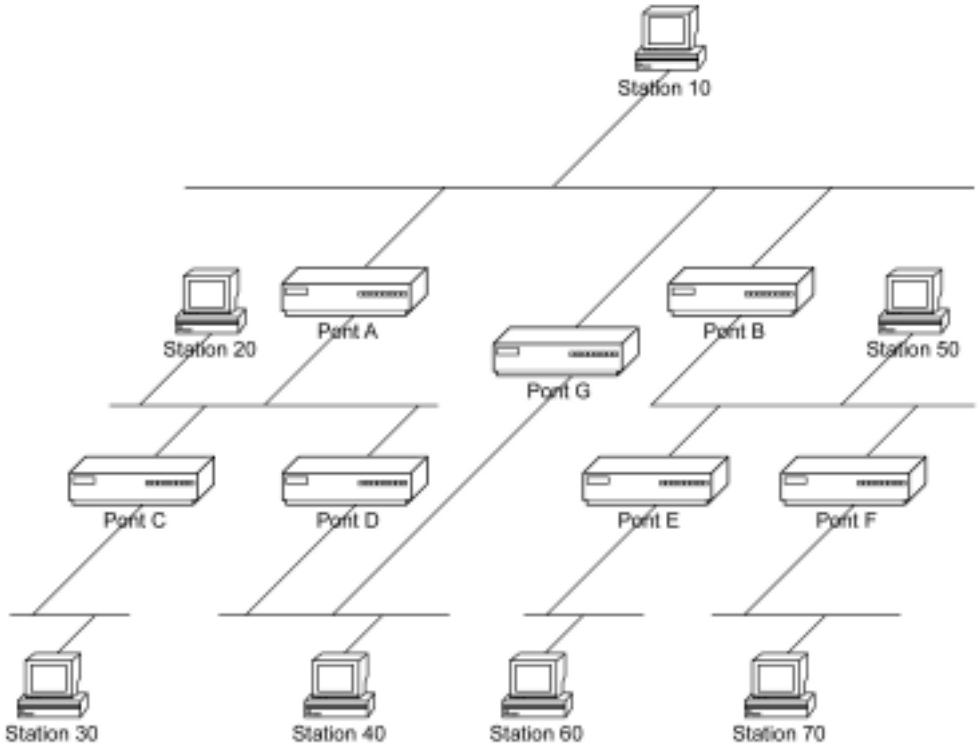


Figure 7.9 Représentation de l'acheminement à travers les ponts et les routes distinctes

De plus, si une nouvelle station est ajoutée à un des segments, il faut reprogrammer les tables de tous les ports. Comme on peut l'imaginer, ce processus risque d'échapper à tout contrôle au fur et à mesure que les stations changent de place, sont ajoutées ou retirées du réseau.

Il est donc indispensable de mettre sur pied un dispositif qui puisse procéder à la mise à jour automatique des tables d'acheminement.

7.5 Fonction de pontage transparent (*Transparent Bridging*)

La norme IEEE802.1d recommande une manière d'aborder le problème qui permettait aux ponts d'établir automatiquement leur table d'acheminement et de

la mettre à jour sans intervention du gestionnaire du réseau. Cette méthode est connue sous le nom de fonction de pontage transparent (*Transparent Bridging*). Cette méthode permet aux ponts de savoir en tout temps dans quelle situation se trouvent les stations par rapport à leurs ports.

Le but des ponts est, d'une part, de relayer et de filtrer les trames, et de l'autre, d'obtenir des informations qui lui seront utiles lors de la prise de décision relative au relais et au filtrage des trames.

Le pont doit recevoir la trame. S'il y trouve une erreur, il la met de côté. Si la trame lui est adressée, il la traite localement et ne procède pas au relais.

Le pont commence par analyser l'adresse de destination. Si celle-ci se trouve dans sa table d'acheminement et que le port qui mène à la station destination est autre que le port d'origine de la trame, le pont l'y relaie. Si, au contraire, le port menant à la destination est le même que le port d'origine de la trame, cette dernière n'est pas acheminée. Enfin, si l'adresse de destination ne figure pas dans sa table d'acheminement, le pont relaie la trame par tous les ports à l'exception du port d'origine.

Lors de la réception des trames de données, le pont assume une autre fonction : celle de l'apprentissage. Dans sa table d'acheminement, le pont copie l'adresse source de la trame des données ainsi que l'identification de son port d'entrée. De la sorte, le pont tire parti de la réception des trames pour mettre à jour ses tables d'acheminement même si l'entrée de l'adresse est déjà existante. Cette phase d'apprentissage permet de procéder à la mise à jour des tables d'acheminement automatiquement. De plus, si une station venait à changer de réseau local et que ses trames parvenaient au même pont par un autre port, l'inscription dans la table d'acheminement serait automatiquement mise à jour.

Si une inscription dans la table d'acheminement n'a pas été mise à jour après un certain délai, son entrée en sera effacée. Par la suite, le pont devra distribuer à travers tous les ports sauf celui qui la reçoit, toute trame reçue et envoyée à cette adresse.

Afin d'éviter les boucles, chaque pont applique l'algorithme *Spanning Tree*. Nous examinerons cet algorithme un peu plus loin dans le présent chapitre.

Les réseaux locaux sont raccordés au pont par des ports. Chacun d'entre eux comprend une entité MAC.

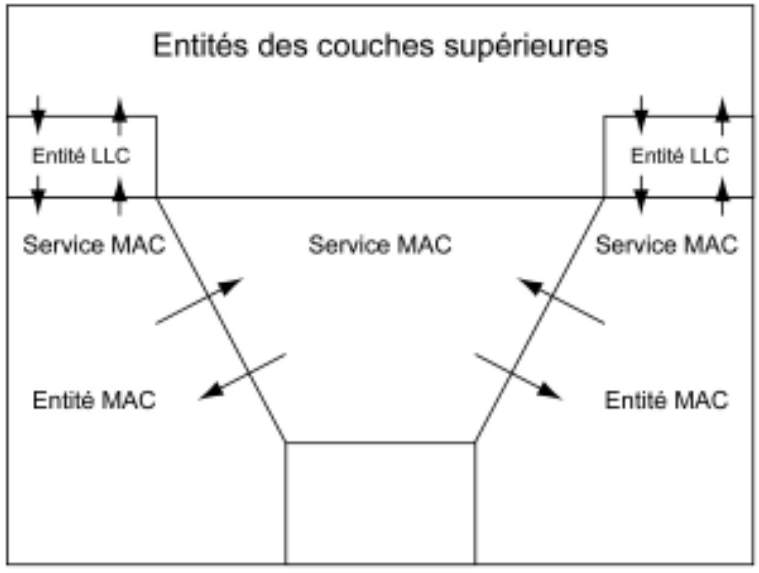


Figure 7.10 Exemple type d'une opération de base de fonction de pontage transparent

Comme l'indique la figure 7.10, le protocole MAC est traité par son entité MAC. Celle-ci se saisit de toutes les trames qui lui parviennent y compris celles qui ne lui sont pas adressées. Ces trames MAC peuvent être de trois types :

- Des trames adressées au pont
- Des trames de contrôle
- Des trames de données

La trame MAC, qui contient des données devant être relayées par le pont, est d'abord défaite de son en-tête. La trame LLC qui reste est encapsulée de nouveau afin d'être conforme au segment sur lequel elle sera transmise.

7.5.1 Acheminement des trames

La base de données de chaque pont devrait contenir les adresses de toutes les stations connues y compris celle du port du pont où l'on peut les rejoindre. Si la base de données n'a pas d'entrée pour une station, alors l'on fait appel à la fonction d'apprentissage d'adresse (*Address learning*). L'apprentissage de l'adresse se fait lors du passage de chaque trame par le pont. Cet apprentissage se fait à partir de l'adresse source. Le pont qui reçoit une trame va reconnaître la source comme provenant du port d'où la trame a été reçue. Ultérieurement, toutes les trames adressées à cette station seront acheminées vers ce port (*Forwarding)*. Si l'une de ces trames provient du même port, elle sera bloquée. L'adresse

d'acheminement (*Forwarding address)* est associée au port. La figure 7.11 illustre le processus.

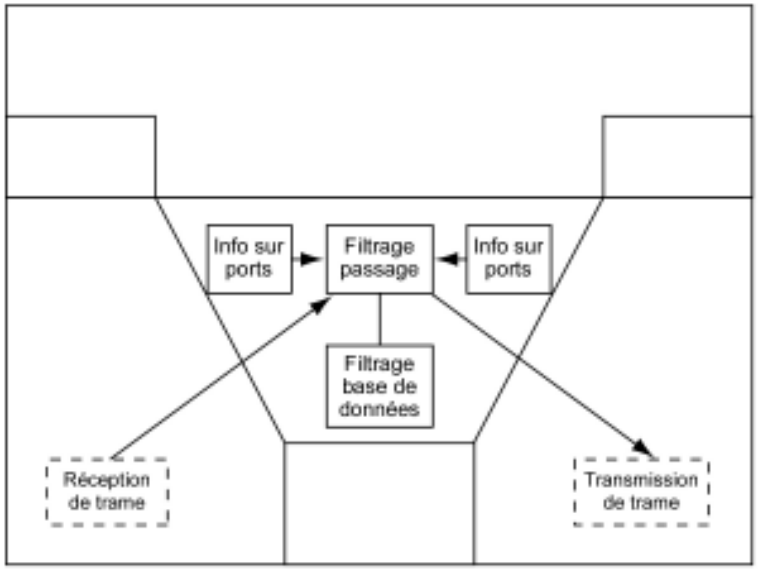


Figure 7.11 Représentation type de l'acheminement d'une trame

Le schéma de l'algorithme de fonctionnement du pont, lors de l'acheminement de la trame, est représenté à la figure 7.12.

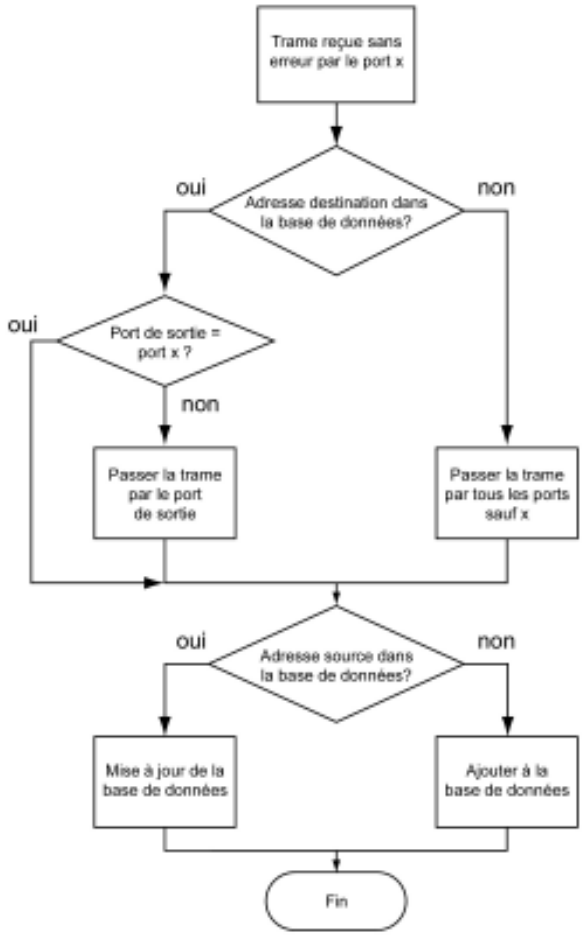


Figure 7.12 Représentations de la fonction apprentissage d'adresse

7.6 Technique de pontage d'acheminement par la source

Les anneaux à jeton peuvent être interconnectés par des ponts. La station qui se trouve sur un anneau peut ainsi communiquer avec la station se trouvant sur un autre anneau. Lorsque plusieurs anneaux sont interconnectés, il importe d'être informé au sujet du chemin que devra emprunter une trame pour parvenir à destination.

Lorsqu'IBM a introduit l'anneau à jeton, une technique de pontage par rapport aux services de transfert transparent par pont a été mise de l'avant. Il s'agit du pontage d'acheminement par la source (*Source Routing Bridging*).

Le principe de cette technique consiste à lier chaque trame au chemin qu'elle doit emprunter en identifiant le pont et l'anneau qu'elle doit traverser. Chaque anneau et chaque pont sont assignés par l'administrateur de réseau avec un numéro d'identification unique.

Avant de transmettre des trames destinées à une autre station, la station source doit connaître le chemin qu'emprunteront ses trames pour se rendre à leur point de destination. Si le chemin n'est pas connu, la station source transmet en premier lieu une trame test qui va permettre de savoir si la station de destination se trouve sur le même anneau. Si une réponse à cette trame test est reçue, c'est que la station recherchée se trouve effectivement sur le même anneau. Dans le cas contraire, l'absence de réponse est un signal qui indique que la station se trouve sur un autre anneau.

La source transmet la trame ARE (*ALL Routes Explorer*/ Exploration de toutes les routes). Cette trame traverse tous les anneaux et tous les ponts du réseau. Au passage, chacun des ponts y insère le numéro de l'anneau local ainsi que son propre numéro d'identification. L'en-tête de chaque trame comprend un champ pour ces identificateurs.

Le champ RIF (*Routing Information Field*/Information de chemin) est le champ de l'en-tête et comprend aussi un bit, celui du RII (*Routing Information Indicator*/Indicateur d'information de chemin). Ce bit est à 0 lorsque la source et la destination se trouvent sur le même anneau ; autrement, il est à 1. C'est donc dans le champ RIF que chaque pont traversé rajoute ses numéros d'identification.

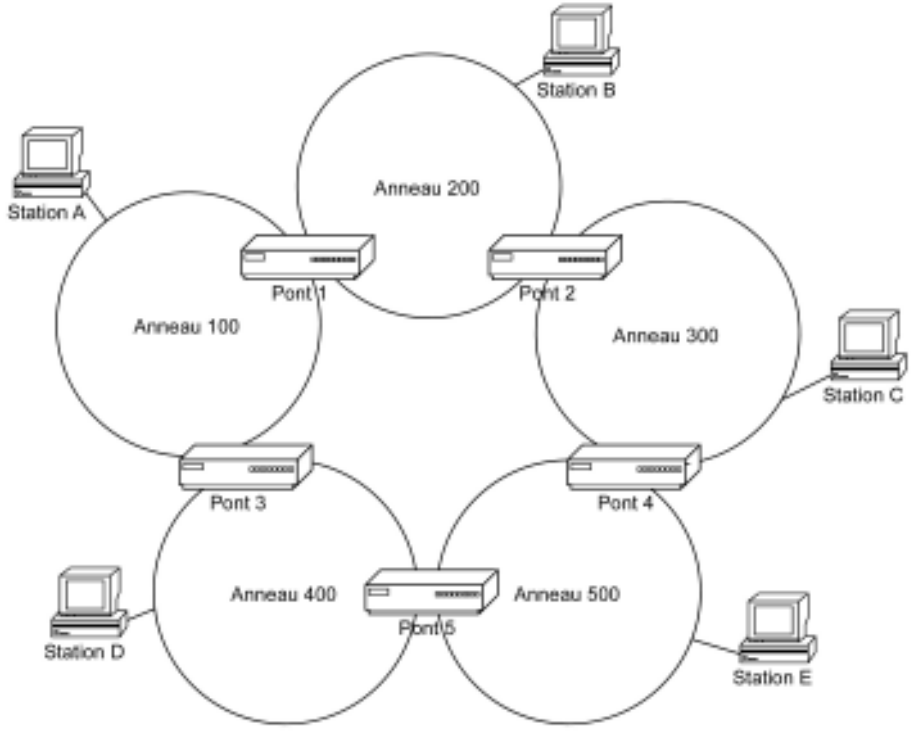


Figure 7.13 Représentation de l'interconnexion d'anneaux à jeton

Dans l'exemple de la figure 7.13, si la station A transmet une trame ARE pour trouver le meilleur chemin en vue d'atteindre la station E, cette dernière reçoit deux trames ARE avec leurs RIF différents, notamment :

Anneau 100 – Pont 1 – Anneau 200 – Pont 2 – Anneau 300 – Pont 4 – Anneau 500
Anneau 100 – Pont 3 – Anneau 400 – Pont 5 – Anneau 500

La station E renvoie à la source chacune des trames ARE reçues. La Station A doit donc décider du meilleur chemin en se fondant sur l'information reçue.

La décision peut être l'une des suivantes :

- Utiliser la première réponse reçue, c'est-à-dire le premier ARE qui revient à la source et qui comporte un cheminement dans son RIF.
- Utiliser le chemin avec le moins de sauts.
- Utiliser le chemin avec le plus grand MTU (*Maximum Transfer Unit*), c'est-à-dire qui permet de recevoir la trame la plus grande.

Normalement, la première décision est la méthode la plus couramment utilisée.

7.7 Algorithme *Spanning Tree*

L'apprentissage de l'adresse (*Address learning*) est efficace si le réseau est un arbre. Toutefois, s'il y a des routes alternatives, des boucles se forment. Dans ce cas, le processus d'apprentissage de l'adresse va se faire à l'infini, ce qui rend l'acheminement de trames instable.

Le danger de blocage peut naître de la présence de boucles lors de la diffusion de trames. Imaginons un réseau local ne comportant qu'une simple boucle (voir figure 7.14). Nous avons deux ponts connectés de chaque côté à un segment. Si la station A diffuse une trame, les deux ponts doivent relayer cette trame à l'autre segment. Il est évident que chacun des ponts attendra son tour de transmission. Supposons que le pont 1 relaie en premier cette trame au deuxième segment, le pont 2 va recevoir une deuxième copie de la trame. La première copie est déjà dans la file d'attente pour être aussi transmise au deuxième segment. Sur réception de la deuxième copie de la trame au port 2, le pont 2 remarque que l'adresse source dans la trame est celle de la station A. Le pont 2 enregistre alors la nouvelle position fictive de la station A et procède au relais de la trame sur le premier segment.

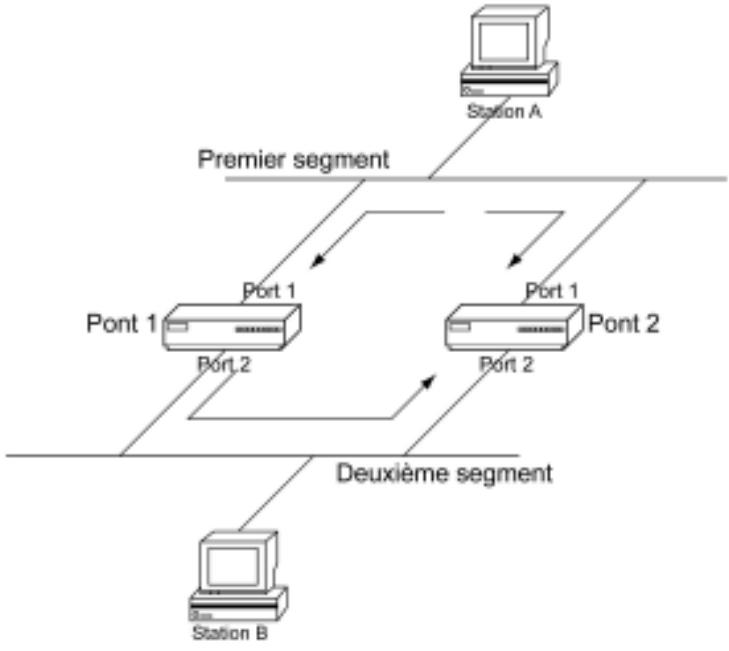


Figure 7.14 Représentation type d'une boucle dans un réseau à arborescence

Les trames vont alors circuler indéfiniment dans le temps à travers la boucle et les ponts vont continuellement changer la position des stations dans leur table d'acheminement.

Ce comportement se présente également lorsque des trames à diffusion individuelle (*unicast)* sont transmises.

Afin de pallier un tel problème, le protocole STP (*Spanning Tree Protocol*) est appliqué. Ce protocole élimine les boucles et permet au pont de procéder à l'apprentissage de l'adresse sans créer de conflits. Le protocole est basé sur les trois points suivants :

- À chaque pont est assigné un identifiant unique (adresse MAC du pont plus la priorité).
- Chaque port est identifié de façon unique dans son pont par un identifiant du port.
- Coûts associés au lien en terme d'entier.

L'identifiant du pont BID (*Bridge ID*) se compose de huit octets. Six d'entre eux comprennent l'adresse MAC du port et les deux autres sont réservés pour la priorité du pont. La valeur par défaut de cette priorité est de 32768.
Le coût associé au lien reflète la vitesse du lien. Auparavant, ce coût était le rapport entre 1000 Mbps et la vitesse du lien. De nos jours, les vitesses peuvent excéder 1000 Mbps ce qui donnerait un rapport fractionnel inacceptable pour la représentation d'un coût. Le IEEE a donc établi un tableau des coûts pour les vitesses de lien les plus courantes que nous reproduisons ci-dessous. Le tableau 7.1 en est un exemple.

Tableau 7.1 Représentation du tableau des coûts en rapport aux vitesses de liens

Largeur de bande	Coût pour le *Spanning Tree*
4 Mbps	250
10 Mbps	100
16 Mbps	62
45 Mbps	39
100 Mbps	19
155 Mbps	14
622 Mbps	6
1 Gbps	4
10 Gbps	2

Le protocole *Spanning Tree* permet la transmission de BPDUs (*Bridge PDU*) entre ponts pour transmettre les informations requises et déterminer lesquelles des lignes doivent ou ne doivent pas être empruntées de manière à éviter les boucles.

Les BPDUs comprennent les informations essentielles pour permettre aux ponts de décider du meilleur chemin à prendre dans le but de former une arborescence et éviter les boucles.

Les BPDUs de configuration comprennent, parmi bien d'autres, les quatre champs suivants :

- Identifiant du pont racine (*Root BID*)
- Coût du chemin racine (*Root Path Cost*)
- Identifiant du pont qui transmet (*Sender BID*)
- Identifiant du port (*Port ID*)

Une description plus complète des BPDUs sera faite plus loin.

Le protocole *Spanning Tree* réalise ses objectifs en passant par les trois étapes suivantes :

- Établir l'unique pont racine dans le réseau (*Root Bridge*)
- Définir les ports racine (*Root Port*)
- Choisir les ports désignés dans chaque pont (*Designated Port*)

7.7.1 Procédure de l'établissement de l'unique pont racine dans le réseau (Root Bridge)

Lorsqu'un pont est initialisé, il transmet des BPDUs de configuration aux ponts voisins. Dès le début, ce pont s'identifie comme étant le pont racine. Il met donc son identifiant dans le champ *Root BID* de chaque BPDU qu'il transmet. Si le pont reçoit un BPDU d'un autre pont avec une valeur de *Root BID* plus petite que la sienne, il adopte alors cette valeur et la place dans le champ *Root BID* de ses nouveaux BPDUs.

Éventuellement et après l'échange de plusieurs BPDUs, entre tous les ponts du réseau local, celui qui a la plus petite valeur de BID est considéré comme étant le pont racine (*Root Bridge*).

7.7.2 Procédure de la définition des ports racine (Root Port)

La seconde étape vise à définir tous les ports racine au sein du réseau local. Le port racine est celui de chacun des ponts qui se trouve le plus près du pont racine.

Avant de compléter cette étape, il importe d'établir le coût du chemin (*Path Cost)* entre les ponts conformément au tableau 7.1. Chaque pont établit le coût des liens *(Path Cost*) associés aux ports d'entrées.

Le coût des chemins racine (R*oot Path Cost*) est ensuite établi. Le pont extrait le coût du chemin racine (R*oot Path Cost*) des BPDUs reçus et l'ajoute au coût du chemin (*Path Cost*) associé au lien d'entrée du BPDU.

Le pont reçoit des BPDUs dans chacun de ses ports et, pour chacun d'entre eux, procède au calcul avec les données qu'il obtient, notamment le coût du chemin qu'il connaît et le coût du chemin racine reçu.

Chaque pont peut donc décider de la plus petite valeur à partir des calculs de coûts du chemin racine. Le pont désigne le port ayant la plus petite valeur comme étant le port racine.

7.7.3 Procédure du choix du port désigné dans chaque pont (Designated Port)

La troisième étape consiste à désigner, parmi certains ponts, le port d'entrée qui sera exploité par chaque segment pour atteindre le pont racine (R*oot Bridge*). Dans le réseau, chaque segment doit avoir un port désigné. En effet, grâce à cette étape, la boucle sera contournée.

Tous les ports du pont racine sont des ports désignés. Ceci est établi en raison du fait que le coût de cheminement dans le pont racine est à 0.

Lorsqu'un pont doit décider si un de ses ports est un port désigné pour un segment, il compare le coût du chemin racine (qui le mène au pont racine) pour le port sur le segment en question au coût du chemin racine contenu dans les BPDUs qu'il reçoit sur le même segment. Si son coût de chemin racine est le plus bas, il assigne alors le port sur ce segment comme port désigné. Par contre, si la valeur reçue d'un BPDU est moindre, il assigne le port sur ce même segment comme port non désigné.

Si la valeur du coût du chemin racine sur le port est égale au coût du chemin racine retrouvé dans un BPDU reçu, il compare son propre BID à celui de l'identifiant du pont qui a transmis (*Sender BID*) ce BPDU. Si son BID est plus petit, il assigne alors le port sur ce segment comme port désigné, autrement, il l'assigne comme port non désigné.

7.7.4 Exemple d'application du protocole Spanning Tree

Prenons l'exemple du réseau de la figure 7.15. Nous pouvons y identifier une boucle. Le protocole *Spanning Tree* y sera appliqué pour l'éliminer.

Les ponts de ce réseau doivent tout d'abord établir lequel sera le pont racine. Supposons que le BID des ponts soit égal à leur numéro de pont. Si ceux-ci sont initialisés, chaque BPDU qui sera transmis inclura le BID du pont qui l'a créé dans le champ *Root BID*. Après un certain nombre d'échanges, tous les BPDUs auront la valeur du BID du pont 1 dans leur champ *Root BID*. Cela nous indique que le pont 1 a été désigné pont racine.

Chacun des ponts établit le coût de chemin pour chaque segment auquel ils sont reliés. Puisque tous les segments de notre exemple sont de 10 Mbps, le coût de chemin de chaque segment sera de 100 conformément au tableau 7.1.

Le pont racine transmet un BPDU avec la valeur du champ *Root Path Cost* qui est de 0. Le pont 2 reçoit au port 1 ce BPDU et ajoute le contenu du champ *Root Path Cost* (qui est de 0) à la valeur du *Path Cost* du segment 2 (qui est de 100). Le pont 2 transmet un BPDU sur le segment 1 par le port 2 avec la valeur calculée du *Root Path Cost* de 100 (0+100). Le pont 3 reçoit ce BPDU par le port 2. Pour lui aussi la valeur du *Path Cost* du segment 1 est de 100. Il additionne cette valeur au contenu du champ *Root Path Cost* (valeur 100) qui se trouve dans le BPDU reçu et obtient une valeur de 200.

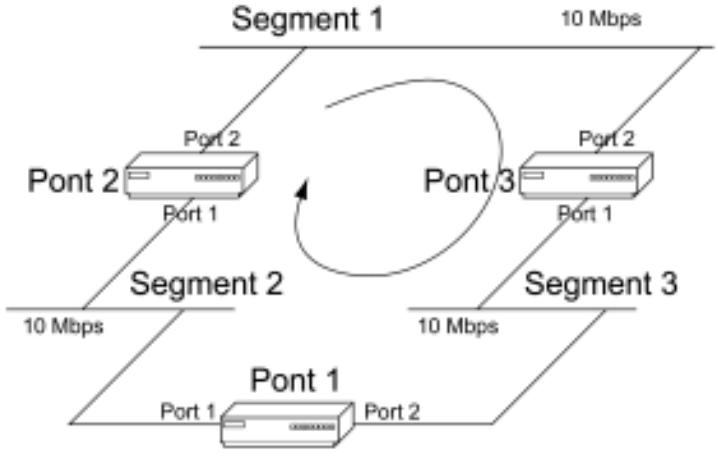


Figure 7.15 Exemple de réseau local avec boucle

Par contre, le pont 3 reçoit aussi sur son port 1 un autre BPDU transmis par le pont 1 à partir de son port 2. En effectuant son calcul, le pont 3 établit le coût du *Root Path Cost* à 100. Ceci étant plus petit que ce qui a été calculé avec le BPDU du pont 2, il choisit son port 1 comme port racine. Le pont 2 effectue le même calcul et opte pour son port 1 comme port racine.

Les ports du pont racine sont toujours des ports désignés. Cela signifie que les segments qui y sont reliés sont toujours en opération. Les segments 2 et 3 vont toujours être utilisés pour transmettre les données aux ponts 2 et 3. Par contre, le segment 1 cause un problème parce qu'il forme la boucle. Il ne faut donc pas que l'un des ports du pont 2 ou 3, qui y est connecté, soit désigné et l'autre non désigné. Le choix entre les deux ports se fait en sachant lequel des deux BID est le plus bas. Dans notre cas, il s'agit du pont 2. Ainsi, le port 2 du pont 2 sera le port désigné. En d'autres termes, cela revient à dire que les trames de toutes les stations connectées au segment 1 vont être transmises à travers le port 2 du pont 2 et non par le pont 3. La figure 7.16 illustre en partie l'exemple.

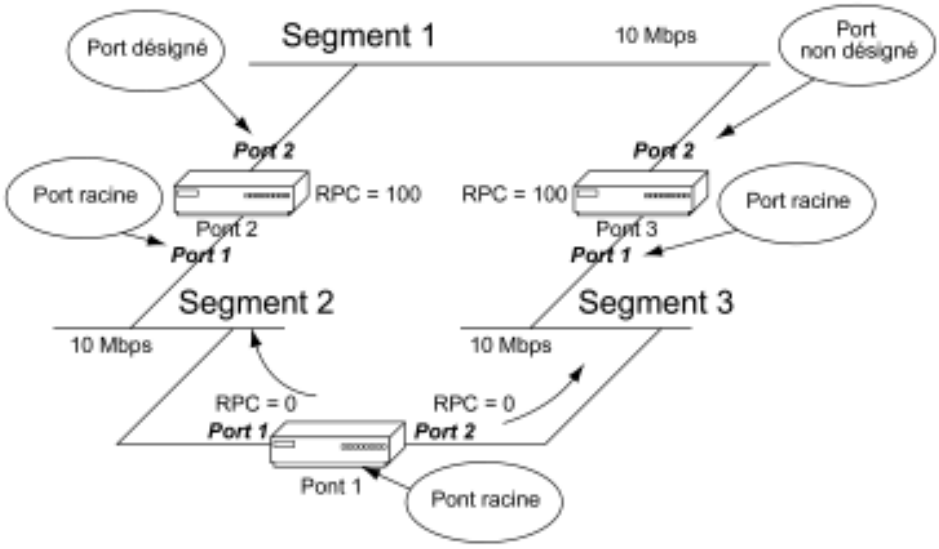


Figure 7.16 Exemple de l'application du STP

Définitions :

Root bridge : Pont ayant la plus petite valeur d'identifiant.

Path cost : Coût de la transmission d'une trame à travers ce chemin ou port.

Root port : Chaque pont découvre le premier saut vers le chemin le moins coûteux à destination du pont racine.

Root path cost : Pour chaque pont, le coût du chemin vers le pont racine avec un coût minimum.

Designated bridge, designated port : Pont sur le LAN qui offre le chemin avec un coût minimum (*minimum cost path*) jusqu'au pont racine.

Le *Spanning Tree* est construit comme suit :

1- Identifier le pont racine.

2- Identifier le port racine sur tous les autres ponts. C'est-à-dire choisir le *Chemin le moins coûteux en partant de chaque pont*. Chaque pont calcul le coût de tous les chemins possibles menant au pont racine. Il choisit le moins coûteux et désigne le port qui l'y mène : port racine

3- Identifier le port désigné sur chaque LAN. En choisissant le *Chemin le moins coûteux en partant de chaque segment de réseau*. Le chemin le plus court qui permet de connecter un segment au pont racine doit être identifié. Les ponts qui vont permettre au segment d'atteindre cet objectif identifient le port rattaché à ce segment comme port désigné.
   Dans le cas où plusieurs ponts ont le même coût de chemin racine, le pont avec la plus grande priorité en sera le pont désigné. Si le pont désigné a deux ou plusieurs ports attachés au LAN, le port dont l'identifiant a la plus petite valeur est choisi.
4- Les ports qui ne sont pas identifiés ports racines ou ports désignés sont des ports bloqués.

7.7.5 Les trames BPDU

Les trames BPDU sont échangées entre les ponts. De cette façon, chacun des ponts peut déterminer la position du pont racine et faire connaître ses paramètres aux autres ponts du réseau. C'est aussi grâce aux BPDUs que tout changement à la topologie amorce une nouvelle configuration de l'arbre développée par le *Spanning Tree*.

Deux types de BPDU sont échangés entre les ponts :

- BPDU de configuration
- BPDU information de changement de topologie ou TCN (*Topology Change Notification*) BPDU.

Les BPDU de configuration sont générés par le pont racine et sont distribués aux autres ponts au moyen des chemins tracés par le *Spanning Tree*. Les TCN BPDUs sont transmis dans l'autre sens, c'est-à-dire des autres ponts au pont racine pour aviser ce dernier du changement de topologie.

Le format du BPDU de configuration revêt le format illustré à la figure 7.17.

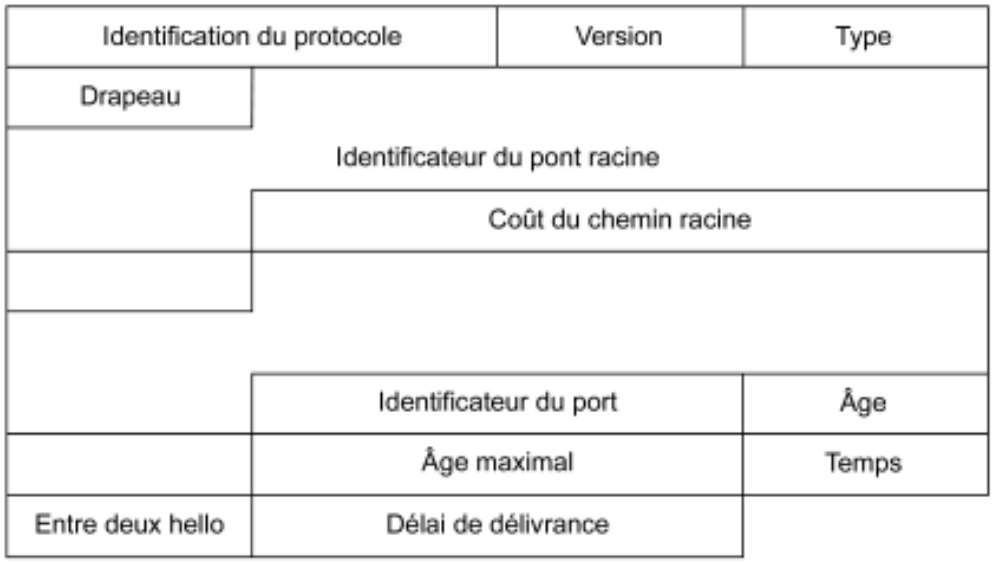


Figure 7.17 Format du BPDU de configuration

Les champs du BPDU de configuration sont les suivants :

- « Identification du protocole » – Ce champ long de deux octets est à 0 pour indiquer que c'est un BPDU.
- « Version » – Ce champ long d'un octet correspond à la version 0.
- « Type » – Champ long d'un octet. La valeur 0 indique qu'il s'agit d'un BPDU de configuration, alors qu'une valeur de 128 indique qu'il s'agit d'un TCN BPDU.
- « Drapeau » – Dans ce champ d'un octet, le bit le moins significatif est le TC (*Top Change*) et signale qu'il faut effectuer un changement de configuration. Le bit le plus significatif est le TCA (*Top Change Ack*). Il est utilisé pour indiquer l'accusé de réception d'une trame TCN BPDU.
- « Identification du pont racine » – Champ long de huit octets. Les deux octets les plus significatifs sont utilisés pour indiquer la priorité. Les six autres octets indiquent l'adresse MAC du pont racine.
- « Coût du chemin racine » – Champ long de trois octets. Valeur utilisée dans l'établissement du *Spanning Tree*.
- « Identification du pont » – Ce champ qui transmet le message est codé de la même manière que pour l'identification du pont racine.
- « Identification du port » – Champ long de deux octets.
- « Âge du message » - Ce champ long de deux octets comprend un compteur en multiple de 1/256 secondes. Si le pont ne reçoit plus de

message BPDU de configuration du pont racine, ce compteur est incrémenté dans les BPDU que le pont transmet.

- « Âge maximal du message » – Étape que doit atteindre le compteur d'âge du message avant de l'effacer.
- « Temps entre deux Hello » – Ce champ de deux octets indique le temps entre deux BPDU de configuration transmis par le pont racine. Le temps alloué entre deux BPDU est connu sous le nom de *Hello Time*.
- « Délai de délivrance » – Ce champ de deux octets représente la durée passée à l'état écoute et à l'état apprentissage dans l'établissement du *Spanning Tree*.

Le TCN BPDU ne comprend que les trois premiers champs du BPDU de configuration.

7.7.6 Procédure dans le changement de topologie

Le changement de topologie est une situation courante et la procédure automatique enclenchée pour l'effectuer est très efficace. La procédure en question utilise des BPDUs.
L'exemple suivant décrit la procédure telle qu'elle est déployée lors d'un changement dans la condition des liens ou des ponts. Nous allons voir comment cette procédure traite les messages TCN BPDUs et BPDUs de configuration.

Dans le cas où un lien interconnecté au port à l'état réacheminement (*Forwarding*) ou à l'état apprentissage (*Learning*) est défectueux, le pont doit bloquer ce port et transmettre lui-même le TCN BPDU.

Le réseau de la figure 7.18 a défini une arborescence à l'aide du *Spanning Tree*. Le pont A est la racine et le lien associé au port 2 du pont E est bloqué. L'accès à la station 4 se fait par le pont C à travers un port désigné, soit le port 2.

Supposons que le port 2 du pont C détecte une anomalie persistante sur son lien, le pont C fait passer ce port de l'état réacheminement à l'état blocage (*Blocking*). Sans l'intervention des TCN BPDU, la reconfiguration du *Spanning Tree* est lente à cause du temps nécessaire pour la mise à jour des tables d'acheminement des ponts. Si ces tables ne sont pas utilisées, elles sont mises à jour aux 300 secondes. Les ponts mettront donc 300 secondes avant de réaliser que l'accès du pont C à la station 4 ne peut pas se faire. Ceci veut dire qu'une partie du réseau peut être en panne pendant cinq minutes.

L'utilisation des messages TCN BPDU permettra une reconfiguration beaucoup plus rapide comme nous l'illustrerons avec notre exemple.

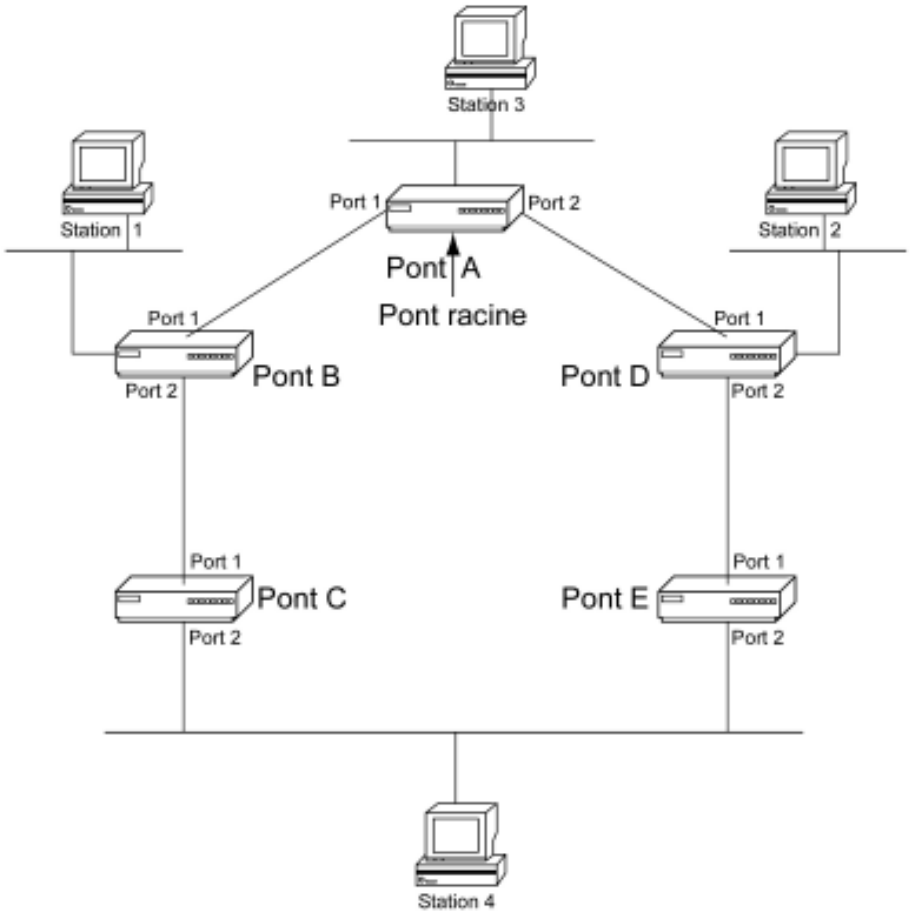


Figure 7.18 Changement de topologie dans le *Spanning Tree*

Lorsque le pont C fait passer l'état du port 2 d'apprentissage à blocage, il transmet un message TCN BPDU de son port racine, le port 1, vers le pont racine. Ce message est transmis après chaque intervalle de temps Hello qui est défini pour chaque pont, et ce, jusqu'à la réception d'un accusé de réception du message.

Seul le port désigné des autres ponts qui reçoivent ce message doit réagir. Étant un port désigné, le port 2 du pont B répond au message TCN BPDU envoyé par le pont C en lui transmettant un BPDU de configuration avec le drapeau TCA positionné. Le drapeau TCA est l'accusé de réception attendu par le pont C. Le pont B relaie également le message TCN BPDU au pont racine A par son port racine, le port 1.

À la réception du message TCN BPDU, le pont racine A transmet au pont B un message BPDU de configuration en y positionnant le bit TCA pour acquitter le message reçu et le bit TC pour amorcer le changement du temps de mise à jour des tables d'acheminement.

Par la suite, le pont racine A transmet, à tous les ponts, des messages BPDUs de configuration en ne positionnant que le drapeau TC. Ces messages BPDUs de configuration sont transmis continuellement pour une durée totale équivalente à la valeur de « âge maximal » + « délai de délivrance », ce qui représente approximativement 35 secondes.

À la réception des BPDUs de configuration avec le drapeau TC positionné, les ponts abrègent le temporisateur de mise à jour des tables d'acheminement de 300 secondes à la valeur contenue dans le champ « délai de délivrance », ce dernier étant normalement de 15 secondes.

Une fois le temps du temporisateur expiré, le pont efface les entrées de ses tables d'acheminement. Notons que ces entrées indiquaient le port emprunté pour atteindre une destination voulue et définie par son adresse MAC. Lorsque l'adresse de destination n'est pas comprise dans sa table d'acheminement, le pont qui reçoit une trame va simplement la diffuser à travers tous ses autres ports.

Si après l'épuration des temporisateurs, la station 2 transmet une trame destinée à la station 4, tous les ponts qui la reçoivent vont la diffuser sur tous leurs ports, excepté le port de réception. La station 4 reçoit cette trame par le port 2 du pont E. Elle y répond en empruntant le même port. Le pont E constate donc que la station 4 se trouve sur son port 2. De la même façon, tous les autres ponts découvrent la position des stations 2 et 4 et prennent connaissance des ports qu'ils devront utiliser pour les atteindre.

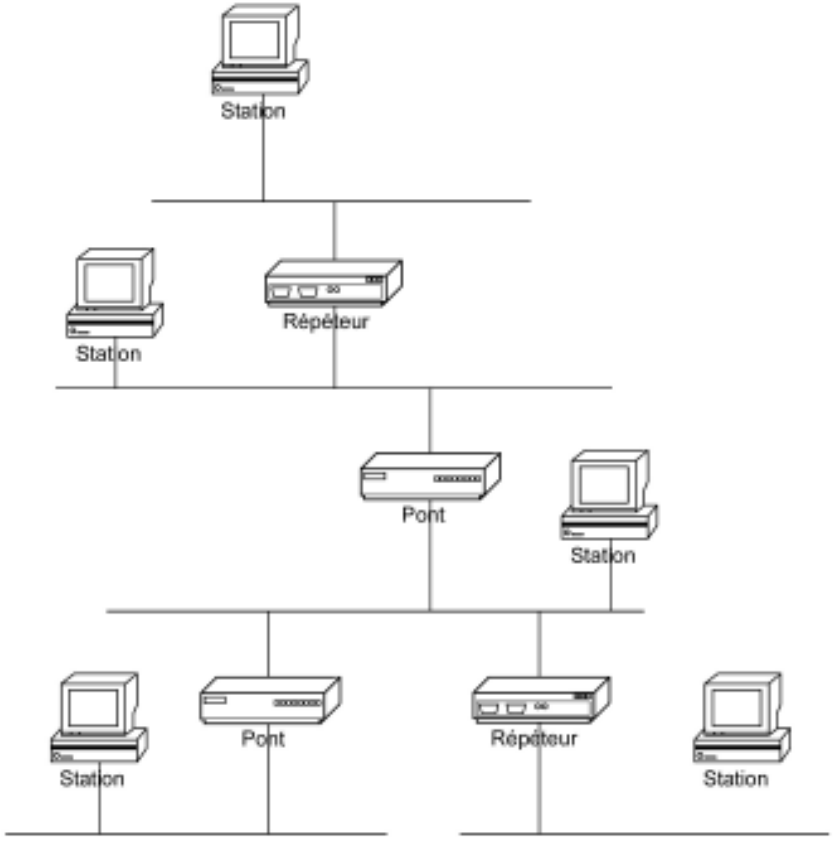
7.8 Exercices

7.1

Il a été fait mention qu'un réseau local traditionnel utilisant efficacement les ponts, comprend la répartition du trafic dont 80 % est destiné à l'intérieur du segment et 20 % à l'externe. Si le trafic externe venait à augmenter au-delà de 20%, expliquez comment et pourquoi le pont commencerait à agir comme un concentrateur.

7.2

Dans le diagramme suivant, indiquez les domaines de collision et de diffusion.



7.3

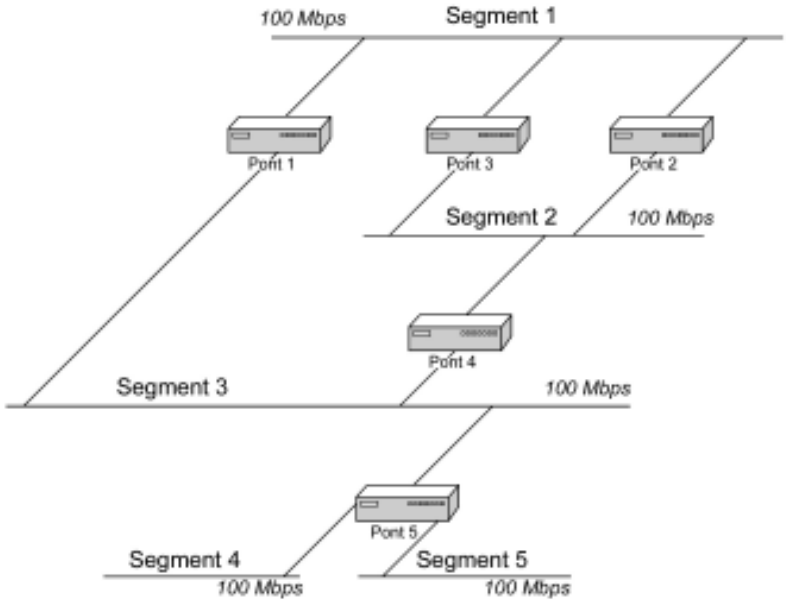
Établissez le tableau global de l'acheminement fixe ainsi que les tableaux d'acheminement pour les ponts A, B, et G de la figure 7.9.

7.4

Dans un réseau local Ethernet comprenant des boucles, si le *Spanning Tree* n'est pas utilisé, le processus d'apprentissage de l'adresse va se faire à l'infini, ce qui rend l'acheminement de trame instable. Expliquez pourquoi.

7.5

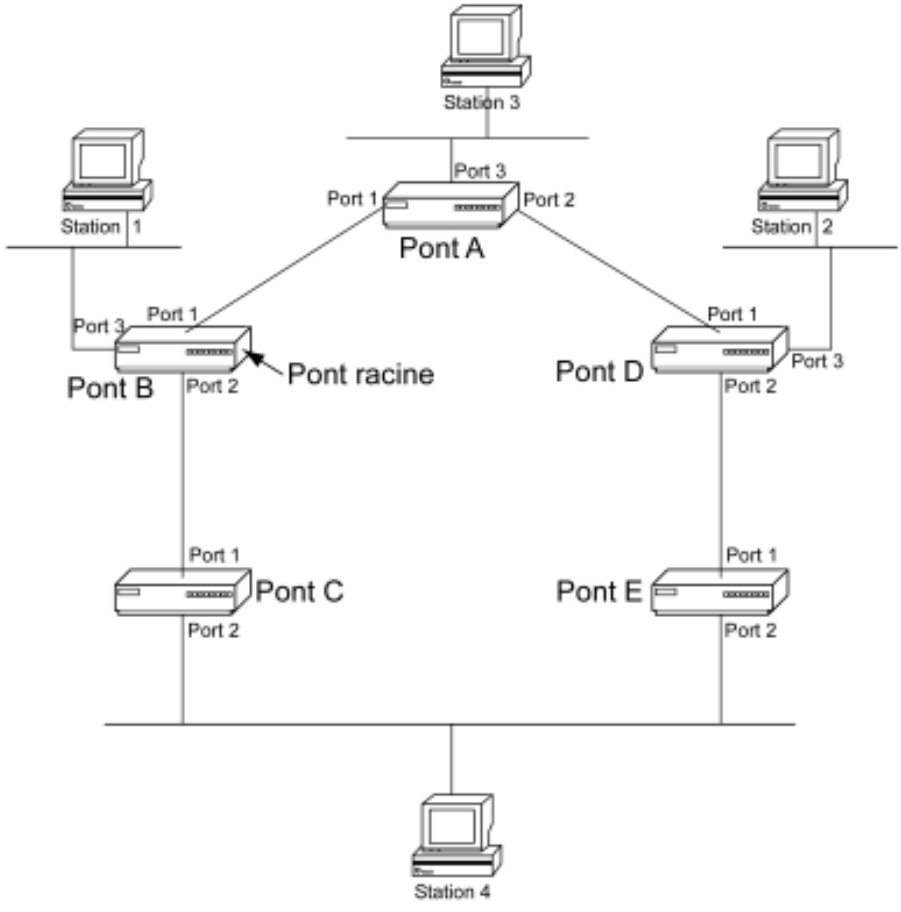
Appliquez le *Spanning Tree* sur le réseau suivant et trouvez le pont racine, les ports *racine* et les ports désignés.



7.6

Dans le réseau suivant, le pont racine est le pont B. Le port 2 du pont E est à l'état réacheminement (*Forwarding*); c'est-à-dire que le lien sur ce port est actif. C'est le port 2 du pont C qui est à l'état blocage.

Le pont E détecte par la suite un problème au port 2. Il le passe donc de l'état réacheminement à l'état blocage. Décrivez, en détail, comment les ponts réagissent en utilisant les messages TCN BPDU et BPDU de configuration.



8 Les réseaux locaux virtuels (VLAN)

8.1 Introduction

Le domaine de diffusion dans un réseau local permet à chaque terminal s'y trouvant de distribuer des informations à tous les autres terminaux en ne les transmettant qu'une fois et sans craindre qu'un terminal situé hors du domaine de diffusion puisse les intercepter.

Ce domaine de diffusion ne fonctionne que sur la couche MAC qui fait partie de la couche liaison du modèle OSI. Les routeurs qui agissent à la couche 3 et qui se trouvent dans le réseau vont empêcher la diffusion de propagation au-delà de ses interfaces. Ainsi, les domaines de diffusion peuvent être définis physiquement par des routeurs.

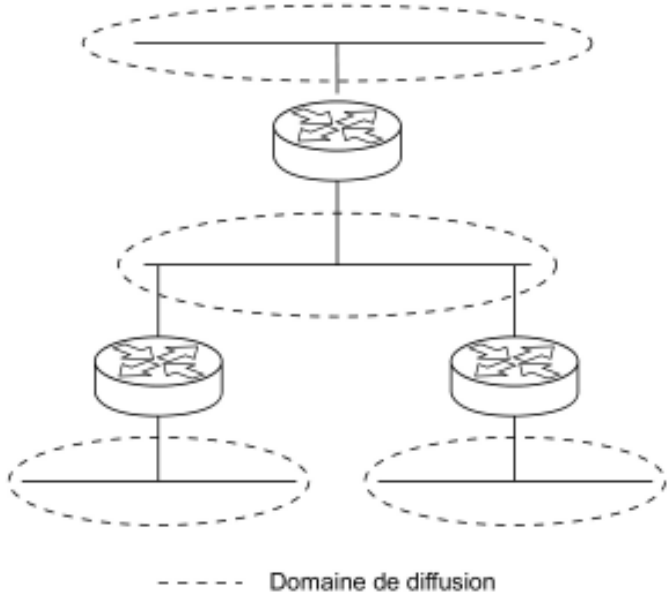


Figure 8.1 Domaines de diffusion définis par les routeurs

Avec les routeurs, les domaines de diffusion sont bien définis et dans la topologie du réseau ils sont facilement reconnaissables. Ce système est illustré à la figure 8.1. Par contre, lorsque des commutateurs sont utilisés pour configurer un réseau local, les domaines de diffusion ne peuvent pas être aussi facilement illustrés (figure 8.2).

Les commutateurs étant des ponts multi-ports et fonctionnant sur la couche MAC peuvent également engendrer des domaines de diffusion dans la mesure où leur configuration leur permet de les définir.

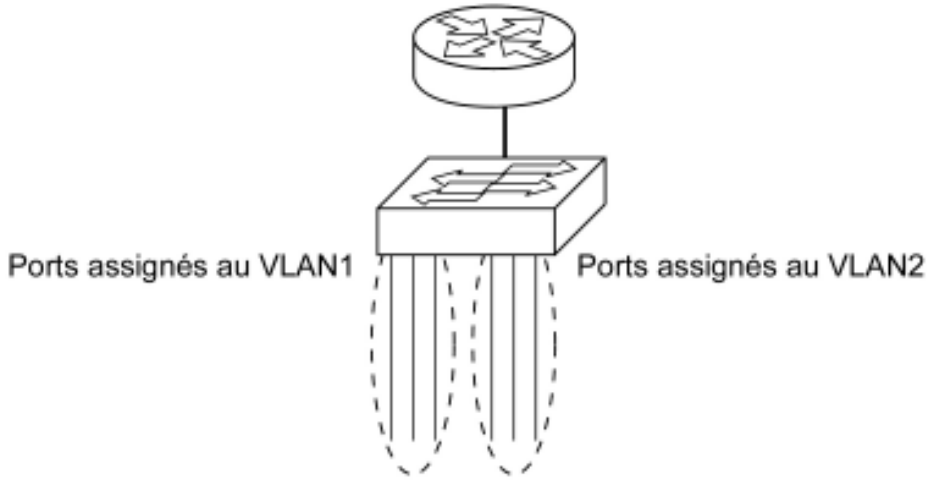


Figure 8.2 Commutateur et VLANs

Un réseau local virtuel ou VLAN (*Virtual Local Area Network*) représente le domaine de diffusion existant dans un réseau local. Cette définition ne décrit pas à sa juste valeur ce que le VLAN peut faire ni ce en quoi il s'avère utile.

Envisageons le cas type du réseau d'une organisation dont les bureaux se trouvent dans un campus ou un grand bâtiment. Cette organisation compte plusieurs départements dont le personnel de chacun de ces départements est réparti à travers les étages et partage des locaux avec du personnel des autres départements. La direction désire tout d'abord mettre sur pied un réseau sécurisé reliant toutes les employées du même département. De plus, il importe que les interconnexions entre hôtes du même service soient inter reliées et ne traversent pas les segments des autres services. Il est impensable de concevoir des réseaux autonomes réservés à chacun des services en raison des coûts et de l'inefficacité d'avoir plusieurs réseaux pour une seule et même entreprise.

Tout service devrait pouvoir diffuser des informations à tous ses bureaux sans pour autant que les autres services n'aient la possibilité d'y accéder.

Un VLAN prenant en charge, par exemple, tous les hôtes du département d'ingénierie serait la solution idéale. De la sorte, tous les hôtes du département d'ingénierie se trouveront sur le même domaine de diffusion tout en étant séparés des autres hôtes, et ce, sans tenir compte de l'emplacement des bureaux.

Pour pouvoir concevoir une telle configuration, il nous faut recourir à des commutateurs qui, dans la définition de leurs accès, comportent la possibilité de créer des VLANs. La figure 8.3 en donne un bon exemple.

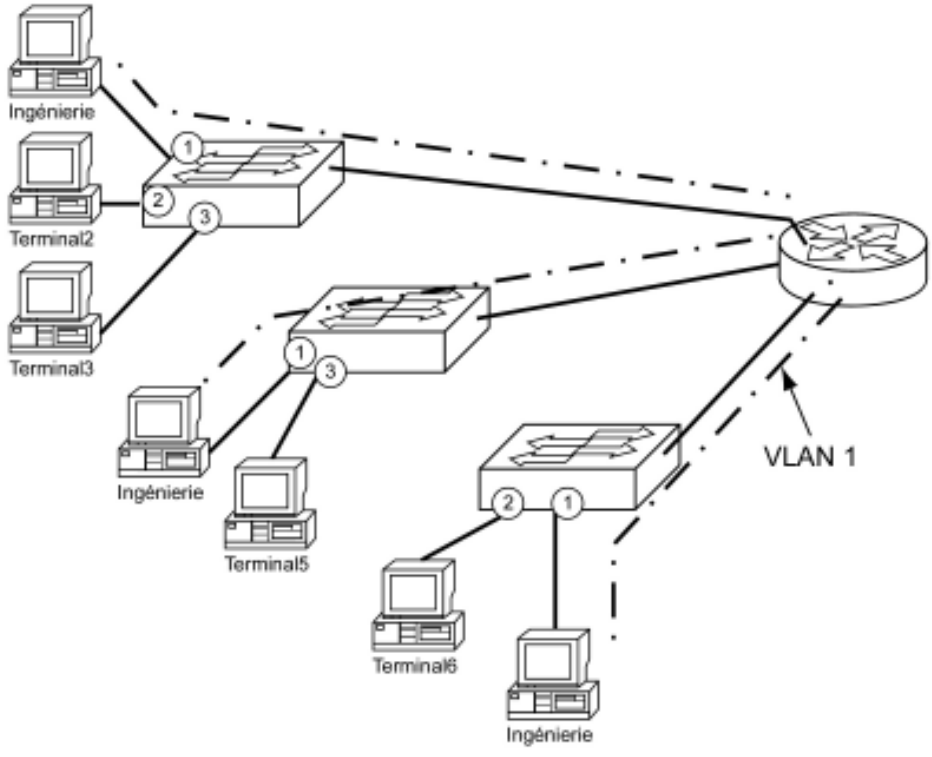


Figure 8.3 Configuration d'un VLAN dans un réseau commuté

Avec le VLAN, le trafic est réparti de façon plus précise. Ainsi, l'utilisation de la largeur de bande du réseau s'en trouve réduite. Lors de la diffusion d'informations, les commutateurs n'envoient les données que vers des terminaux se trouvant dans le même VLAN.

Lorsque les données d'un VLAN sont transmises à un autre, le transfert doit se faire par commutation de la couche 3. Les VLANs sont interconnectés par des routeurs. Le Catalyst de Cisco est un exemple du genre d'équipement comprenant des commutateurs et routeurs pouvant définir des VLANs et permettre que le routage se fasse de l'un à l'autre.

Un commutateur peut être configuré pour créer des ponts virtuels et pour que chacun des domaines de diffusion défini soit un VLAN. Cependant le trafic d'un VLAN ne peut-être acheminé à un autre VLAN sans qu'il y ait d'équipement d'interfonctionnement de la couche 3.

8.1.1 Types de VLANs

Les VLANs peuvent être élaborés à partir des éléments distincts suivants :

1. configuration des ports du commutateur
2. adresse MAC de la station
3. protocole de la couche 3 et
4. sous-réseau IP

8.1.2 Configuration des ports du commutateur

Le type de VLAN qui est créé à partir de la configuration des ports en est la conception la plus simple. Les ports des commutateurs sont associés à un VLAN et chacun d'entre eux comprend un numéro d'identification. L'adresse MAC du port du commutateur est associée au VLAN qui lui est assigné. Cette méthode, telle qu'illustrée à la figure 8.4, est facile à configurer. Il s'agit simplement d'associer les numéros de VLAN aux adresses MAC des ports des commutateurs dans des tables tenues par ces derniers. Toutes les communications, tout comme la diffusion de données par un terminal dans un VLAN, seront envoyées par les commutateurs aux ports appartenant au même VLAN.

Cependant, le désavantage de cette conception repose sur le fait que le VLAN est associé aux ports et non aux terminaux. Ainsi, si un terminal est déplacé d'un port à un autre et qu'il désire conserver le même VLAN, il doit demander qu'une reconfiguration soit faite lui assignant son VLAN au nouveau port du commutateur et en en retirant l'ancien.

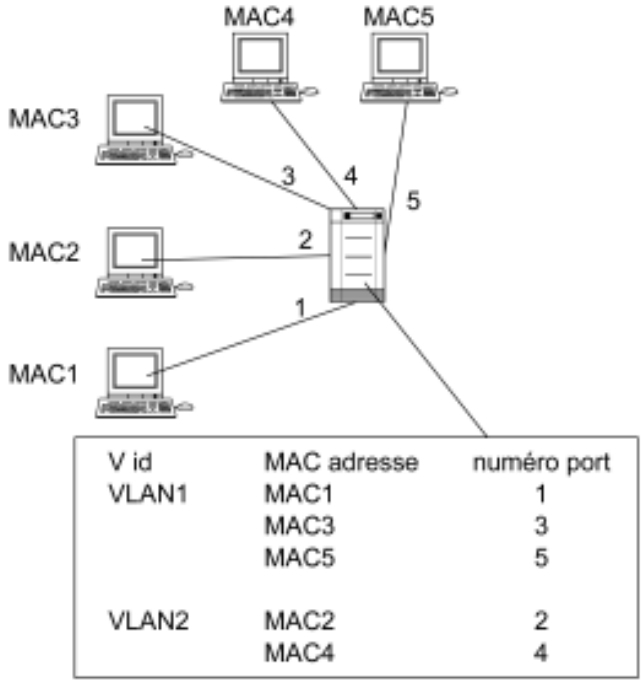


Figure 8.4 Conception d'un VLAN à partir des ports

8.1.3 Adresse MAC de la station

Le VLAN conçu à partir de l'adresse MAC de la station permet à un terminal de se déplacer sans devoir reconfigurer ses ports d'accès. L'adresse MAC de la station permet au commutateur d'identifier le VLAN auquel elle appartient. Comment identifier la station et en connaître son adresse MAC?

Lorsque la station transmet des trames, tel qu'illustré à la figure 8.5, le commutateur en prélève l'adresse source MAC et, en consultant la table de VLANs basée sur l'adresse MAC, il peut alors savoir de quel VLAN cette station relève. Le commutateur peut ainsi mettre à jour la table de VLANs en y indiquant le port auquel la station est connectée. La table de VLANs comprend le numéro de VLAN, les adresses MAC des terminaux associées et les numéros de port où ces adresses ont été détectées.

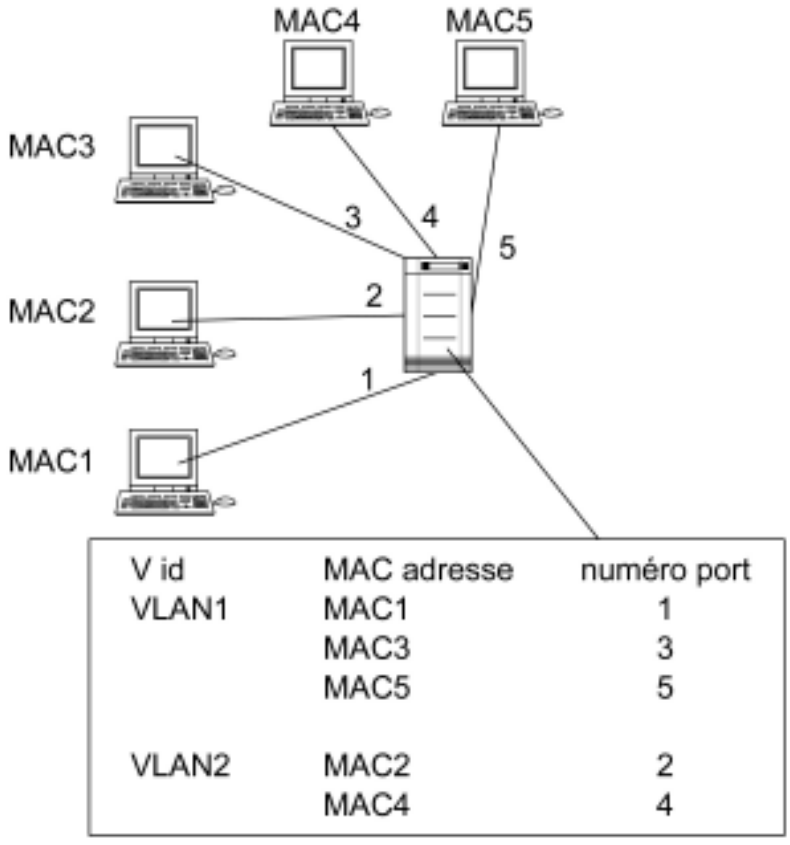


Figure 8.5 Conception d'un VLAN à partir de l'adresse MAC de la station

Le seul problème avec ce genre de conception repose sur le fait que les adresses MAC doivent être entrées manuellement dans les tables de VLANs pour certains types de commutateurs.

8.1.4 Protocole de la couche 3

Le VLAN conçu à partir du protocole de la couche 3 permet de faire passer les trames transportant des paquets d'un protocole spécifique de la couche réseau à travers un commutateur sans recourir au routeur. Un VLAN tel qu'illustré à la figure 8.6, et réservé pour le protocole IPX, acheminera les trames comprenant des paquets IPX à toutes les stations qui lui sont associées. La table de VLANs indiquera également le port du commutateur d'où ce terminal, qui a recours au protocole IPX, transmet. Les commutateurs pourront ainsi distribuer les données d'un type de protocole par le VLAN qui lui est associé.

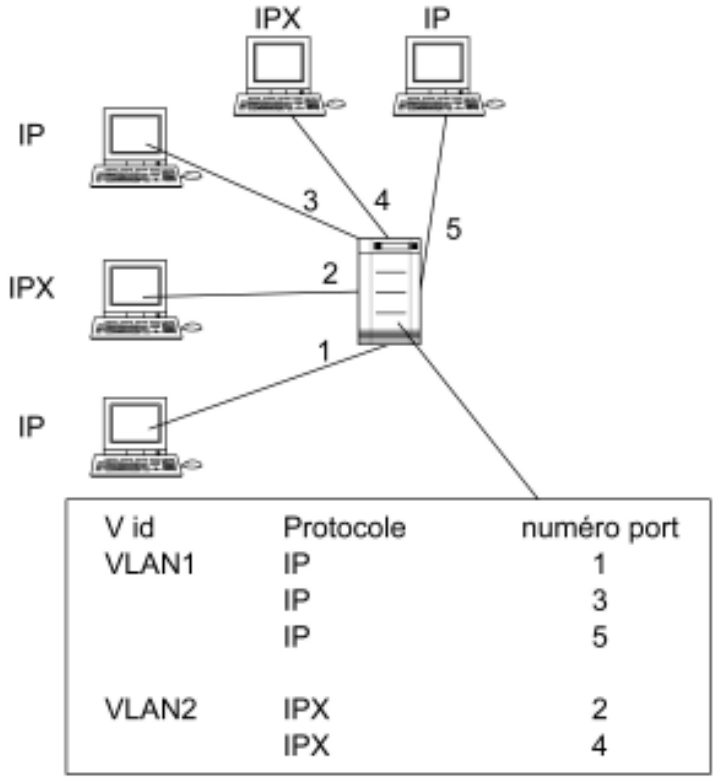


Figure 8.6 Conception d'un VLAN à partir du protocole de la couche 3

8.1.5 Sous-réseau IP

Le VLAN conçu à partir du sous-réseau IP inclut toutes les stations figurant dans le même sous-réseau. Une station n'est membre d'un VLAN qu'à la condition que son adresse IP fasse partie du sous-réseau du VLAN. La station est identifiée lorsqu'elle transmet des paquets IP. Son adresse est prélevée par le commutateur qui va l'associer au VLAN de son sous-réseau. Le commutateur prend également note du port d'où le paquet émane.

En utilisant le VLAN conçu à partir du sous-réseau IP, le commutateur peut créer différents domaines de diffusion sans recourir au routeur. En outre, les stations dotées de sous-réseaux distincts, peuvent coexister sur le même port que celui du commutateur grâce au VLAN conçu à partir du sous-réseau IP. En d'autres termes, le commutateur est en mesure d'identifier plusieurs VLANs sur un seul et même port.

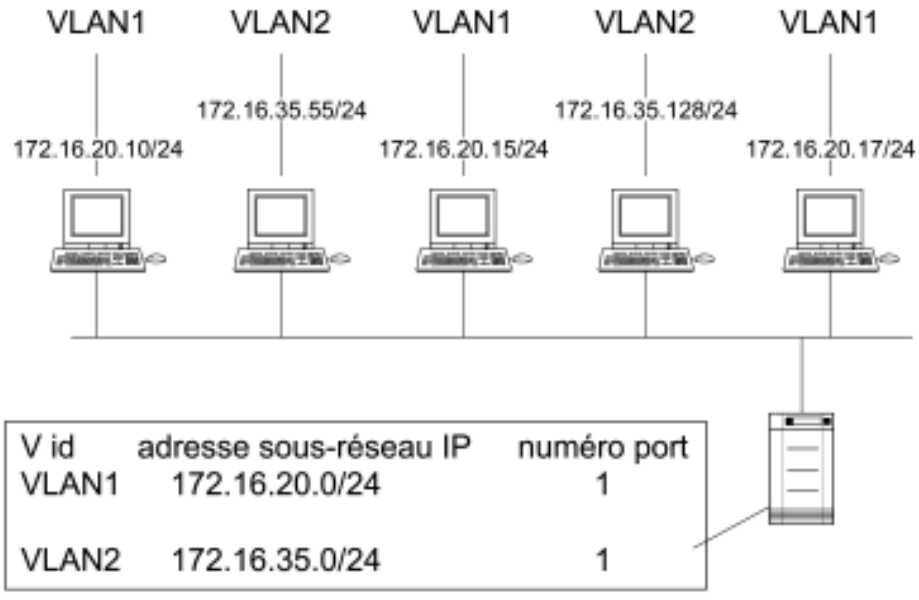


Figure 8.7 Conception d'un VLAN à partir du sous-réseau IP

L'exemple illustré à la figure 8.7 nous montre de quelle manière plusieurs stations peuvent cohabiter sur le même port sans pour autant créer d'interférences les unes par rapport aux autres. Lorsqu'il y a diffusion sur le VLAN1 seuls les terminaux répondant à ce VLAN traiteront les informations. Cet exemple est un cas extrême où le commutateur n'est pas exploité de façon optimale. Le fait d'avoir plusieurs stations sur le même segment, même avec différents VLANs, n'empêchera pas les collisions. Pour éviter toute collision, le commutateur devrait avoir un terminal par port.

Ce mécanisme de VLAN peut se faire localement dans un commutateur sans autre indication dans la mesure où les VLANs sont contraints d'avoir un même commutateur. Lorsqu'il faut étendre le VLAN sur plusieurs commutateurs, il faut pouvoir identifier les paquets afin de savoir qu'ils appartiennent à ce VLAN. De plus les commutateurs doivent pouvoir reconnaître le numéro de VLAN des autres commutateurs de façon à respecter l'uniformité des VLANs dans le réseau. Ces fonctions sont rendues possibles grâce à l'étiquetage du VLAN.

8.2 Étiquetage de VLAN selon le IEEE802.1Q

Le protocole IEEE802.1Q permet d'obtenir une conception normalisée et d'étendre la reconnaissance de VLANs à travers plusieurs commutateurs. Ce protocole ne tient pas compte des types de VLANs voire même de leur

définition. Le seul type de VLAN dont il est fait mention repose sur la conception du VLAN basé sur les ports de commutateurs.

L'opération d'infrastructure d'un VLAN exige que l'information des VLANs se tienne et soit logique dans tous les commutateurs. Ces informations peuvent être adressées manuellement à chacun des commutateurs. Par ailleurs, le protocole GARP (*Generic Attribute Registration Protocol*) avec GVRP (*GARP VLAN Registration Protocol*) qui est le protocole d'enregistrement du VLAN, peut-être utilisé pour adresser les informations automatiquement.

Le protocole GVRP propage, à tous les commutateurs du réseau, l'information relative à l'identification des VLANs, par le biais du VPID (*VLAN Protocol Identification*). Seules les stations qui transmettent des messages GVRP pour s'enregistrer dans un VLAN sont reconnues par les commutateurs. Cette information est adressée à tous les commutateurs.

Conformément au protocole IEEE802.1Q, l'étiquette VLAN d'une trame MAC comprend les trois éléments suivants :

- L'identifiant du protocole VLAN, le VPID qui indique que la trame est étiquetée VLAN;
- L'information de contrôle de l'étiquette TCI (*Tag Control Information*) et;
- Le champ d'information de routage incorporé E-RIF (*Embedded Routing Information Field*) qui est optionnel et qui permet d'incorporer le routage par la source dans l'étiquette du VLAN. Utilisé surtout par l'anneau à jeton.

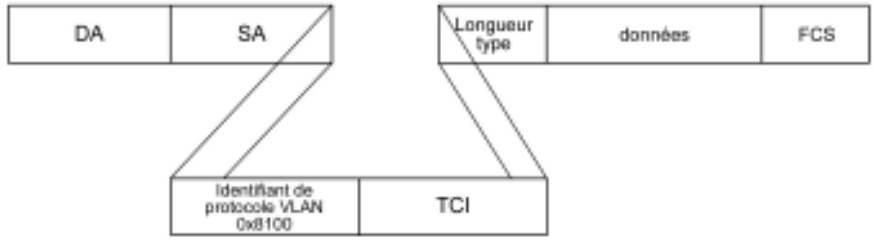


Figure 8.8 Trame VLAN IEEE802.1Q

L'identifiant du protocole a une longueur de 2 octets. Lorsque la trame Ethernet est décodée, le champ qui suit les adresses MAC indique le protocole qui se trouve dans les données. Pour le protocole VLAN le code de ce champ a une valeur égale à 0x8100 (figure 8.8). Cela nous indique que le champ suivant est le TCI et que les champs qui suivent le TCI sont ceux que l'on a l'habitude de trouver dans une trame Ethernet.

Le champ TCI comprend : trois bits chargés d'assurer l'information de priorité de l'utilisateur, 1 bit CFI (*Canonical Format Indicator*) indiquant le format de codage de la trame et, l'identifiant du VLAN qui fournit le numéro de VLAN auquel appartient la trame, sur 12 bits.

Il est à noter que la trame Ethernet est accrue de quatre octets en raison de la présence de l'étiquette de VLAN IEEE802.1Q.

Le IEEE802.1Q définit plusieurs catégories de liens. Le réseau fédérateur (*Trunk links*) comporte des liens qui relient les commutateurs et multiplexent les différents VLANs. Les trames qui le traversent sont étiquetées en vue d'identifier le VLAN auquel ces trames appartiennent. Par contre, les trames qui se trouvent sur les liens d'accès provenant ou destinés à des stations ne sont pas étiquetées. Chacun des liens d'accès est relié à un VLAN par son port ou par l'une des approches mentionnées précédemment. La figure 8.9 illustre ce fonctionnement.

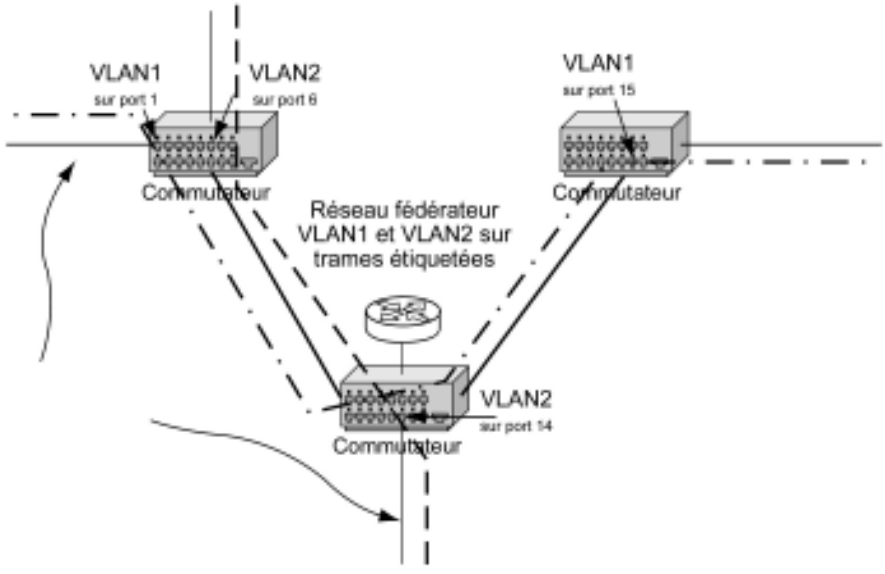


Figure 8.9 Réseau fédérateur et liens d'accès

Par ailleurs, il existe également des protocoles VLANs propriétaires. Cisco a conçu son propre protocole pour les réseaux fédérateurs : le protocole ISL (*Inter Switch Link*). Ce protocole est utilisé avec les liens de Fast Ethernet. Tel qu'illustré à la figure 8.10, les trames Ethernet sont encapsulées avec des en-têtes ISL avant de les mettre en circulation à travers le réseau fédérateur, entre commutateurs Cisco (*Catalyst*).

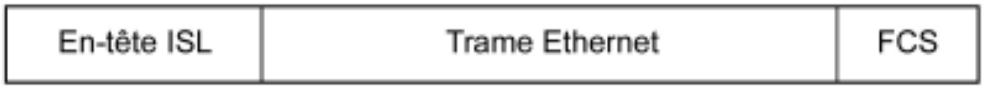


Figure 8.10 Encapsulation ISL

L'identifiant du VLAN figure dans l'en-tête en plus d'autres informations distinctes.

Il importe de noter que chacun des VLANs logiques configurés dans un commutateur agit comme un pont physique indépendant. La détection (*Learning*) de la position du terminal source est faite pour chacun des VLANs et de manière indépendante les uns par rapport aux autres. L'intra VLAN ou le passage de trames d'un VLAN à l'autre se fait aussi conformément à cette détection de position.

Chacun des VLANs et chacun des commutateurs et ports qui sont associés à ces VLAN établissent leur propre arborescence en faisant appel au Spanning Tree. Des mécanismes sont générés pour éviter qu'il y ait des boucles et, ils sont appliqués à chacun des VLANs pris individuellement.

Les VLANs dans toutes leurs complexités offrent une très grande flexibilité et sécurité dans les réseaux locaux. Nous avons vu dans le chapitre précédent qu'un réseau local qui comprend des boucles va avoir recours au Spanning Tree. Puisque le VLAN est un réseau local, il pourrait définir un Spanning Tree différent d'un autre VLAN.

Aujourd'hui, la plupart des équipements de commutation utilisés dans les réseaux locaux permettent de créer des VLAN et d'établir le Spanning Tree.

8.3 Exercices

8.1

Comment un routeur peut-il distinguer une trame Ethernet d'une trame VLAN IEEE 802.1Q?

8.2

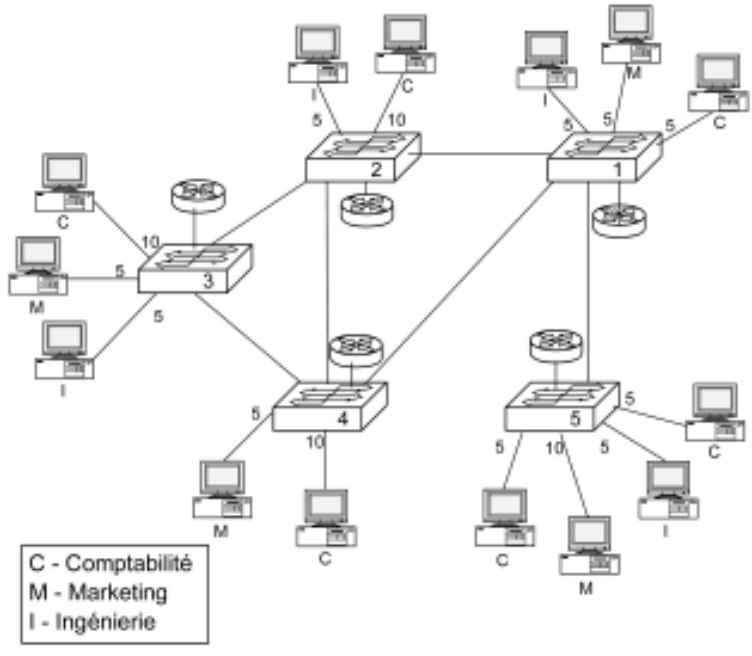
Comment le VLAN peut-il protéger les stations d'intrusions indésirables?

8.3

Dans un VLAN avec configuration de ports, comment l'intrusion indésirable peut-elle être effectuée?

8.4

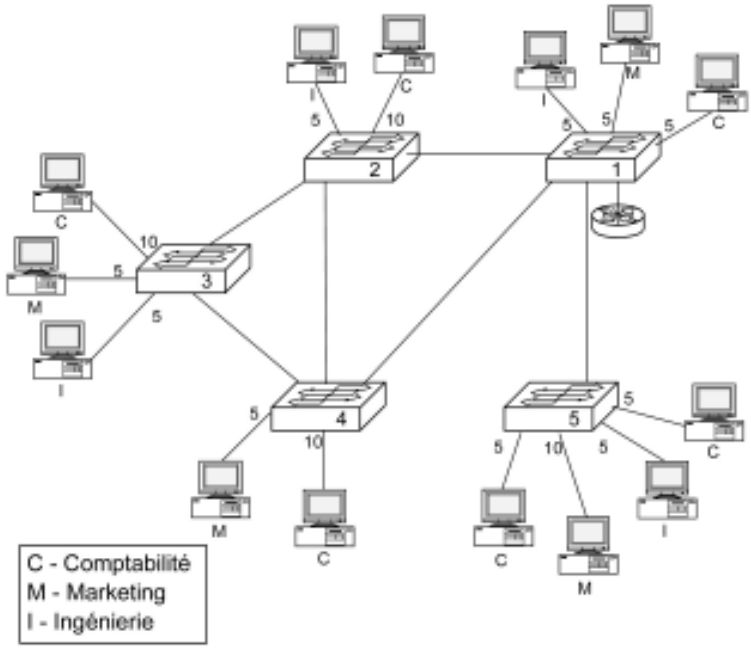
Le réseau d'une organisation comprend cinq commutateurs. Des terminaux de différents départements sont connectés à chacun de ces commutateurs. Trois VLAN doivent être définis, l'un pour le département de comptabilité, l'autre pour le marketing et le troisième pour le département d'ingénierie. Tous les liens sont à 100 Mbps. Les valeurs associées aux ports d'accès des terminaux sont les coûts pertinents au calcul du Spanning Tree.



Les VLANs sont élaborés par configuration des ports du commutateur. Définissez les VLANs et calculez le Spanning Tree pour chaque VLAN.

8.5

Indiquez comment les VLANs des différents départements pourraient se déployer et s'interconnecter dans le réseau suivant.



9 Le protocole Internet IP

9.1 Introduction

Le premier réseau de transmission de données couvrant de très grandes distances a été développé en 1969, pour l'agence américaine ARPA (*Advanced Research Project Agency*), aux États-Unis d'Amérique. Ce premier réseau a été nommé ARPANET et le protocole utilisé était alors le NCP (*Network Control Protocol*/Programme de gestion de réseau). En 1980, il prit le nom de protocole TCP/IP (*Transmission Control Protocol*/*Internet Protoco*l). ARPANET a été créé à cause de « la guerre froide ». Le département de la défense américaine (DoD) voulait en toute urgence créer des réseaux de commandes et contrôles militaires qui pourraient pallier toute attaque nucléaire. Les communications et transmissions de données ne devaient en aucun cas cesser d'être transmises ou reçues à cause de bris de liens. La transmission devait continuer de passer par le biais d'une autre route.

La technologie utilisée fut la commutation par paquets. Les paquets sont acheminés au sein même du réseau jusqu'au point de destination final. Les équipements qui procèdent à l'acheminement des paquets et dédiés à cette tâche se nomment des routeurs. Ils forment des nœuds dans le réseau. Les routeurs d'ARPANET sont connus sous le nom de PSN (*Packet Switch Node*/Nœud de commutation par paquets). Au début, ils étaient connus sous le nom de IMP (*Interface Message Processor*/Processeur de traitement de messages d'interface). Internet s'inscrit dans l'évolution logique d'ARPANET.

9.2 Modèles et protocoles

Les protocoles de la couche liaison ont été traités sous plusieurs aspects dans les chapitres précédents. Ce chapitre nous introduit à la couche réseau et plus particulièrement celle du modèle TCP/IP contenant le protocole IP (*Internet Protocol*).

Les spécifications pour les protocoles d'Internet sont écrites par le groupe de travail de l'IETF (*Internet Engineering Task Force*) lui-même sous surveillance du IAB (*Internet Architecture Board*). Les protocoles d'Internet sont décrits dans les documents RFC (*Request for Comment*). Ces protocoles, à l'instar de ceux observés par l'OSI, sont conformes à l'architecture en couches. Toutefois, les couches des protocoles Internet ne sont pas toutes identiques à celles de l'OSI. Par exemple et contrairement à l'OSI, les protocoles TELNET ou FTP comportent dans la couche supérieure leurs fonctions de présentation.

On peut comparer les deux modèles dans la figure 9.1 et remarquer les couches communes.

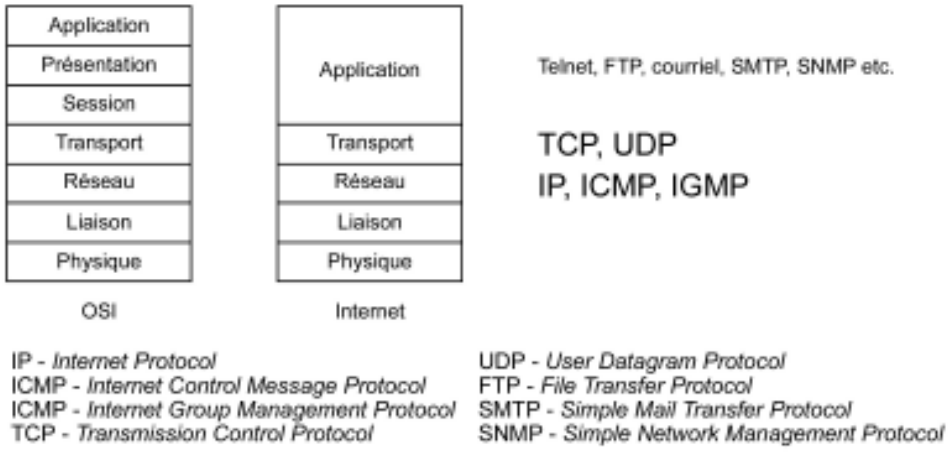


Figure 9.1 Comparaison des protocoles entre OSI et Internet

Plusieurs applications Internet sont disponibles. Nous retrouvons par exemple les protocoles FTP (*File Transfer Protocol*), SMTP (*Simple Mail Transfer Protocol*), SNMP (*Simple Network Management Protocol*) etc.

Le multiplexage et le démultiplexage de protocoles permettent aux différents services d'application de partager l'utilisation d'un protocole commun à la couche inférieure.

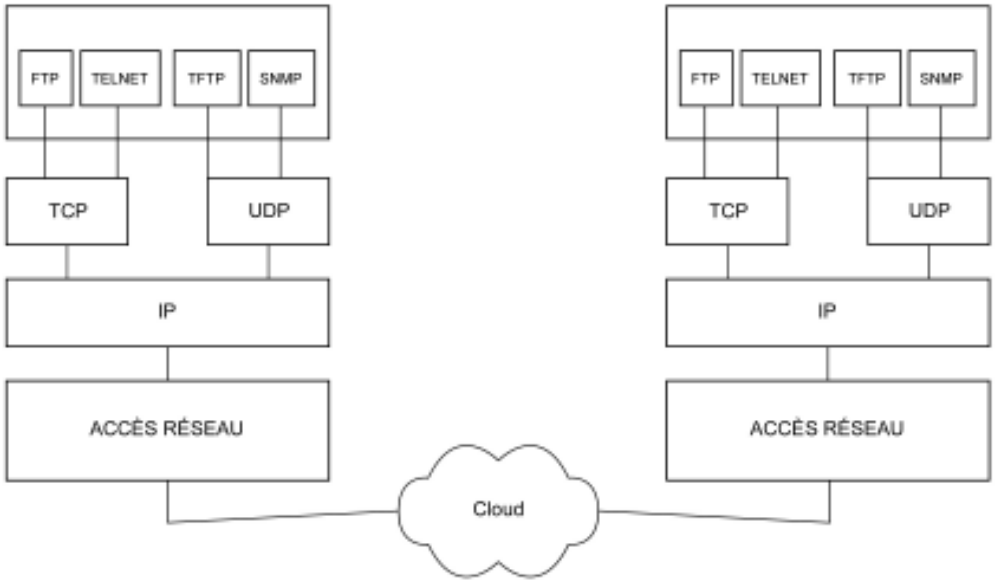


Figure 9.2 Représentation d'une pile de protocoles

La couche application d'un hôte pourrait avoir deux services activés et ceux-ci pourraient être partagés par la même couche transport. Les services FTP et TELNET de la figure 9.2 utilisent la même couche TCP. Ceci survient grâce au multiplexage. Par ailleurs, nous voyons que les services SNMP et TFTP (*Trivial File Transfer Protocol*) se partagent l'utilisation du même UDP.

En ce qui a trait à Internet, les protocoles et les interactions de chaque couche, depuis la couche réseau jusqu'à la couche application, sont définis dans un environnement commun. La pile de protocoles est connue sous le nom de « *Suite TCP/IP* » ou « *Protocol Suite* ». Cette « suite » de protocoles comprend bien plus que les protocoles TCP et IP.

La figure 9.3 nous donne un aperçu de ce que l'on désigne par suite TCP/IP.

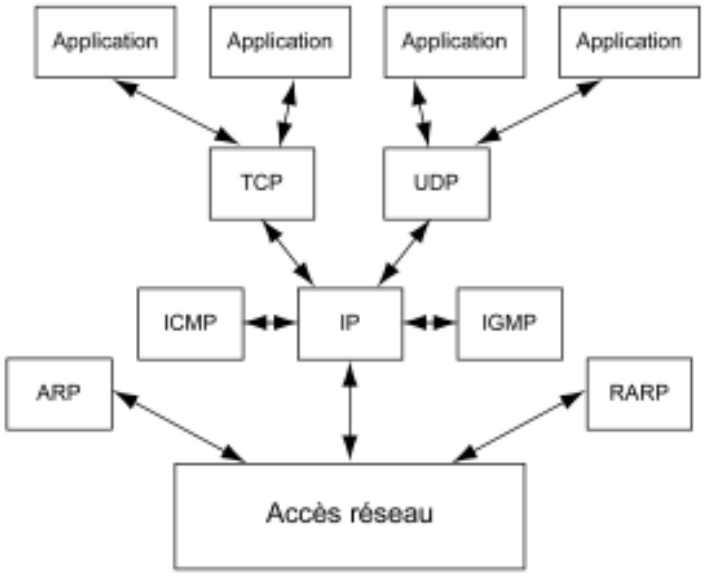


Figure 9.3 Les protocoles d'Internet et leurs relations

Les protocoles compris dans cette suite TCP/IP sont les suivants :

Le protocole IP (*Internet Protocol*) est un protocole de la couche réseau défini dans le RFC 791. C'est un protocole sans connexion. Les unités de données de ce protocole sont désignées par le terme « datagramme ». Le protocole IP permet leur acheminement de la source à la destination à travers un réseau dont les couches réseau sont uniformes.

Le protocole ARP (*Address Resolution Protocol*/Protocole de résolution d'adresse) est exploité pour trouver une adresse MAC à partir d'une adresse IP. Le ARP est formulé dans le RFC 826.

Le protocole RARP (*Reverse Address Resolution Protocol*/Protocole de résolution inverse d'adresse) est l'inverse ARP. Avec ce protocole, il nous est donné d'obtenir l'adresse IP à partir d'une adresse MAC. Ce protocole est utilisé par des stations mobiles pour obtenir leurs adresses IP. Ces stations transmettent leur adresse MAC au serveur pour lui permettre d'y associer les adresses IP. RARP est défini dans le RFC 903.

Le protocole ICMP (*Internet Control Message Protocol*) est utilisé pour répondre au besoin d'interaction qui n'existe pas dans l'environnement sans connexion du protocole IP. C'est par le ICMP que les erreurs de datagrammes ou de réseaux sont signalées. La définition du ICMP figure dans le RFC 792. Pour sa part, le

protocole IGMP (*Internet Group Message Protocol*) figure dans le RFC 2236. Il permet de gérer les groupes de multidistribution (*Multicast*).

Le Protocole TCP (*Transmission Control Protocol*) est un protocole qui appartient à la couche transport. Ce protocole, défini dans le RFC 793, est orienté connexion. Ses unités de service sont désignées par le terme « segment ». En outre, les services d'applications ont recours à lui en raison de sa fiabilité. Par contre, ce protocole n'est pas approprié pour des échanges interactifs ne tolérant pas des délais, comme dans le cas du transport de la voix sur IP.

Le protocole UDP (*User Datagram Protocol*) est aussi un protocole de la couche transport. Cependant, il n'est pas orienté connexion et ne garantit pas la livraison des segments. On peut trouver une description plus élaborée dans le RFC 768.

9.2.1 Acheminement des datagrammes dans le réseau

Tous datagrammes voyageant dans Internet vont fort probablement traverser plusieurs réseaux. L'interconnexion de deux réseaux différents se fait à travers des routeurs (cf. figure 9.4). Le routeur utilisé dans l'interfonctionnement possède les fonctionnalités suivantes :

- assurer le lien entre des réseaux,
- assumer le routage et la livraison des données entre des terminaux reliés à des réseaux différents,
- assumer ces fonctionnalités de façon à ne pas devoir recourir aux modifications d'architecture réseau.

Le routeur doit pour cela prendre en considération les différences entre réseaux, notamment :

- le schéma de l'adressage,
- la longueur maximale du paquet - *requis pour la segmentation*,
- les interfaces en présence et
- le seuil de fiabilité.

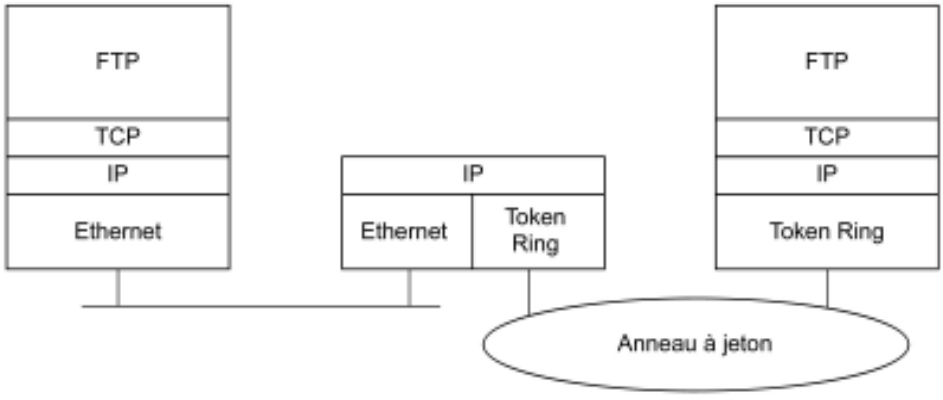


Figure 9.4 Représentation graphique d'un schéma inter réseau

Puisque l'interfonctionnement est établi au niveau IP, il va de soi que tous les protocoles, situés de la couche réseau à la couche application, devront être compatibles.

9.2.2 Encapsulation des messages

Le protocole IP encapsule les messages de la couche transport (TCP ou UDP) dans des datagrammes. Cela revient à ajouter aux segments TCP ou UDP un en-tête propre au protocole IP (figure 9.5). L'en-tête comprend plusieurs informations qui vont permettre aux routeurs d'acheminer le datagramme vers sa destination. Chaque routeur procédera à l'analyse de ce datagramme par sa couche IP. Ses décisions seront fondées sur l'information contenue dans l'en-tête IP.

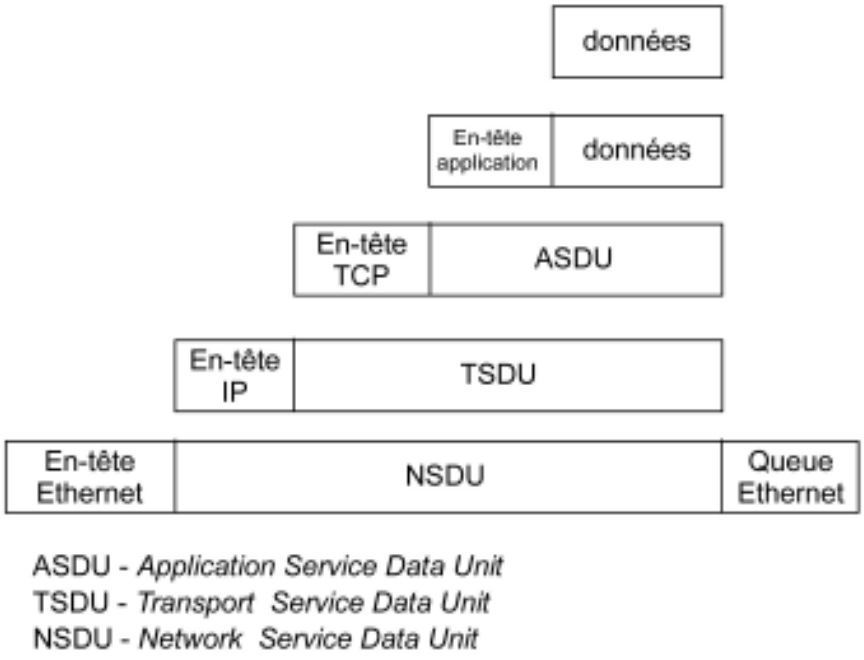


Figure 9.5 Représentation de l'encapsulation d'un datagramme IP

Comme nous en avons déjà fait mention au chapitre 2, l'encapsulation survient à tous les niveaux. Celle qui est faite à la couche 2 par Ethernet permet de savoir à quel protocole recourir à la couche supérieure sur réception de la trame. Nous pouvons remarquer qu'aux figures 9.6 et 9.7 le champ « type » est réservé permettant d'indiquer le type de protocole de la couche réseau.

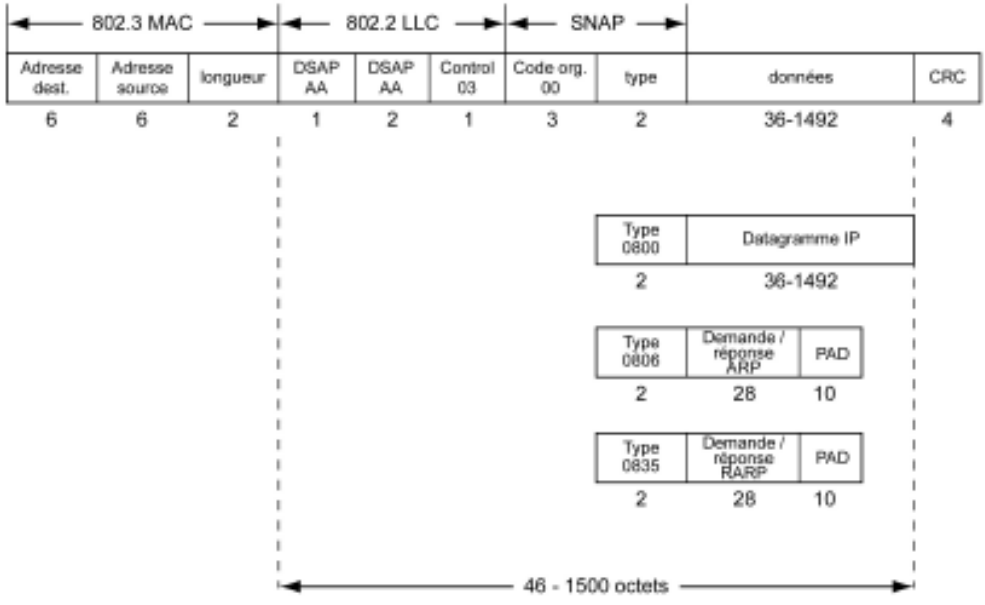


Figure 9.6 Schéma de l'encapsulation pour CSMA/CD SNAP (RFC 1042)

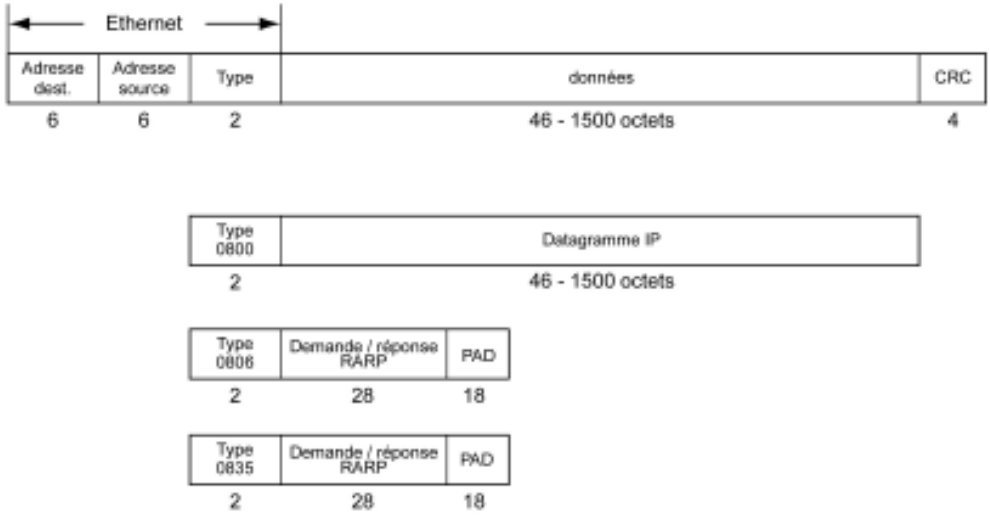


Figure 9.7 Schéma de l'encapsulation pour Ethernet (RFC 894)

9.3 Le protocole IPv4

IP est un protocole orienté sans connexion (*connectionless*). Il ne comporte aucun système de retransmission d'erreurs. Les routeurs ou hôtes qui reçoivent un datagramme IP défectueux transmettent en retour un message ICMP au terminal source. La fiabilité est assurée par le protocole de la couche supérieure TCP, dans le cas ou il est utilisé.

Le protocole IP comprend des PDUs avec un en-tête décrit à la figure 9.8. Cet en-tête procède au relais de toute l'information requise pour le traitement des données par le terminal de réception.

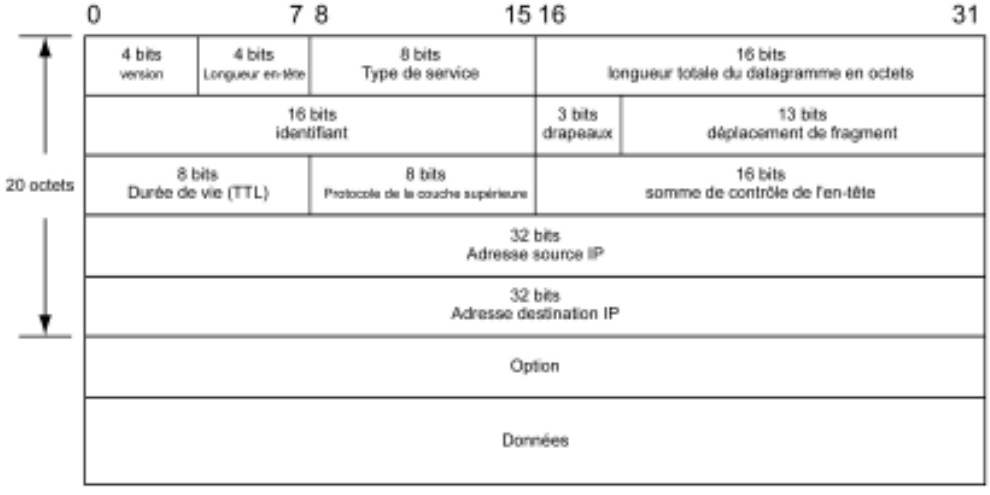


Figure 9.8 Représentation du datagramme selon le protocole IP version 4

La séquence de lecture du datagramme est : premièrement la tranche des bits 0 à 7, deuxièmement celle des bits 8 à 15, puis celle des bits 16 à 23 et, enfin, celle des bits 24 à 31. La version courante du protocole IP est 4 (IPv4).

Les champs de l'en-tête pour la version 4 du protocole IP sont les suivants :

- « Version » – indique la version IP utilisée dans ce datagramme. Dans notre cas il s'agit de la version 4.

- « Longueur en-tête » (*hlen – header length*) : indique la longueur de l'en-tête IP en nombre de mots de 32 bits.

- « Type de service TdS » – Indique le type de service et la priorité du datagramme (défaut = 0). Ce champ comprend huit bits tel qu'illustré dans la figure 9.9.

Figure 9.9 Champ Type de service

La définition initiale de ce champ selon le RFC 791 comprend deux parties :

- la précédence IP, longue de 3 bits, utilisée pour indiquer la priorité ou importance du datagramme,

- et quatre bits spécifiques : délai, débit, fiabilité, coût où chacun des bits pourrait être utilisé pour souligner qu'il y a un TdS propre à une application. Les quatre bits (cf. tableau 9.1) sont exploités pour minimiser le délai, maximiser le débit, maximiser la fiabilité et minimiser les coûts. Pour une application donnée, seulement un de ces quatre bits peut être mis à 1. Si les quatre bits sont à 0, cela indique que le service est normal.

Tableau 9.1 Type de Service (RFC 1340)

Application	Minimise le délai	Maximise le débit	Fiabilité	Coût	Valeur hexadécimale
Telnet / Rlogin	1	0	0	0	0x10
FTP					
contrôle	1	0	0	0	0x10
donnée	0	1	0	0	0x08
SMTP					
phase de commande	1	0	0	0	0x10
Phase donnée	0	1	0	0	0x08

Une modification récente (RFC 2474, 3168, 3260) de ce champ a été apportée pour introduire, dans la transmission des datagrammes, la qualité du service. La méthode utilisée se nomme services différenciés ou *DiffServ* (*Differentiated Services.*)

- « Longueur du datagramme » : indique la longueur du paquet IP en octets.

Dans le but de contrôler la fragmentation et le réassemblage, les champs suivants sont sollicités : Identifiant (*fragid- Fragment ID*), Fanions (*flags),* Déplacement de fragments, (*fragoff – FragmentOffset*).

- « Identifiant » (*fragid*) : (RFC 791) Ce champ identifie de façon unique chaque datagramme transmis par un terminal hôte. Le numéro assigné à ce champ est le même pour tous les datagrammes qui font partie de la même fragmentation. De la sorte, si un NSDU doit être fragmenté, des datagrammes avec le même *fragid* vont être créés. Ce numéro indique au récepteur que les datagrammes avec le même *fragid* appartiennent au même NSDU.

- « Drapeaux » (*flags*) : Ce champ comprend trois bits. Un bit, le drapeau MF (*more fragment flag)*, est utilisé pour indiquer, s'il est mis à un, qu'il y a un prochain datagramme avec le même *fragid* appartenant au même NSDU. Si ce bit est à 0, ceci indique que c'est le dernier (ou le seul) datagramme du NSDU. Il y a un autre bit nommé *don't fragment*. Ce bit demande qu'il n'y ait pas de fragmentation. Si ceci n'est pas possible, une erreur est indiquée avec un ICMP. Le troisième bit est 0 et n'est pas utilisé.

- « Déplacement de fragment » (*Fragoff*) : Ce champ indique le déplacement du contenu de ce datagramme en multiple de huit octets par rapport au début du NSDU. Un exemple est donné à la section 9.5.

- « Durée de vie – TTL » (*time to live*) : Il s'agit du temps pendant lequel le paquet peut exister. Le contenu de ce champ décroît de 1 chaque fois qu'il passe par un routeur ou hôte. Lorsque le TTL atteint 0, le datagramme est détruit et le transmetteur en est avisé par un message ICMP. Le TTL impose une limite au nombre de routeurs visités par le datagramme. Le transmetteur initialise le TTL (typiquement 128) avant de transmettre le datagramme. De cette façon, il lui évite la possibilité de boucler à l'infini.
- « Protocole de la couche supérieure » : Indication du protocole utilisé au dessus du IP. Par exemple, 1 indique ICMP, 6 indique TCP et 17 indique UDP.

- « Somme de contrôle de l'en-tête » (*checksum*) : Vérification d'erreur de l'en-tête du datagramme IP.

- « Source » : Indique l'adresse IP de la source.

- « Destination » : Indique l'adresse IP de la destination.

- « Options » : choix possible
  - *none* - aucun
  - *routage source* - route du paquet préétabli explicitement
  - *enregistrement de route* – route enregistrée dans le champ option
  - *estampille* – indique l'heure (en millisecondes) à chacun des noeuds
  - *sécurité* - option de sécurité rarement utilisée (RFC 1108)
  - *bourrage (padding)* - indique que le remplissage suit

Le bourrage (*padding*) est nécessaire pour que tous les paquets soient des multiples de 32 bits. Au besoin, on doit ajouter des bits de bourrage à la fin de l'en-tête IP si le datagramme n'est pas un multiple de 32 bits.

9.4 Unité de transfert maximum

Tous les réseaux n'acceptent pas la même longueur de trames. À ce sujet, il se peut qu'il y ait une contrainte de longueur pour les trames de la couche liaison. Cette limite de longueur se nomme le MTU (*Maximum Transfer Unit*/Unité de transfert maximum). Si le datagramme à transmettre est plus long que le MTU de la couche liaison, la couche réseau doit procéder à la fragmentation des NSDU (*Network Service Data Unit.*)

Le tableau 9.2 donne quelques exemples de réseaux avec leur MTU.

Tableau 9.2 Exemple de réseaux et de MTU correspondants

Réseaux	MTU (octets)
Hyperchannel	65535
16 Mbps token ring (IBM)	17914
4 Mbps token ring (IEEE802.5)	4464
FDDI	4352
Ethernet	1500
IEEE 802.3/802.2	1492
X.25	576
Point to point (low delay)	296

Lorsque deux hôtes communiquent à travers différents réseaux, chaque lien peut avoir un MTU distinct. La longueur la plus importante repose sur le plus petit

MTU de toute liaison que les paquets traversent entre deux terminaux hôtes. Cette valeur est connue sous le nom de chemin (*path*) du MTU.

Le chemin du MTU n'a pas besoin d'être une constante puisque la route prise entre les deux hôtes peut changer.

Le système de découverte du chemin du MTU à l'aide du ICMP (RFC 1191) permet d'ajuster la fragmentation des NSDU.

9.5 Fragmentation avec le protocole IP

La fragmentation s'impose lorsque certains réseaux ne peuvent pas assumer une certaine longueur de datagrammes. Lorsque le NSDU est trop long, il ne peut être transmis comme tel. Par conséquent, il doit être segmenté. De plus, la segmentation doit aussi prendre en considération l'espace requis par l'en-tête. Les champs impliqués dans la segmentation sont : *fragid, total length, fragoff* et *more bit*.

Les routeurs qui reçoivent des datagrammes doivent savoir, d'après l'adresse destination, quel chemin prendre et donc quel port de sortie emprunter pour les acheminer vers leur destination. Ce port de sortie est connecté à un autre réseau qui n'accepte qu'une certaine longueur de datagramme. Le routeur doit donc fragmenter les datagrammes pour que ceux-ci puissent être transmis à travers le réseau.

Les informations des couches supérieures sont celles qui subissent la fragmentation. En effet, les données comprises dans le datagramme émanent des couches supérieures. Cette fragmentation se fait par le service réseau. Une fois parvenues à la couche réseau (IP), les informations des couches supérieures sont considérées comme des NSDU. Ceux-ci sont aussi connus sous le nom de *NS-User Request Data Unit* dont la longueur peut atteindre 64000 octets.

Les fonctions de fragmentation et de réassemblage sont effectuées sur les NSDU. Les fragments des données contenus dans chaque datagramme IP reçu, et en relation avec le même NSDU, sont rassemblés pour reconstituer le NSDU. Ce n'est qu'une fois le NSDU reconstitué qu'il est transmis au *NS-user,* c'est-à-dire à la couche transport du point de destination.

La fragmentation génère certaines surcharges de travail. Le routeur, qui découpe le datagramme, doit procéder à un nouveau calcul de l'en-tête pour chaque fragment. De plus, puisque le NSDU doit être reconstitué à partir de tous les

fragments au moment de sa réception, si un seul fragment se perd tout le NSDU est détruit.

La perte de fragments occasionne la transmission inutile d'un datagramme. En effet, tous les datagrammes porteurs d'autres fragments sont transmis parce que la perte d'un datagramme n'est réalisée qu'au moment du réassemblage. Le réassemblage n'est fait qu'à la destination, car les datagrammes peuvent emprunter différentes voies dans un réseau sans connexion.

Lorsqu'un NSDU doit être fragmenté, il faut tout d'abord connaître le MTU du réseau. Il faut aussi connaître la longueur de l'en-tête des datagrammes qui vont être créés. Si le datagramme n'utilise pas le champ option, la longueur de l'en-tête est de 20 octets. Afin de déterminer la grandeur des fragments de données, il faut soustraire l'en-tête de la longueur maximale du datagramme (MTU au maximum). Cependant, le reste doit être un multiple de 8 octets. Si tel n'est pas le cas, on doit trouver la longueur qui est un multiple de 8 et qui se rapproche le plus de la valeur calculée sans la dépasser.

« Longueur maximale du datagramme » = [MTU en octet – en-tête en octet]

« Longueur actuelle du datagramme » = 8 x N <= « Longueur maximale du datagramme »

Nous avons maintenant les éléments nécessaires pour préparer nos datagrammes et procéder à la transmission des fragments. La valeur utile est N. Elle sera utilisée lors des calculs du déplacement des fragments.

Étant donné que la communication IP est sans connexion, il n'existe pas de chemin virtuel et les datagrammes transmis n'emprunteront pas forcément le même chemin. Si l'un des datagrammes accuse un retard, le datagramme suivant pourrait bien emprunter un autre chemin et arriver avant lui. Il est donc important de fournir la position des données comprises dans chaque datagramme par rapport au NSDU. Cette position est indiquée par un multiple de N. Le déplacement du premier segment du NSDU est de 0 puisqu'il est au début. Le deuxième segment a un déplacement de N. Le troisième segment a un déplacement de 2 N et ainsi de suite. Il importe de noter que tous les segments sauf le dernier auront la même longueur totale.

Il s'agit donc de donner à tous les segments qui appartiennent au même NSDU le même identificateur unique afin de le signaler au récepteur pour que ce dernier puisse les rassembler. L'identifiant (*fragid*) permet de donner une telle indication. De plus, dans chacun des datagrammes, il ne faut pas omettre de mentionner si

un autre datagramme comportant le segment suivant du NSDU est attendu. Ceci se fait grâce au drapeau MF. Le déplacement du segment ainsi transporté par le datagramme doit être signalé en rapport avec le début de la transmission du NSDU. C'est en cela que N se révèle utile. Le déplacement de fragment (*fragoff*) est cette indication et sa valeur correspond à un multiple de N. La longueur totale de chaque datagramme est signe que le MTU n'a pas été dépassé.

Nous allons illustrer les concepts qui précèdent par un exemple.

Un NSDU de 2000 octets doit être transmis sur un réseau exploitant le protocole IP, et définissant le MTU à 512 octets de longueur. Si l'en-tête IP utilisée se trouve à sa taille minimum (sans option), il s'agit de trouver les valeurs des champs suivants pour chaque datagramme transmis :

a) Identifiant
b) Longueur totale (*Total length*)
c) Déplacement de fragment (*Fragment offset*)
d) Drapeau MF (*More Fragment Flag*)

Réponse :

La longueur minimale de l'en-tête IP est de 20 octets.
La longueur maximale du datagramme est = 512-20 = 492 octets.
Le fragment doit être un multiple de 8 octets sauf pour le dernier datagramme.
Ainsi, le nombre approchant 492 en termes de multiple de 8 est égal à 8 x 61, soit 488 octets.
Nous constatons alors que 61 est la valeur assignée à N.
Il nous est également possible de constater qu'il y a 5 datagrammes à transmettre (2000 / 488 = 4.098).

De ce fait, les datagrammes transmis comprennent les champs suivants :

Datagramme	**1**	**2**	**3**	**4**	**5**
Identifiant	10	10	10	10	10
Longueur totale	508	508	508	508	68
Déplacement de fragment	0	61	122	183	244
Drapeau MF	1	1	1	1	0

Notons que l'identifiant a été choisi arbitrairement. Le point important que nous voulons montrer est que l'identifiant est le même pour tous les fragments.

Chacun des quatre premiers datagrammes compte 508 octets. Ceci représente 488 octets de NSDU plus 20 octets de l'en-tête. Les quatre premiers datagrammes transmettent 1952 octets de NSDU. Il reste donc 48 octets de NSDU transmis par le dernier datagramme (longueur totale du dernier datagramme = 48 octets NSDU + 20 octets en-tête). Le drapeau MF indique durant les quatre premiers datagrammes qu'un prochain datagramme est attendu sauf naturellement pour le dernier.

9.6 Adresse IPv4

Le protocole IP version 4, ou IPv4, est la version courante du protocole Internet. Comme nous le verrons plus loin dans ce chapitre, une nouvelle version, la version 6, a été approuvée. Elle a principalement été développée en raison des problèmes de la limite du nombre d'adresses qu'offre le IPv4.

Dans cette partie, nous allons couvrir exclusivement l'adressage IPv4. Nous verrons aussi que de nouvelles méthodes sont mises en vigueur dans la façon de se servir de l'adresse IPv4, ce qui retarde le besoin de recourir au format d'adresse IPv6.

L'adresse IP se compose de 4 octets ou 32 bits. Elle devrait permettre en principe d'envoyer une adresse à 4 294 967 296 (2^{32}) hôtes. En raison de certaines règles qui prévalent, le nombre d'adresses utilisées sera plus limité.

Pour permettre un regroupement d'adresses, l'adresse IPv4 est représentée en notation *dotted-decimal***.** Cela signifie que chacun des octets est codé conformément au système décimal et est représenté individuellement et séparément l'un de l'autre par un point. Par exemple, **128.83.12.14** représentant 4 octets en *dotted-decimal* équivaut à 2 152 926 222 dans le système décimal global.

Il est plus pratique d'assigner des adresses en *dotted-decimal* plutôt qu'en valeur globale, puisque dans ce cas, la gestion des adresses aurait été impossible. Les adresses en *dotted-decimal* peuvent être assignées en groupe commun ou en adresse de réseau.

Pour simplifier le routage, l'adresse IP est divisée en numéro de réseau et en numéro hôte. Toutes les stations du même réseau se voient allouer le même numéro de réseau. Par exemple, les deux hôtes suivants **128.81.**75.63 et

128.81.10.1 ont le même numéro de réseau et transitent tous deux sur le même réseau.

9.6.1 Classes d'adresses

Le codage d'adresse est structuré de façon à avoir cinq classes d'adresses, répertoriées de A à E (cf. Tableau 9.3). Chacune de ces classes réserve une partie des d'octets pour l'assignation du réseau, et l'autre partie pour désigner les hôtes. Comme règle à suivre, l'adressage IPv4 alloue la répartition du tableau 9.3. Cette dernière dépend des quatre bits les plus significatifs du premier octet de l'adresse.

Tableau 9.3 Détermination des classes

Classes	Bits d'ordre supérieur	Adresse réseau plus bas	Adresse réseau plus haut	Hôte plus bas	Hôte plus haut
	0 1 2 3				
A	0 - - -	**0**.0.0.0	**127**.0.0.0	0.0.0.0	255.255.255
B	1 0 - -	**128.0**.0.0	**191.255**.0.0	0.0.0.0	255.255
C	1 1 0 -	**192.0.0**.0	**223.255.255**.0	0.0.0.0	255
D	1 1 1 0	**224.0.0.0**	**239.255.255.255**	N.A.	N.A.
E	1 1 1 1	**240.0.0.0**	**255.255.255.255**	N.A.	N.A.

Les réseaux de classe A codent leur adresse réseau sur huit bits. Étant donné que pour désigner la classe A, le premier bit doit être zéro (bit 0), il ne reste que 128 adresses. De ces dernières, il faut enlever l'adresse 0.0.0.0 et 127.0.0.0 (voir adresses réservées plus bas). La classe d'adresses A ne compte donc qu'un total de 126 réseaux.

Chacun des réseaux de la classe A exploite 24 bits pour l'adresse hôte. De ce fait, il peut adresser 16 777 214 hôtes (16 777 216 – 2). On enlève l'adresse hôte 0 et son adresse de diffusion du nombre d'adresses hôtes disponibles. L'adresse hôte 0 indique que l'hôte source s'adresse à lui-même, alors que l'adresse diffusion destine le fragment à tous les hôtes du sous réseau.

Les réseaux de la classe B exploitent 16 bits pour adresser leurs réseaux. Par contre, puisque les deux premiers bits du premier octet doivent être 1 0 binaire, les adresses réseaux possibles ne peuvent aller que de 128.0 à 191.255. Ainsi, la classe B dispose de 16 384 (2^{14}) adresses réseaux possibles.

Il reste 16 bits pour les adresses des hôtes de la classe B. Chaque réseau de la classe B peut donc adresser 65 534 hôtes (65 536 – 2).

Pour sa part, la classe C exploite 24 bits pour l'adressage réseau. Les trois premiers bits du premier octet doivent être 110 en mode binaire pour désigner la classe C. Ceci ne nous laisse donc que 21 bits de disponibles pour coder les adresses des réseaux de la classe C. Nous pouvons alors avoir 2 097 152 (2^{21}) réseaux appartenant à la classe C. Huit bits sont réservés à l'adressage des hôtes de la classe C. Ainsi, nous avons 254 (256 – 2) hôtes par réseau de classe C.

La classe D inclut des adresses uniformes sans hôte et est exploitée pour définir les adresses en matière de multidistribution (*Multicast*). Les adresses de cette classe sont des adresses de groupes de terminaux. Des machines appartenant à un groupe particulier sont de ce fait associées à la même adresse de classe D et elles pourront recevoir simultanément la même communication, et ce, peu importe leur emplacement dans le réseau.

La classe E est réservée à un usage futur.

Nous pouvons remarquer que 50% des adresses IP sont des adresses de classe A puisque cette dernière utilise 31 bits sur les 32 disponibles pour ses adresses IP individuelles. La classe B exploite 25% des adresses disponibles. Une adresse de classe B code 30 bits pour chacun de ses hôtes. La classe C, quant à elle, ne couvre que 12,5% d'adresses possibles (29 bits codés par adresse). Le reste, c'est-à-dire 12,5%, s'en trouve partagé par les classes D et E.

9.6.2 Assignation d'adresses

Peu importe la classe d'adresses, tous les hôtes d'un même réseau doivent avoir le même numéro de réseau et un numéro unique d'hôte. Deux réseaux peuvent être interconnectés par un routeur, chaque réseau aura sa propre adresse. Le routeur aura par contre l'adresse d'un réseau sur un port et l'adresse de l'autre réseau sur l'autre port. La figure 9.10 nous donne un exemple de deux réseaux connectés par un routeur et indique les adresses de chaque élément.

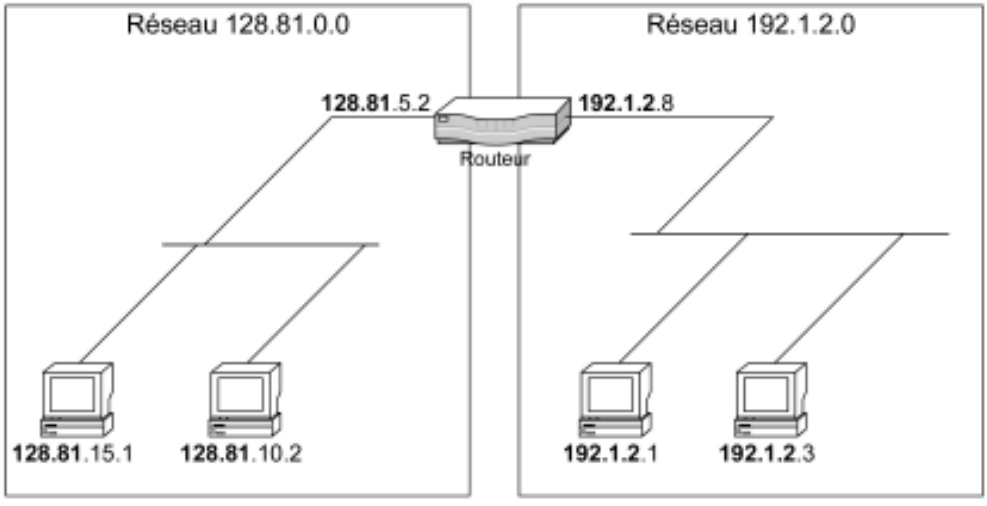


Figure 9.10 Exemple d'assignation d'adresses

9.6.3 Adresses réservées

Quelques adresses IP sont réservées. Si les octets désignant le réseau sont à 0, alors il s'agit d'un hôte du même réseau. Si tous les octets de l'adresse sont 0 (réseau et hôte), il s'agit dans ce cas du même hôte. Ces adresses ne sont utilisables que comme adresses de destination.

Il existe d'autres adresses réservées à des usages précis. L'adresse réseau 127.0.0.0 est une adresse de classe A et est considérée comme adresse locale. Tout datagramme ayant cette adresse ne sort pas du terminal parce que le protocole IP va le considérer comme adressé au terminal lui-même. Ce bouclage local est très utile pour tester l'interaction de processus du même hôte.

Lorsque les octets de l'adresse réservés à l'hôte sont 1 binaire partout (255 décimales) nous nous trouvons face à une adresse de diffusion (*Broadcast*). Par exemple, 128.81.255.255 est une adresse de diffusion sur le réseau 128.81.0.0. C'est dire que tous les hôtes du réseau 128.81.0.0 recevront le paquet.

Certains groupes d'adresses ont été réservés par l'IANA (*Internet Assigned Number Authority*), pour chacune des trois classes d'adresses, et exploités par des réseaux privés. Ces groupes d'adresses tels que définis dans le RFC 1597 sont :

- Classe A de 10.0.0.0 à 10.255.255.255
- Classe B de 172.16.0.0 à 176.31.255.255
- Classe C de 192.168.0.0 à 192.168.255.255

Nous remarquons que le groupe d'adresses réservées de la classe A n'offre qu'une seule adresse réseau numérotée 10.0.0.0. Cependant, cette adresse réseau peut correspondre à 2^{24} adresses hôtes, soit 16 777 216.

Les groupes d'adresses réservées de classe B offrent 16 réseaux commençant par l'adresse réseau 172.16.0.0 et se terminant par l'adresse réseau 172.31.0.0. Chacun de ces réseaux comprend 2^{16} adresses hôtes, soit 65 536.

À la classe C, les groupes d'adresses qui lui ont été réservées lui permettent d'allouer 256 réseaux privés. Ces adresses vont de l'adresse réseau 192.168.0.0 à l'adresse réseau 192.168.255.0. Chacun de ces réseaux ne contient par contre que 254 adresses hôtes.

Aucune adresse privée n'est accessible directement de l'Internet. Le moyen qui en permettrait l'accès est le recours au NAT (*Network Address Translation*) qui sera traité au chapitre 10. Le NAT permet de traduire les adresses internes du réseau privé en adresse externe reconnue par Internet. Un autre moyen d'avoir accès au réseau privé se fait par le biais d'un serveur proxy.

9.6.4 Adresse de sous-réseau

Une adresse de réseau est assignée aux organisations qui en font la demande. La plupart des besoins organisationnels sont comblés avec des adresses de classe C. Chaque adresse de classe C dispose d'un nombre d'adresses d'hôtes égal à $2^8 – 2$, soit 254 adresses uniques. Les organisations qui ont besoin de plus d'adresses se voient assigner soit une autre adresse réseau C soit une adresse réseau B. Une adresse de classe B offre un nombre d'adresses hôtes égal à $2^{16} – 2$, c'est-à-dire 65 534 adresses. Cette dernière option n'est pas la meilleure, si l'organisation à qui l'adresse a été assignée n'utilise qu'un nombre restreint, par exemple 1000 adresses. Dans ce cas, il restera, associé à cette adresse réseau, un nombre considérable d'adresses non utilisées (64 534). Ces adresses ne peuvent pas être assignées à un autre réseau si on se conforme à l'adressage par classe.

Les adresses IP sont de plus en plus demandées. Il n'y a que 32 bits dans une adresse. Il s'agit donc d'en tirer le maximum d'avantages et ceci est possible si on utilise l'approche de sous-adressage (*Subnet Addressing*). Il s'agit de ne plus se baser exclusivement sur les structures d'adresses réseaux et hôtes selon la classification définie auparavant. Un réseau utilisant l'adresse équivalente à la classe B ne sera pas nécessairement assigné 65 534 adresses hôtes. Cette plage d'adresse que représentait la classe B peut être partagée entre plusieurs sous-réseaux. Dans ce contexte, chaque échange de paquets doit prendre en

considération l'adresse IP et le masque (*Subnet Mask*) qui permettra de déterminer l'adresse réseau et l'adresse hôte.

Ainsi, si nous considérons l'adresse 150.214.209.2, selon la méthode d'adressage, ceci est une adresse de classe B dont l'adresse réseau est 150.214.0.0. Cependant, si un masque est défini par 255.255.255.0, la nouvelle adresse réseau, telle qu'illustrée à la figure 9.11, devient 150.214.209.0 et ce réseau ne peut avoir que 254 hôtes. On voit donc qu'à partir d'une adresse de classe B, on peut définir plusieurs adresses sous-réseau et donc étendre le champ d'adressage.

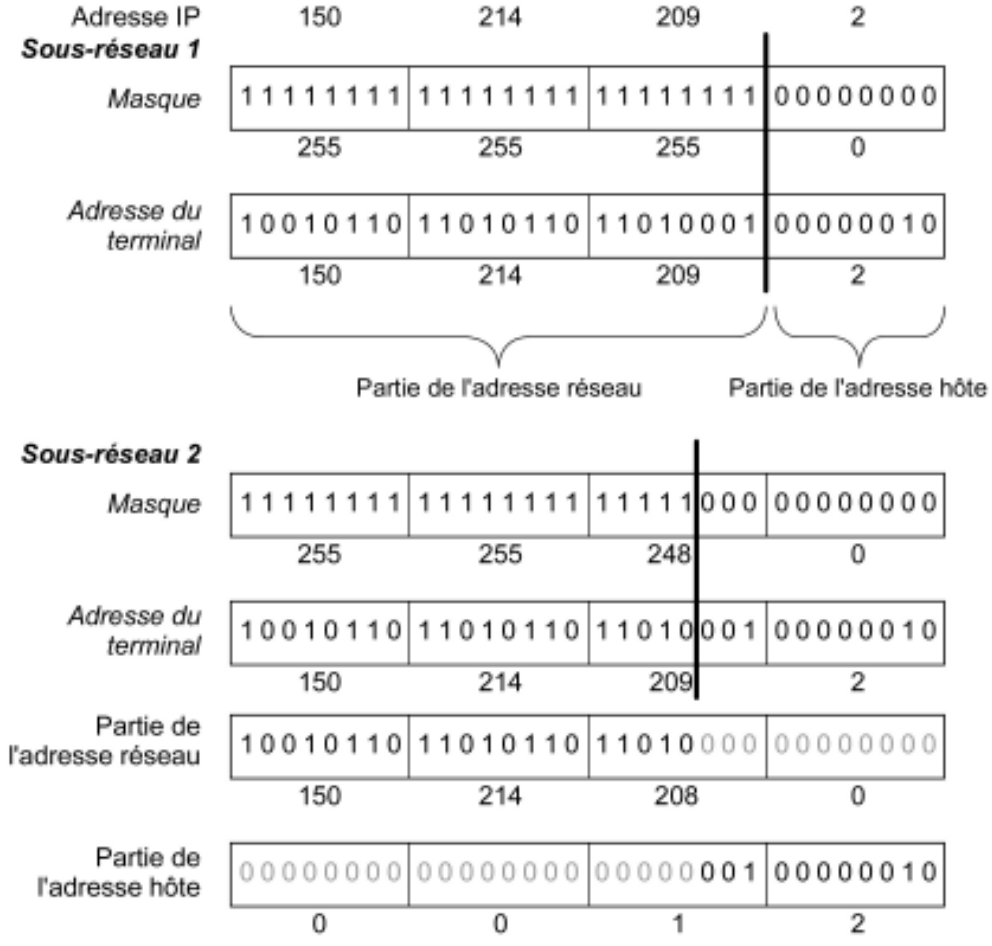


Figure 9.11 Représentation d'un masque de sous-réseau

Si le masque est 255.255.248.0 pour la même adresse IP, la partie de l'adresse réseau, illustrée aussi à la figure 9.11, devient 150.214.208.0/21 parce que seuls les 21 bits indiquant l'adresse réseau sont représentés. La partie de l'adresse hôte

de ce terminal est 0.0.1.2, c'est-à-dire les 11 bits les moins significatifs de l'adresse. Les adresses hôtes possibles sont au nombre de 1046 (2^{11} –2) pour ce nouveau sous-réseau.

Ces adresses de sous-réseau sont des adresses sans classe (*Classless addresses*). Il est impératif d'observer certaines conditions qui doivent être respectées pour utiliser ces adresses. Chacune des adresses du sous-réseau doit être accompagnée de son masque. Grâce à ce masque, les adresses de sous-réseau et d'hôte peuvent être établies. Le masque ne peut définir un sous-réseau qui serait en contradiction avec la classe de l'adresse. Par exemple, une adresse de classe B utilise les deux octets les plus significatifs pour identifier le réseau. Si cette adresse est utilisée pour définir des sous-réseaux, le masque ne peut pas avoir moins de deux octets de couverture réseau.

Si nous reprenons l'adresse de notre exemple, c'est-à-dire 150.214.209.2, le masque par défaut de la classe B est 255.255.0.0. On ne peut pas utiliser un masque 255.0.0.0 parce que ceci constituerait une violation du concept de l'adressage traditionnel (ceci change dans le cas de CIDR que nous verrons plus loin). Par conséquent, le concept d'adressage sans classe ne doit pas être en violation avec l'adressage traditionnel. Dans le cas de notre exemple, les sous-réseaux peuvent être établis en s'attribuant une partie des deux octets normalement réservés aux hôtes.

L'identification d'un sous-réseau est faite en spécifiant tant l'adresse que le masque. Ainsi, 150.214.209.0 255.255.255.0 est une adresse de sous-réseau. Une méthode de notation plus simple pour identifier l'adresse de ce sous-réseau est la suivante : 150.214.209.0/24. Le numéro 24 qui suit l'adresse identifie le nombre de 1 qui se trouve dans le masque.

L'adressage basé sur les classes n'utilise pas de masque. Ce dernier est implicite et fait partie de la définition des classes. Puisque le masque définit la partie de l'adresse réseau et que les classes sont organisées pour délimiter les adresses réseau, il est clair que, dans ce cas, le masque n'est pas nécessaire.

9.6.5 Établissement de sous-réseau

Considérons l'adresse privée 192.168.1.0/24 et établissons six sous-réseaux, le nombre d'hôtes par sous-réseau et l'adresse de diffusion de chacun de ces sous-réseaux.

Une bonne façon de procéder est de représenter en premier l'adresse et le masque sous leur forme binaire :

192	168	1	0
11000000	10101000	00000001	00000000

255	255	255	0
11111111	11111111	11111111	00000000

Les sous-réseaux ne peuvent être créés qu'à partir des huit derniers bits parce que nous avons affaire à une adresse de classe C. Trois bits sont nécessaires pour assigner les six sous-réseaux. Le nouveau masque s'écrit comme suit :

255	255	255	224
11111111	11111111	11111111	11100000

Ce nouveau masque permet d'adresser huit sous réseaux. Ce masque peut aussi être indiqué par /27 à la suite de l'adresse IP. Il reste donc cinq bits pour les adresses hôtes. Ces cinq bits peuvent adresser 30 (32-2) hôtes par sous réseau (on soustrait 2 parce que les adresses locales et de diffusion ne sont pas comptées).

L'adresse de diffusion est par définition celle qui n'est codée que de 1 dans le champ d'adresse hôte. Puisque le champ d'adresse hôte comprend les cinq derniers bits, l'adresse de diffusion pour chacun de ces sous-réseaux est en représentation binaire 11111 ou 31 en décimale.

Il serait intéressant de lister tous les sous-réseaux. Les adresses de ces sous-réseaux sont :

192.168.1.0/27	11000000 10101000 00000001 **000**00000
192.168.1.32/27	11000000 10101000 00000001 **001**00000
192.168.1.64/27	11000000 10101000 00000001 **010**00000
192.168.1.96/27	11000000 10101000 00000001 **011**00000
192.168.1.128/27	11000000 10101000 00000001 **100**00000
192.168.1.160/27	11000000 10101000 00000001 **101**00000
192.168.1.192/27	11000000 10101000 00000001 **110**00000
192.168.1.224/27	11000000 10101000 00000001 **111**00000

Il est déconseillé d'utiliser les sous-adresses 000 et 111 surtout si les routeurs exploités dans le réseau ont recours au protocole de routage RIPv1 (voir chapitre 11 sur le routage). Dans ce cas, il ne nous reste plus que six sous-réseaux de disponibles.

Le problème de sous-réseau peut être envisagé sous un autre angle. Plutôt que de préciser le nombre de sous-réseaux désirés, la question viserait à définir des sous-réseaux ayant au minimum un certain nombre connu d'hôtes.

Considérons cette fois-ci l'adresse réseau privée suivante : 172.25.0.0 avec le masque 255.255.0.0 (ce masque est le masque implicite de la classe B dont fait partie l'adresse). Il est question de définir des sous-réseaux dont chacun doit inclure au moins 55 hôtes. Nous allons tout d'abord représenter l'adresse sous sa forme binaire :

172	25	0	0
10101100	00011001	00000000	00000000
255	255	0	0
11111111	11111111	00000000	00000000

Pour desservir les 55 adresses hôtes, le champ adresse hôte doit être de six bits ($2^6 - 2 = 62 \geq 55$). On se doit ainsi de réserver six bits pour les adresses hôtes. Ceux-ci peuvent desservir 64 adresses distinctes dont 62 correspondant à des adresses hôtes (soit bien plus qu'il n'en faut).

Les autres bits peuvent donc être réservés en vue de définir des sous-réseaux. De ce fait, dix bits sont disponibles pour les sous-réseaux. Ils peuvent donner lieu à 1022 (1024-2) sous-réseaux. Le nouveau masque prend la forme suivante : 255.255.255.192.

172	25	0	0
10101100	00011001	00000000	00000000
255	255	255	192
11111111	11111111	**11111111**	**11**000000

9.6.6 Masque à longueur variable

L'adressage IPv4 fondé sur les classes exploite une adresse comportant deux sections : la première étant le préfixe réseau et la seconde le numéro d'hôte.

Lorsque le sous-adressage a été adressé, le numéro d'hôte a été désigné pour représenter deux nouvelles sous-sections : le numéro de sous-réseau et le numéro d'hôte. Le sous-réseautage (*Subnetting*) appliqué par certains routeurs sur les adresses de sous-réseau, présente un avantage : celui de ne pas perturber le routage sur l'Internet. Seuls les routeurs de l'organisation ou du fournisseur de

réseau pourront reconnaître les adresses de sous-réseau et acheminer les datagrammes vers leur point de destination.

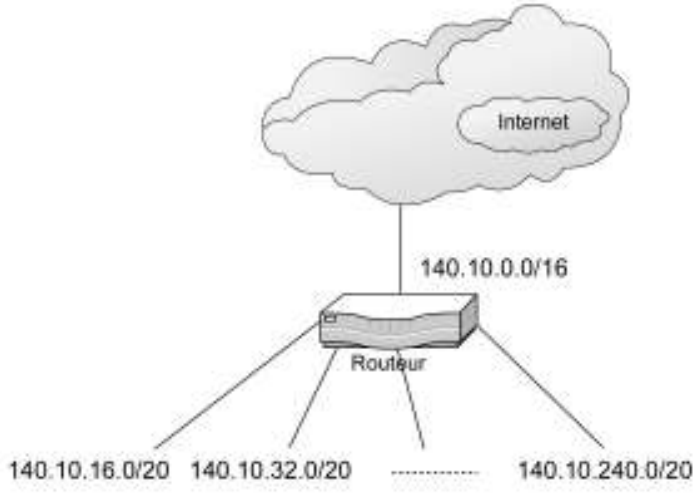


Figure 9.12 Sous-réseautage

Le mécanisme de sous-réseautage n'a aucune incidence sur le nombre absolu d'adresses Internet. Par contre, le sous-adressage permet d'optimiser l'utilisation des adresses disponibles.

Sur Internet, le routage ne prend en compte que le préfixe réseau.

L'adresse IP comprend le préfixe réseau, le numéro du sous-réseau et l'adresse hôte. Lorsqu'une adresse de sous-réseau est établie avec son masque, selon la méthode du sous-adressage, les sous-réseaux générés sont tous de la même longueur ; c'est-à-dire que chacun des sous-réseaux comprend le même nombre de sous-adresses. Par exemple, l'adresse 140.10.0.0/16, illustrée à la figure 9.12, peut desservir 65 534 adresses hôtes. Même si l'un de ces sous-réseaux n'a pas besoin d'autant d'adresses, il se les voit assignées. Par conséquent, un bon nombre de ces adresses sont perdues.

Pour remédier à ce problème, chaque adresse sous réseau peut prendre un masque différent et conforme au besoin du sous réseau. On qualifie cette option de masque de sous-réseau à longueur variable, soit le VLSM (*Variable Length Subnet Mask*). Le sous-adressage peut survenir à plusieurs niveaux.

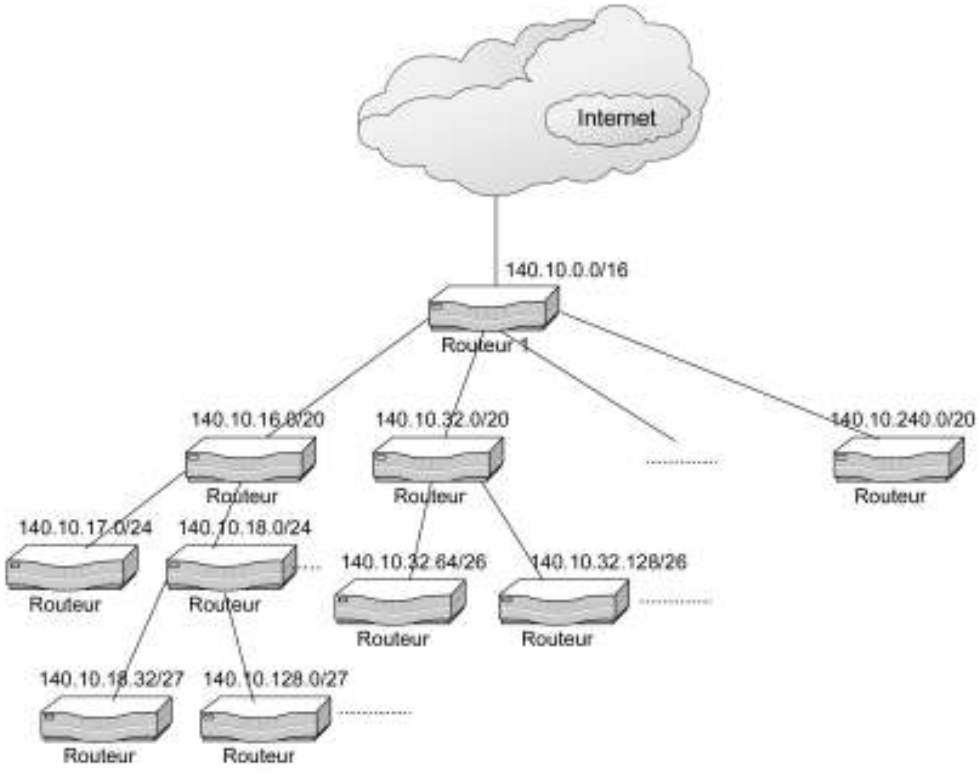


Figure 9.13 Représentation d'un sous-réseautage VLSM

Prenons l'exemple de l'adresse 140.10.0.0/16 représentée à la figure 9.13. Le préfixe réseau est 140.10. L'Internet achemine tous les datagrammes dotés de ces deux premiers octets de l'adresse au routeur 1. Ce routeur utilise un nouveau masque de 20 bits et peut à son tour avoir accès à 15 sous-réseaux commençant par l'adresse sous réseau 140.10.16.0/20 et allant jusqu'à 140.10.240.0/20.

La représentation binaire de ces adresses nous permettra d'avoir une meilleure compréhension de ce concept. Il est bon de mentionner que les chiffres binaires soulignés nous réfèrent à l'adresse du sous-réseau et les chiffres gras le nouveau masque. Ainsi, les adresses du routeur à l'accès Internet sont :

140.10.0.0/16 **10001100** **00001010** 00000000 00000000

Les adresses d'accès du routeur 1 sont :

140.10.16.0/20	10001100	00001010	**0001**0000	00000000
140.10.32.0/20	10001100	00001010	**0010**0000	00000000
140.10.48.0/20	10001100	00001010	**0011**0000	00000000
140.10.64.0/20	10001100	00001010	**0100**0000	00000000
140.10.80.0/20	10001100	00001010	**0101**0000	00000000
140.10.96.0/20	10001100	00001010	**0110**0000	00000000
140.10.112.0/20	10001100	00001010	**0111**0000	00000000
140.10.128.0/20	10001100	00001010	**1000**0000	00000000
140.10.144.0/20	10001100	00001010	**1001**0000	00000000
140.10.160.0/20	10001100	00001010	**1010**0000	00000000
140.10.176.0/20	10001100	00001010	**1011**0000	00000000
140.10.192.0/20	10001100	00001010	**1100**0000	00000000
140.10.208.0/20	10001100	00001010	**1101**0000	00000000
140.10.224.0/20	10001100	00001010	**1110**0000	00000000
140.10.240.0/20	10001100	00001010	**1111**0000	00000000

Le routeur dont l'accès est 140.10.16.0/20 atteint, à son tour, 15 autres sous-réseaux qui débutent à l'adresse 140.10.17.0/24 et se terminent à l'adresse 140.10.31.0/24.

À titre d'exemple, la représentation binaire des deux premières sous adresses est :

140.10.17.0/24	10001100	00001010	0001**0001**	00000000
140.10.18.0/24	10001100	00001010	0001**0010**	00000000

Le routeur dont l'accès est 140.10.32.0/20 accède, quant à lui, à 63 sous-réseaux commençant à l'adresse 140.10.32.64/26 et se terminant à l'adresse 140.10.47.192/26. Les deux premières adresses sont représentées en mode binaire :

140.10.32.64/26	10001100	00001010	0010**0000**	**01**000000
140.10.32.128/26	10001100	00001010	0010**0000**	**10**000000

La dernière adresse de ce sous-réseau est :

140.10.47.192/26	10001100	00001010	0010**1111**	**11**000000

Un autre niveau de sous-réseau commence à partir du routeur 140.10.18.0/24. Ce dernier a, à son tour, accès à sept autres sous-réseaux dont les adresses sont les suivantes :

140.10.18.32/27	10001100	00001010	00010010	**001**00000
140.10.18.64/27	10001100	00001010	00010010	**010**00000
140.10.18.96/27	10001100	00001010	00010010	**011**00000
140.10.18.128/27	10001100	00001010	00010010	**100**00000
140.10.18.160/27	10001100	00001010	00010010	**101**00000
140.10.18.192/27	10001100	00001010	00010010	**110**00000
140.10.18.224/27	10001100	00001010	00010010	**111**00000

L'adresse de diffusion (*Broadcast*) des sous-réseaux 140.10.18.96/27 est :

140.10.18.127/27	10001100	00001010	00010010	**011**11111

9.6.7 L'adressage sans classe

Le concept du routage inter-domaine sans classes, ou CIDR (*Classless Inter-Domain Routing*), a été pensé en vue d'améliorer l'utilisation et l'allocation des adresses IPv4.

Le CIDR fonctionne exactement de la même façon que le VLSM traité précédemment. La différence entre le CIDR et le VLSM tient au fait que le CIDR répartit les adresses à travers Internet alors que le VLSM le fait au niveau d'une organisation seulement.

Le CIDR, défini dans les RFC 1517, 1518, 1519 et 1520, n'utilise pas le concept de classes d'adresses vu précédemment. Les adresses utilisent uniquement le masque pour définir le préfixe réseau. Par ailleurs, le masque peut prendre n'importe quelle longueur, quelle que soit l'adresse, sans devoir se soucier de la classe à laquelle cette dernière pourrait appartenir.

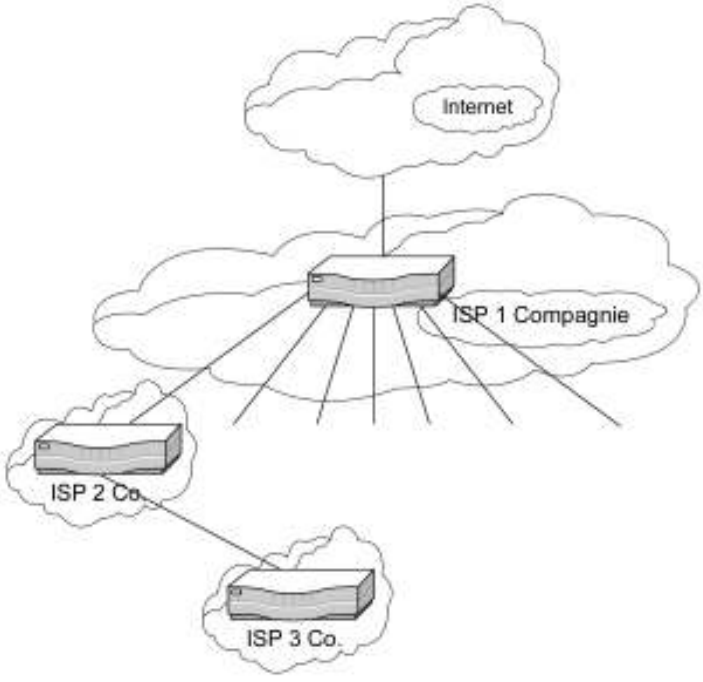


Figure 9.14 Répartition de fournisseurs Internet

Le CIDR implique les fournisseurs de services à tous les niveaux d'adresses comme il est illustré dans la figure 9.14. Le VLSM et le CIDR se conforment aux mêmes conditions, listées ci-dessous, à part la dernière qui est spécifique au CIDR :

- Les préfixes étendus (*Extended Network prefix*) comprennent le préfixe réseau (*Network prefix*) et le numéro de sous-réseau (*Subnet number*), lesquels doivent être acheminés par le protocole de routage.

  Tous les protocoles de routage n'acheminent pas nécessairement les masques de sous-réseau. Dans une telle situation, le CIDR ne fonctionnerait pas. Il faut donc s'assurer que les protocoles qui le font, comme OSPF ou IS-IS ou même RIPv2, soient dans les routeurs du sous-réseau. Il est bon de noter que le protocole de routage RIPv1 n'achemine pas les masques et n'est donc pas approprié dans le cas présent.

- L'algorithme d'acheminement par correspondance le plus long (*Longest Match Based Forwarding Algorithm*) doit être implémenté dans les routeurs.

Cela revient à dire tout simplement que plus le préfixe étendu est long plus précis sera le choix du chemin par routeur. Effectivement, ce dernier prend sa décision en se fondant sur la correspondance la plus longue. Ainsi, un datagramme dont l'adresse de destination est 140.10.5.1 sera reçu par le routeur dont les entrées suivantes figurent dans sa table de routage :

140.10.0.0/16 et 140.10.5.0/24

De la sorte, le routeur choisira le chemin indiqué par la correspondance la plus longue, soit 140.10.5.0/24.

- Pour permettre l'agrégation des tables de routage, il faut que les adresses du sous-réseau soient hiérarchiquement assignées. Celles-ci devront se soumettre à une topologie arborescente du réseau. Ce cas a déjà été illustré à la figure 9.13.

Outre ces conditions, communes au CIDR et au VLSM, une quatrième condition propre au CIDR est exigée :

- Les hôtes, tout comme les routeurs qui font partie d'un environnement CIDR, doivent fonctionner sans tenir compte des classes d'adresse. Ainsi, le routeur qui reçoit un datagramme pour l'acheminer n'analyse pas la classe de l'adresse destination, mais se base plutôt sur le masque qui accompagne l'adresse pour en connaître le préfixe étendu. Une adresse pourrait avoir un masque /16 alors que, selon sa classe, elle pourrait être de classe C. La classe C ne peut agir que sur un masque /24. À titre d'exemple, prenons l'adresse 192.100.60.4/16. Le préfixe réseau est 192.100, parce que le masque est de 16 bits. Si les classes devaient être prises en considération, l'adresse serait identifiée comme étant de classe C et l'adresse réseau serait 192.100.60 puisque, implicitement, le masque de classe C est de 24 bits (/24).

Le CIDR est très efficace lorsque les assignations d'adresses faites par chaque fournisseur de service ne changent pas. Par contre, si une organisation décide de changer de fournisseur dont les préfixes réseau sont différents, il sera plus simple à ladite organisation de conserver les mêmes adresses plutôt que de devoir restructurer tout l'adressage de son sous-réseau.

Les routeurs du nouveau fournisseur doivent par contre tenir compte de ces nouvelles adresses et faire une annonce d'« exception » sur Internet. L'annonce proclame que les adresses qui suivent ne seront plus acheminées par le fournisseur de services habituel, mais bien plutôt par un autre.

Lorsqu'on a recours à un grand nombre d'annonces d'« exceptions », l'efficacité du CIDR s'en trouve amoindrie et les tables de routage deviennent plus grandes.

9.6.7.1 Exemple d'application CIDR

Nous présentons un exemple pour mieux comprendre l'application de ce concept. Un important fournisseur de services se voit allouer le bloc d'adresses 192.160.0.0/16. Il est orienté CIDR.

Trois organisations lui demandent de leur fournir des adresses IP selon leurs besoins. L'organisation Orphée a besoin de 2000 adresses IP pour tous ses besoins présents et futurs. L'organisation Aphrodite a besoin de seulement 1000 adresses IP. La troisième organisation, Pénélope, demande 4000 adresses IP.

Pour plus de simplicité, nous allons utiliser dans notre solution les masques contigus.

Pour combler le besoin d'adresses IP des trois organisations, nous allons déterminer tout d'abord le nombre de bits nécessaires à allouer à chacun des groupes.

Organisation	Nombre d'hôtes	Valeur binaire	Nombre de bits	Masque requis
Orphée	2000	11111010000	11	/21
Aphrodite	1000	1111101000	10	/22
Pénélope	4000	111110100000	12	/20

Le fournisseur dispose au départ de l'adresse et du masque suivants :

192.160.0.0/16	11000000	10100000	00000000	00000000
255.255.0.0	11111111	11111111	00000000	00000000

Les masques alloués aux organisations sont les suivants :

Orphée	255.255.248.0	**11111111.11111111.11111**000.00000000
Aphrodite	255.255.252.0	**11111111.11111111.111111**00.00000000
Pénélope	255.255.240.0	**11111111.11111111.1111**0000.00000000

Toutes les adresses commencent par 192.160. C'est au troisième octet que l'adresse doit être unique. Il devra donc avoir un code différent pour tous les sous-réseaux offerts par le fournisseur de service.

Puisque l'adresse réseau doit être unique, l'exclusivité ne peut se faire que sur les quatre bits les plus significatifs du troisième octet. Ce sont les seuls bits communs aux trois organisations.

Le troisième octet peut être codé de la façon suivante pour les trois organisations :

Orphée	**1000**0xxx
Aphrodite	**0100**00xx
Pénélope	**1100**xxxx

Les x font partie de l'adresse hôte. Nous voyons que le code des bits communs est propre à chacune des organisations. Pourquoi avoir commencé par les bits les plus significatifs (1000, 0100, 1100)? Tout simplement en prévision des prochains sous-réseaux qui seront assignés par le fournisseur de service.

Nous pouvons maintenant assigner les adresses aux trois organisations en supposant que leurs adresses sont uniques et qu'elles n'ont pas déjà été assignées par le fournisseur de services à d'autres organisations. L'assignation des adresses est :

Orphée de 192.160.128.1/21	**11000000**	**10100000**	**10000**000	00000001
à 192.160.135.254 /21	**11000000**	**10100000**	**10000**111	11111110
Aphrodite de 192.160.64.1/22	**11000000**	**10100000**	**01000**00	00000001
à 192.160.67.254 /22	**11000000**	**10100000**	**01000**11	11111110
Pénélope de 192.160.192.1/20	**11000000**	**10100000**	**1100**0000	00000001
à 192.160.207.254 /20	**11000000**	**10100000**	**1100**1111	11111110

Orphée aura 2046 adresses IP, Aphrodite aura 1022 adresses IP et Pénélope aura 4094 adresses IP.

9.6.7.2 Récapitulation des routes (Route Summarization)

Nous avons déjà vu aussi bien pour VLSM que pour CIDR que pour permettre l'agrégation des tables de routage, il faut que les adresses du sous-réseau soient hiérarchiquement assignées. Ceci permettra au routeur d'utiliser efficacement leur protocole de routage.

Les réseaux peuvent être très grands et comprendre des centaines voire des milliers de sous-réseaux. Les protocoles de routage de chaque sous-réseau annoncent toutes les adresses des sous-réseaux qu'ils supportent à leur tour.

Toutes ces annonces deviennent extrêmement nombreuses et peuvent ralentir le traitement des datagrammes lors de leur acheminement par les routeurs. Ceci arrive à cause des longues listes d'entrées dans les tables de routage et le traitement continu du CPU.

La solution à ce problème est la récapitulation des routes (*Route Summarization*) par les routeurs. La seule chose que le routeur qui se trouve à la tête d'une arborescence de sous-réseaux puisse annoncer est, non pas tous les sous-réseaux qu'il supporte, mais seulement la partie de leur adresse réseau commune à tous. Une seule annonce remplacerait toutes les annonces de sous-réseaux. Ceci n'est possible que s'il y a agrégation des adresses.

Un exemple de récapitulation est illustré dans la figure 9.15 :

Nous avons quatre réseaux ayant les adresses représentées comme suit en *dotted decimal* et en binaire:

10.10.101.0	**00001010.00001010.0110**0101.0000000
10.10.103.0	**00001010.00001010.0110**0111.0000000
10.10.105.0	**00001010.00001010.0110**1001.0000000
10.10.107.0	**00001010.00001010.0110**1011.0000000

Nous remarquons que les 20 bits les plus significatifs des adresses réseaux sont communs. Il existe une différence à partir du 21e bit. Le routeur qui supporte tous ces réseaux en leur donnant accès à l'extérieur peut annoncer au nœud d'accès à Internet qu'il supporte tous les sous-réseaux qui ont les bits représentant l'adresse réseau 10.10.96.0/20 et tous les sous-réseaux qui pourraient être engendrés. Ainsi au lieu d'annoncer chacun des sous-réseaux, il n'annonce qu'une seule adresse réseau, notamment 10.10.96.0/20.

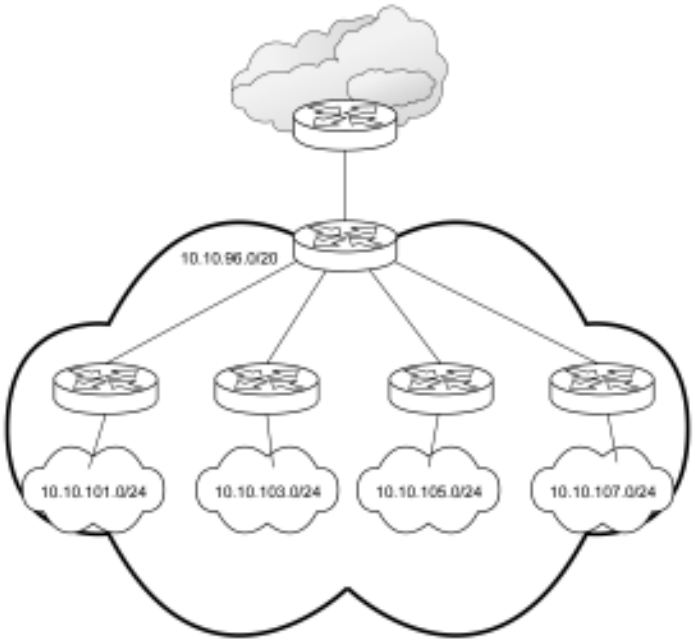


Figure 9.15 Récapitulation des routes

9.7 IPv6

Le protocole IPv4 permet l'adressage d'un grand nombre de réseaux et d'hôtes. Grâce aux 32 bits d'adresses, IPv4 peut théoriquement adresser près de 4 milliards de destinations. Comme nous l'avons vu précédemment, le système d'adressage limite le nombre de destinations possibles. On est malgré tout très proche des limites tolérables d'adresses disponibles.

L'IETF s'est attaqué, depuis 1992, à ce problème et développe le protocole IPv6 qui est le protocole de nouvelle génération. Il est d'ailleurs aussi connu sous l'acronyme de IPng (IP *next generation*).

Selon le RFC 2460, IPv6 apporte des changements marquants par rapport au IPv4. Les principales contributions de IPv6 sont :

- Capacité d'adressage plus étendue puisque le IPv6 utilise 128 bits pour représenter une adresse au lieu de 32.
- En-tête simplifié. Ceci permet un traitement rapide du paquet IPv6 par le routeur.

- Amélioration du support disponible à toutes extensions de l'en-tête ou options.
- Étiquetage des paquets pour permettre la différenciation de trafic.
- Capacité d'authentification et confidentialité.

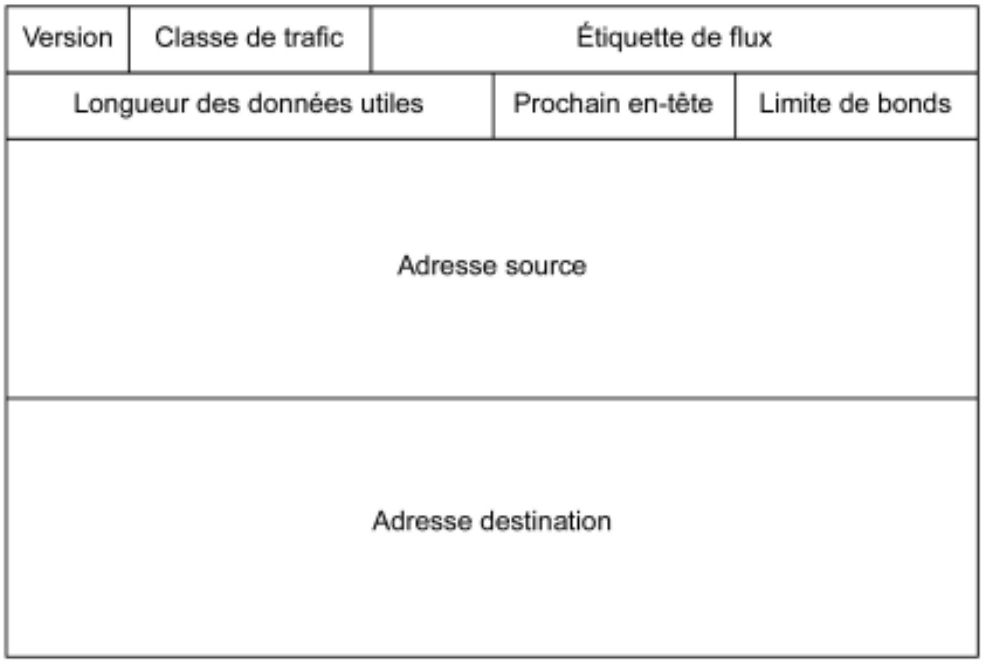


Figure 9.16 En-tête du paquet IPv6

L'en-tête du paquet IPv6 illustré dans la figure 9.16 comprend les champs suivants :

- Version – Ce champ de 4 bits est codé à la valeur 6.
- Classe de trafic – Long de 8 bits, ce champ est utilisé par les nœuds émetteurs et de réacheminement pour identifier les différentes classes ou priorité du paquet IPv6.
- Étiquette de flux – C'est un champ long de 20 bits qui peut être utilisé par la source pour étiqueter des séquences de paquets de façon à aviser les routeurs IPv6 de les traiter spécialement par exemple lorsqu'un service temps réel est requis.
- Longueur des données utiles – La longueur des données du paquet IPv6 est spécifiée dans ce champ de 16 bits.
- Prochain en-tête – Étant donné que plusieurs types d'extensions ou d'option sont possibles, ce champ de 8 bits permet d'identifier l'en-tête de celui qui sera utilisé juste après l'en-tête IPv6 (TCP, ou routage).

- Limite de bonds – Ce champ de 8 bits ressemble au champ TTL du IPv4. Il est décrémenté de 1 par le nœud qui réachemine le paquet. Si le contenu de ce champ est 0 le paquet est détruit.
- Adresses source et destination – Chacune de ces adresses est codée sur 128 bits.

9.7.1 Extension d'en-tête IPv6

Les options de IPv6 ne font pas partie de l'en-tête de base. De plus, il existe différents types d'extensions qui peuvent être utilisés. L'en-tête de ces extensions est placé après l'en-tête du paquet.

Les extensions possibles utilisées dans un paquet IPv6 sont :

- Options saut par saut
- Options destination
- Routage
- Fragment
- Authentification
- Charge de sécurité encapsulée

L'extension « Authentification » est définie dans le RFC 2402 et l'extension « Charge de sécurité encapsulée » l'est dans le RFC 2406. Les autres extensions sont définies dans le RFC 2460.

Tous les en-têtes de ces extensions qui sont identifiés dans les champs « Prochain en-tête » à part celui des *options saut par saut* ne sont traités que par le nœud de destination.

Chaque en-tête d'extension comprend aussi un champ « Prochain en-tête » pour identifier l'en-tête de la prochaine extension qui est utilisée. S'il n'y a pas de prochaine extension, le dernier champ « Prochain en-tête » indique le TCP.

La figure 9.17 comme dans le RFC 2460 illustre ce mécanisme. Les en-têtes des extensions doivent respecter l'ordre si les autres extensions sont présentes. Ainsi, l'en-tête d'extension ne peut pas apparaître avant celui de routage.

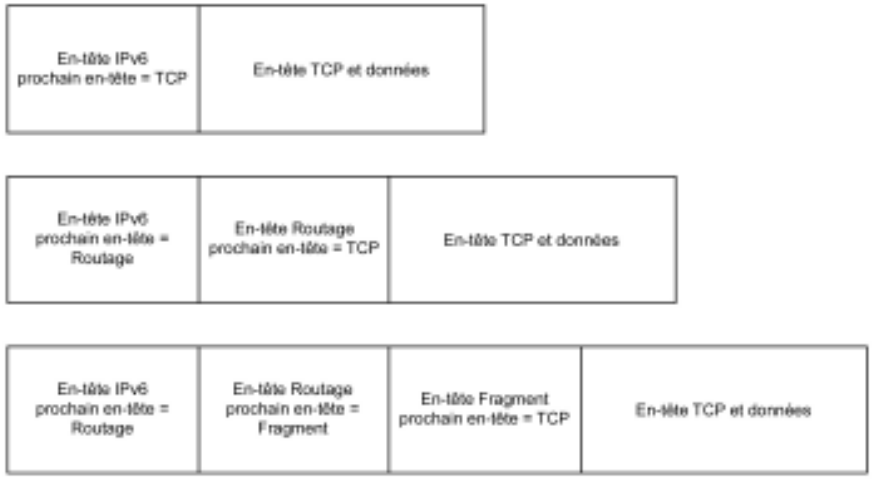


Figure 9.17 Identification des en-têtes d'extensions

9.7.2 Modèle d'adressage IPv6 (RFC 3513)

L'adresse IPv6 comprend 128 bits et identifie des interfaces ou ensemble d'interfaces. Il existe trois types d'adresses IPv6 :

- Unicast – L'adresse identifie une seule interface.
- Multicast – Une seule adresse multicast identifie un groupe d'interfaces qui ne se trouvent pas nécessairement sur le même nœud. Un paquet transmis avec ce type d'adresse est reçu par chacune des interfaces du groupe.
- Anycast – Comme pour le multicast, l'adresse anycast identifie un groupe d'interfaces. Toutefois, les paquets transmis avec ce type d'adresse ne sont reçus que par un membre du groupe (le plus proche par exemple).

Il n'existe pas de type d'adresses pour la diffusion. C'est le multicast qui est utilisé à la place.

L'adresse IPv6 est normalement représentée par 8 valeurs hexadécimales, une par 16 bits, séparées par deux points (:). Pour reprendre l'exemple du RFC 3513 nous avons la représentation d'une adresse valable :

FEDC :AB98 :7654 :3210 :FEDC :BA98 :3210

Lorsque des valeurs de l'adresse sont à 0, certaines règles doivent être respectées. Si une valeur comprend des 0 dans la partie la plus significative, il n'est pas

nécessaire de les écrire. Par exemple, la valeur 0123 peut s'écrire 123. Lorsque la valeur est complètement 0 on peut suivre la syntaxe suivante :

- La valeur 0000 peut être remplacée par le signe **::** ; ce signe ne peut paraître qu'une fois dans l'adresse. Ainsi l'adresse suivante :

  9876:0:5432:10:1234:5678:6543:5874

  peut-être écrite

  9876::5432:10:1234:5678:6543:5874

- S'il y a plus d'une valeur 0 consécutive, on remplace toutes ces valeurs 0 par **::** .

  Par exemple, 78:0:0:0:47:80:90:76 est remplacé par 78::47:80:90:76
  Par contre, il faut faire attention lorsque les valeurs 0 sont séparées.
  Par exemple, 78:0:0:77:0:0:46:21 est remplacé par 78::77:0:0:46:21
  ou par, 78:0:0:77::46:21

Une autre forme alternative est de coder les six premières valeurs de 16 bits en hexadécimale et remplacer les deux dernières valeurs de 16 bits en quatre chiffres *dotted decimal* comme dans le format de IPv4. Par exemple l'adresse suivante IPv6 :

0:0:0:0:0:EF28:AC32:2741 peut être représentée sous la forme suivante :
0:0:0:0:0:EF28:172.50.39.65 ou sous sa forme plus compacte
::EF28:172.50.39.65

Le préfixe de sous-réseau du nœud est indiqué comme dans le CIDR du IPv4. Une adresse du nœud avec le numéro de son sous-réseau peut être, par exemple :

Adresse de nœud	9876:0:5432:10:1234:5678:6543:5874
Numéro du sous-réseau	9876:0:5432:10::/60

Certaines adresses sont spécifiquement définies. Il y a l'adresse 0:0:0:0:0:0:0:0 ou ::/128 non spécifiée qui ne peut pas être assignée à un nœud ni être réacheminée par ce dernier. Elle est utilisée par un hôte qui essaye de connaître sa propre adresse.

Les systèmes d'exploitation définissent des adresses IPv6 aux interfaces réseaux. Ce sont les *Link-local addresses* ou adresses de liens locaux. Le préfix de ces adresse est fe80 :: selon la notation IPv6.
Il y a l'adresse boucle avec retour (*loopback*) 0:0:0:0:0:0:0:1 ou ::1/128 qui est utilisée par un nœud pour s'envoyer un paquet IPv6. Son effet est local. Le nœud ne réachemine pas ce paquet.

Toutes les adresses multicast ont leur préfixe de 8 bits à 1. Ceci est identifié par la notation suivante : FF00 ::/8. Le format de l'adresse multicast illustré dans la figure 9.18 comprend en plus du préfixe, 4 bits du champ drapeau, 4 bits du champ portée, et 112 bits pour l'identifiant du groupe.

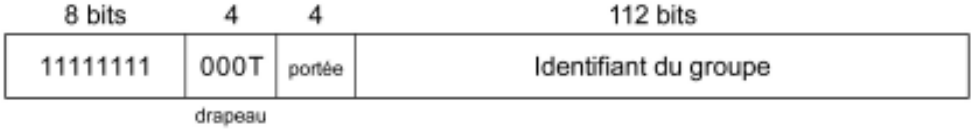


Figure 9.18 Adresses multicast

Seul le bit T est utilisé dans le champ drapeau. S'il est à 0, l'adresse multicast de groupe est une adresse connue assignée par IANA. Par contre si T est à 1, l'adresse n'est pas permanente.

Le champ « Portée » est utilisé pour limiter la portée du groupe multicast. Par exemple lorsque ce champ est codé 1 la portée de l'adresse est locale et se limite à la même interface. Lorsque le champ est codé E hexadécimal la portée est globale et inclut Internet.

L'adresse anycast, comme l'adresse multicast, ne peut être utilisée comme adresse source. L'adresse anycast ne peut être assignée qu'à un routeur mais pas à un hôte. L'adresse anycast de sous-réseau du routeur est prédéfinie et comporte le format suivant :

N bits de préfixe sous-réseau suivi de 128 – N bits 0. Ceci est illustré dans la figure 9.19.

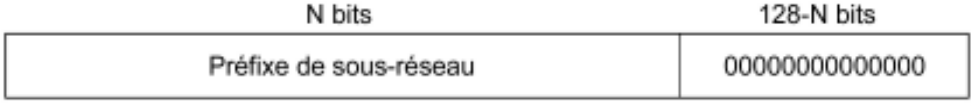


Figure 9.19 Adresse anycast

Il existe aussi des formats d'adresse unicast développés pour la transition IPv6. Ceci permet d'utiliser dynamiquement un tunnel dans l'infrastructure IPv4 pour des paquets IPv6. Les adresses spéciales unicast IPv6 utilisées dans des nœuds IPv6 permettant la transition comprennent 96 bits de 0 comme préfixe et 32 bits comme adresse valable IPv4.

Un deuxième type d'adresse unicast IPv6 est utilisé pour permettre la transition de IPv4 à IPv6. Le format de cette adresse IPv6 comprend 80 bits de 0 comme préfixe suivi de 16 bits de 1, et suivi de l'adresse IPv4.

Il n'existe pas de NAT pour les adresses IPv6. Le NAT n'est appliqué qu'à l'adressage IPv4 pour permettre de multiplier les accès avec l'utilisation d'un nombre réduit d'adresse. IPv6 n'a nullement besoin de ce mécanisme.

IPv6 peut être utilisé sur un réseau Ethernet. Dans ce cas l'adresse MAC peut composer l'adresse IPv6 de la façon suivante. Sachant que l'adresse MAC est composée de 48 bits dont les 24 premiers bits représentent le code du manufacturier (OUI), deux octets FF FE sont insérés juste après. Ainsi le nouveau code nommé EUI-64 est de 64 bits. L'adresse IPv6 de 128 bits comprend 45 bits de préfixe de réseau, 16 bits pour identifier le site du sous réseau et 64 bits pour identifier l'interface pouvant être le EUI-64. Afin d'indiquer que l'adresse EUI-64 représente une adresse d'équipement, le septième bit est changé de zéro à un.

9.7.3 Transition de IPv4 à IPv6 (RFC 4213)

Il est évident qu'une transition rapide ne peut être effectuée de IPv4 à IPv6. Toutefois, une adresse IPv4 peut être intégrer dans une adresse IPv6 dans le format décrit précédemment. Tel qu'indiqué sur la figure 9.20, il est par contre nécessaire de développer des méthodes qui permettraient d'acheminer des datagrammes IPv6 à travers des réseaux IPv4 vu que ces derniers existeront pour un certain temps et feront partie de la toile pour longtemps.

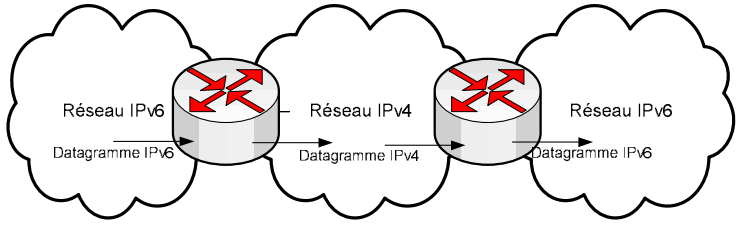


Figure 9.20 Réseaux de transition

Deux méthodes sont proposées à cette fin. La première est connue sous le nom de *Dual-Stack*. La deuxième est nommée *Tunneling*.

Il est important de noter que certains éléments de réseaux, tel que les routeurs, peuvent fonctionner en mode IPv4 aussi bien que IPv6 (RFC 4213). Lorsqu'un routeur se trouve en bordure d'un réseau IPv6 et IPv4, il doit appliquer l'une des méthodes précitées pour acheminer les datagrammes du réseau IPv6 à IPv4.

Lorsque la méthode *Dual-Stack* est appliquée, le routeur de bordure copie le champ de données du datagramme IPv6 reçu, dans le champ de données d'un datagramme IPv4. Il effectue par la suite une translation de l'adresse IPv6 reçue, à l'adresse correspondante IPv4 qu'il insère dans le datagramme ainsi que quelques options et le transmet dans le réseau IPv4.

La méthode *Dual-Stack* a toutefois un inconvénient, elle ne permet pas le transfert d'options particulières au protocole IPv6. Ces options sont donc perdues durant la transition. Elle permet par contre d'adresser un hôte dans un réseau IPv4 à partir d'une source dans un réseau IPv6.

La deuxième méthode *Tunneling* permet le transfert de datagramme IPv6 dans son intégralité à travers un réseau IPv4. Il faut par contre que la destination finale de ce datagramme soit dans un réseau IPv6. Le datagramme IPv6 est donc entièrement encapsulé dans un datagramme IPv4. Ceci est illustré dans la figure 9.21. Le champ de données du datagramme IPv4 comprend tout le datagramme IPv6 incluant son en-tête et tous ses champs. Ce nouveau datagramme IPv4 ainsi constitué est adressé à un routeur de bordure se trouvant sur le chemin vers la destination. Il devra donc traverser un ou des réseaux IPv4 avant d'atteindre ce routeur de bordure. Ce dernier dé-encapsule de datagramme IPv4 reçu et acheminera le datagramme IPv6 émergeant vers sa destination.

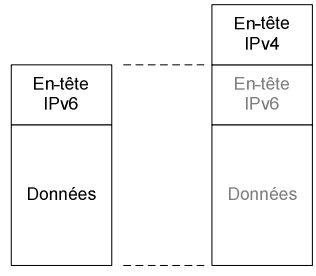


Figure 9.21 Encapsulation d'un datagramme IPv6 dans IPv4

Lorsque le MTU du réseau IPv4 dans lequel le nouveau datagramme doit être acheminé, est plus petit que la grandeur du datagramme, il est nécessaire d'effectuer une fragmentation. Les fragments IPv4 doivent être réassemblés au routeur de bordure accédant au réseau IPv6. Le problème réside essentiellement dans ce routeur de bordure récepteur des fragments. Ce routeur pourrait ne pas avoir la capacité de les traiter et de les réassembler soit par manque de mémoire ou par manque de ressources.

9.8 Exercices

9.1

Une application génère des données qui forment des NSDUs d'une longueur de 1024 octets. Ces NSDU doivent être transmis par le protocole IP. Le Path MTU signalé est de 256 octets. Pour l'un des NSDUs :

a) Si l'en-tête IP est à son minimum, donnez la valeur des champs impliqués dans la fragmentation des NPDUs.
b) Si le champ option utilise huit octets, donnez la valeur des champs impliqués dans la fragmentation.

9.2

Six (6) datagrammes avec en-tête de 20 octets sont reçus et ont tous le bit du fanion *More* à 1, sauf le dernier. Chacun de ces datagrammes est long de 1020 octets. Par contre, le dernier datagramme est long de 1000 octets. Donnez la longueur du NSDU qui sera reconstitué.

9.3

Les données hexadécimales suivantes représentent l'en-tête d'un datagramme.

a) Indiquez chacun des champs avec leur valeur.
b) Indiquez le nom du protocole utilisé dans la couche supérieure.

45 00 01 56 98 36 00 00 ff 11 17 fc 0a 64 00 01 ff ff ff ff

9.4

Pour les adresses suivantes, indiquez leur classe, leur adresse réseau et leur adresse hôte :

128.168.15.12
125.10.167.75
222.144.248.30
227.38.45.2

9.5

Pour les adresses suivantes, indiquez le préfixe réseau, l'adresse du sous-réseau, le masque et l'adresse hôte :

155.77.85.13/18
121.78.126.3/28
192.35.68.4/26
190.32.176.14/20

9.6

Pour les adresses suivantes, déterminez le nouveau masque requis pour obtenir le nombre d'hôtes demandés tout en maximisant le nombre de sous-réseaux :

Adresse	Nombre d'hôtes	Masque
148.72.0.0/16	125	_____
146.98.0.0/16	60	_____
172.126.192.0/18	350	_____
203.76.240.0/24	30	_____

9.7

Pour les adresses suivantes, déterminez le nouveau masque requis pour obtenir le nombre de sous-réseaux demandés tout en maximisant le nombre d'hôtes :

Adresse	Nombre de sous-réseaux	Masque
132.240.0.0/16	200	_____
190.128.64.0/18	100	_____
218.1.125.0/24	60	_____
195.192.135.0/24	32	_____

9.8

Un routeur est relié au réseau 165.236.0.0/19. Un masque y est appliqué afin d'accéder à sept sous-réseaux. Le sous-réseau 3 doit accéder à son tour à un nombre de sous-réseaux comprenant chacun 254 hôtes.

a) Indiquez le masque appliqué au premier sous-réseau.
b) Indiquez le masque appliqué au sous réseau 3 et lister les sous-réseaux qui en résultent.

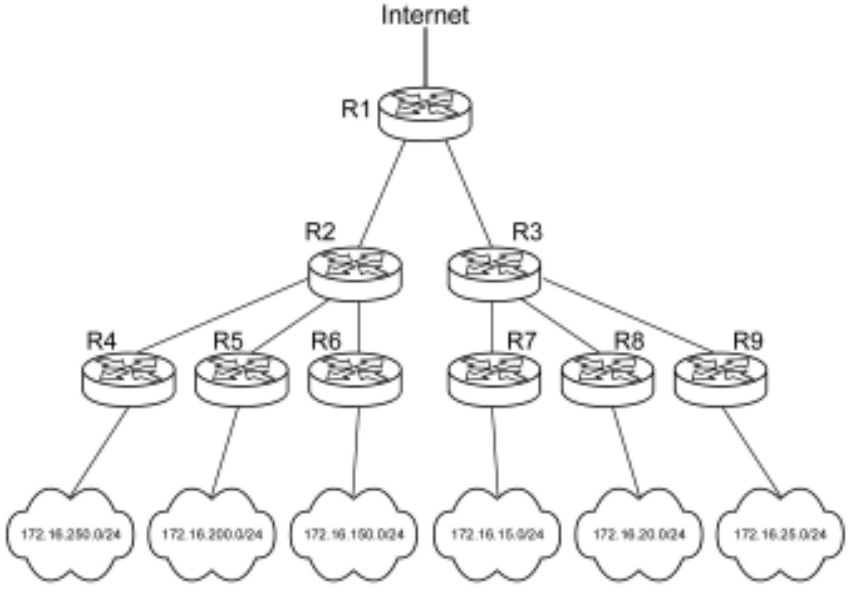
9.9

Une organisation voudrait développer un réseau à partir d'un seul routeur d'accès. L'adresse d'accès à ce routeur est : 188.205.128.0/18. L'organisation désire créer deux autres niveaux de sous-réseaux après le premier routeur (routeur d'accès). Le premier niveau doit comprendre pas moins de dix sous-réseaux et le deuxième niveau pas moins de cinq sous-réseaux pour chacun des réseaux du premier niveau.

a) Indiquez, en format binaire, le masque utilisé pour obtenir le premier niveau et donner les adresses des sous-réseaux du premier niveau.
b) Indiquez, en format binaire, le masque utilisé pour le deuxième niveau ainsi que les adresses des sous-réseaux.
c) Donnez le nombre d'hôtes supportés par le niveau 2 et inscrire les adresses hôtes pour l'un des sous-réseaux.

9.10

Une récapitulation des routes doit être appliquée par les routeurs R1 à R9. Quelles seront les annonces que devront faire chacun des routeurs au routeur du niveau supérieur?



9.11

Donnez toutes les représentations comprimées des adresses IPv6 suivantes :

a) 1000 :0001 :0000 :0000 :0001 :0000 :0000 :0000
b) 0000 :0000 :0000 :0001 :0000 :0000 :0000 :0000
c) FF00 :1234 :5678 :0000 :0000 :0000 :192.168.0.0
d) 1010 :0000 :1010 :0000 :1010 :0000 :1010 :0000

9.12

Indiquez si les adresses IPv6 suivantes sont valides, sinon donnez la bonne représentation :

a) ::ABCD:1234/48
b) ABCD::1234:0000/60
c) ABCD:5678:1234::/48
d) 4587:7895:5542::/60
e) 0:45:78:/48

9.13

Quelle serait la méthode de transition la plus appropriée dans les cas suivants. Expliquer :

a) Le terminal source est dans un réseau IPv6 et la destination dans un réseau IPv4.
b) Le terminal source est dans un réseau IPv4 et la destination dans un réseau IPv6.
c) Le terminal source est dans un réseau IPv4 et la destination dans un réseau IPv4; par contre les datagrammes doivent transiter par un réseau IPv6.
d) Le terminal source est dans un réseau IPv6 et la destination dans un réseau IPv6; par contre les datagrammes doivent transiter par un réseau IPv4.

10 Support au protocole IP

10.1 Introduction

La série de protocoles qui soutient le réseau Internet est connue, tout comme nous l'avons souligné au chapitre 9, sous le nom de « *Suite TCP/IP* » ou de « *Protocol Suite* ».

À titre indicatif la figure 10.1, adaptation de la figure 9.3, nous en donne un aperçu.

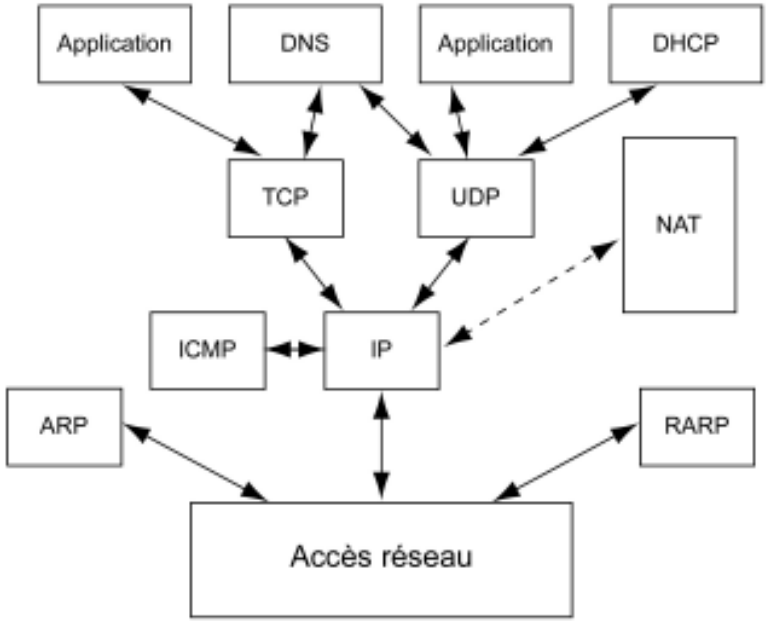


Figure 10.1 Illustration des protocoles d'Internet et leurs interrelations

Le présent chapitre traite des protocoles suivants :

Le protocole ARP (*Address Resolution Protocol*) qui permet d'obtenir l'adresse physique et d'amorcer la communication à partir de la couche liaison.

Le protocole RARP (*Reverse Address Resolution Protocol*) est utilisé par les stations dépourvues de mémoire afin de se procurer une adresse IP à partir de l'adresse MAC. Ce protocole est de moins en moins utilisé, car il est remplacé par le protocole DHCP dont nous traiterons plus loin.

Le protocole ICMP (*Internet Control Message protocol*) a été conçu comme complément aux fonctions manquantes du protocole IP. À titre indicatif, il s'agit ainsi de messages d'erreurs prévenant la source de la non réception des datagrammes.

Le protocole DHCP (*Dynamic Host Configuration Protocol*) nous permet d'assigner à titre provisoire des adresses IP.

Le service NAT (*Network Address Translation*) traduit des adresses IP internes ou privées en adresses IP publiques dans l'Internet.

Le service DNS (*Domain Name Service*) nous permet de faire la correspondance entre le nom et l'adresse IP d'un équipement donné.

10.2 Protocole de résolution d'adresse ARP

L'adresse IP ne revêt son plein sens que dans le réseau Internet. Toutefois, la communication entre terminaux doit tout d'abord être établie à partir de l'adresse matérielle. Dans un environnement de réseaux locaux cette adresse est déjà identifiée comme adresse MAC.

Par ailleurs, dans le chapitre 4 nous avons déjà précisé que, lorsqu'une station désire transmettre à une autre station il faut que l'adresse MAC du terminal de destination figure dans le champ Adresse de destination de la trame Ethernet. Si cette adresse n'est pas précisée, même si la bonne adresse IP se trouve dans le datagramme encapsulé dans la trame Ethernet, le terminal de destination ne pourra jamais savoir que cette trame lui est destinée.

Il ne serait pas pratique de devoir connaître l'adresse MAC des terminaux de destinations pour établir une communication. Par contre, il faut définir le mécanisme qui permettra d'acheminer les données vers le bon destinataire en ne

référant qu'à l'adresse IP. Un tel mécanisme devra faire appel à un procédé qui permettrait d'obtenir l'adresse MAC destination à partir de l'adresse IP.

Le protocole de résolution d'adresses ARP permet d'établir un lien entre les adresses IP et MAC d'une station donnée sur le réseau local. Lorsque le protocole ARP est appliqué au réseau Ethernet, il est possible d'obtenir, à partir de l'adresse IP de 32 bits de la station de destination, l'adresse MAC de 48 bits qui lui correspond.

La résolution d'adresse par le biais du protocole ARP doit se faire dans le même réseau. Ainsi, un terminal ne peut obtenir l'adresse MAC d'une autre station que dans la mesure où tous les deux se trouvent dans le même réseau.

Cependant, la communication ne peut pas être limitée aux terminaux d'un seul et même réseau. Ainsi, lorsqu'un terminal désire communiquer avec un autre terminal se trouvant dans un autre réseau local, la connexion et l'échange d'information restent conformes au modèle TCP/IP.

Si la station de destination se trouve dans un autre réseau, la station source devra alors passer par au moins un routeur pour pouvoir y accéder. Ce routeur se trouve dans le même réseau que la station source. Aussi, et pour lui transmettre des trames, la station source devra savoir quelle est l'adresse MAC du routeur.

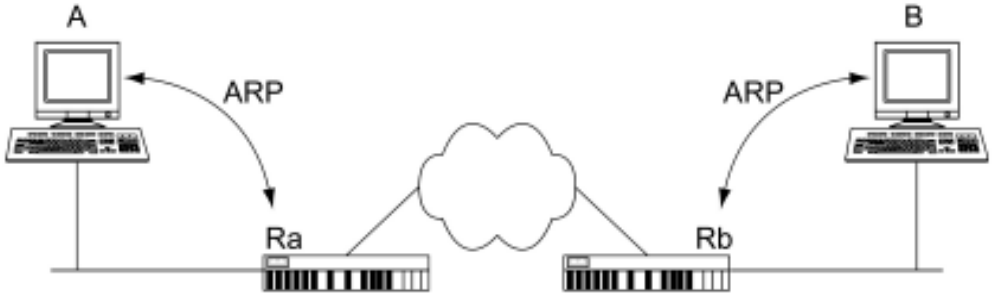


Figure 10.2 Illustration d'un ARP de bout en bout

En soumettant une demande de résolution d'adresse dans son réseau, le terminal source demande à toutes les stations, incluant le routeur, que celle qui reconnaît son adresse IP s'identifie.

Le routeur répond à cette demande en signalant son adresse MAC. La station source transmet par la suite une trame Ethernet comportant dans l'Adresse source son adresse MAC et celle du routeur comme adresse de destination. Toutefois, le datagramme IP contenu dans cette trame englobe d'une part

l'adresse source IP de la station émettrice et de l'autre l'adresse IP du terminal de destination de la station ciblée. La figure 10.2 illustre la situation d'un terminal source « A » qui désire communiquer avec le terminal destination « B ». Au niveau IP les adresses source et destination des datagrammes sont respectivement les adresses IP de « A » et de « B ». Par contre, au niveau MAC le terminal « A » adresse une trame MAC avec les adresses source et destination MAC de « A » et du routeur « Ra » respectivement. Ce dernier achemine l'information au routeur « Rb » qui va transmettre, toujours au niveau MAC, une trame MAC avec les adresses source et destination MAC du routeur «Rb » et du terminal « B » respectivement.

Chaque hôte ou station dispose d'une mémoire « cache » ou antémémoire ARP. Cette mémoire comporte les réponses ARP les plus récentes et fait état de la correspondance entre adresses IP et adresses MAC. Chaque fois qu'une station désire communiquer, elle consulte en premier lieu l'antémémoire ARP. Si l'adresse MAC recherchée et correspondant à une adresse IP s'y trouve, elle y fait appel, autrement elle transmet une demande ARP. Il importe de souligner que la réponse à cette demande mettra aussi à jour l'antémémoire ARP.

Le temps d'expiration des entrées dans l'antémémoire ARP varie entre 5 et 20 minutes tout dépendant des systèmes. Les entrées ne sont effacées que lorsqu'elles n'ont pas été utilisées pendant tout le temps que dure l'expiration. Une fois cette dernière écoulée, une nouvelle demande ARP devra être émise si on a besoin de cette adresse à nouveau étant donné que son entrée dans la mémoire n'existe plus.

La commande DOS « arp » nous permet de voir le contenu de l'antémémoire ARP. Cette commande est utilisée avec l'option ***a*** pour visionner le contenu. Exemple :

```
C:\> arp –a

Interface 192.168.100.10 on interface 0x10003
Interface Address      Physical Address      Type
192.168.100.1          00-05-9a-05-bc-15     dynamic
```

Les routeurs disposent également d'une antémémoire ARP que l'on peut consulter. Dans le cas des routeurs Cisco la commande est « show arp ».

Nous verrons le fonctionnement de la résolution d'adresse dans différents cas, par le biais d'exemples. Il importe cependant de bien comprendre le format du paquet ARP.

10.2.1 Format de paquet ARP

Tout comme le datagramme IP, le paquet ARP, est transporté dans une trame Ethernet. Ainsi, toute demande ARP doit être transmise dans une trame de diffusion pour que toutes les stations aient la possibilité de vérifier le contenu. Par contre, la *réponse* ne se trouve que dans une trame non diffusée (contenant une adresse destination) puisque seule la station qui reconnaît son adresse va retourner la réponse à la station source.

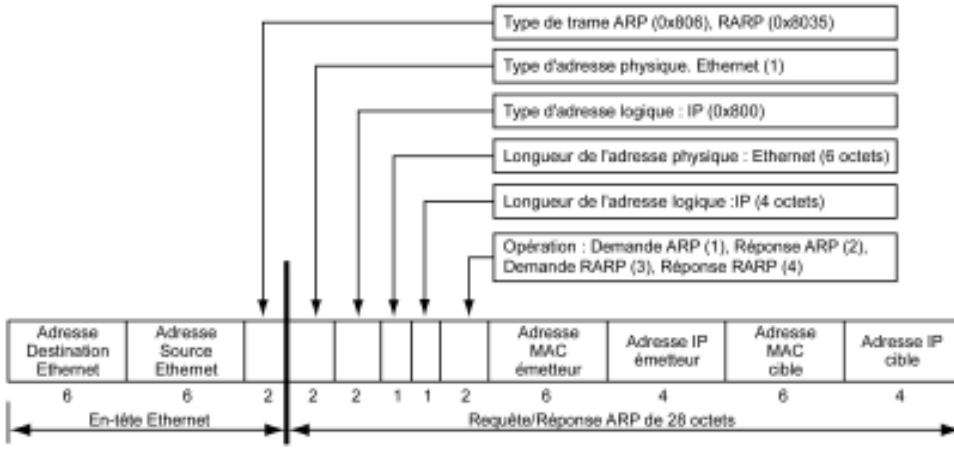


Figure 10.3 Format de paquet ARP et RARP

Il importe de noter que ce paquet est utilisé par le protocole ARP de même que par le protocole RARP que nous traiterons ultérieurement.

Le champ type comporte le code qui mentionne le protocole et donc le format du contenu de la trame. Le code du protocole ARP est 0x806 en hexadécimal. Celui du protocole RARP est 0x8035.

Le paquet ARP comprend 28 octets. Le protocole ARP n'a pas été pensé que pour Ethernet bien que de nos jours il ne soit pratiquement utilisé que par lui. La figure 10.3 illustre les champs du paquet.

- Le premier champ du paquet ARP désigne le type d'adresse physique. Dans le cas d'Ethernet, ce champ d'une longueur de 2 octets est codé à 1.
- Le deuxième champ, également d'une longueur de 2 octets, désigne le type d'adresse logique. Dans le cas présent, il s'agit de l'adresse IP et il existe un code en hexadécimal d'une valeur égale à 0x800.
- Le troisième champ, d'une longueur de 1 octet, fait état de la longueur de l'adresse physique qui est égale à 6 octets dans le cas d'une adresse MAC.

- Le quatrième champ indique la longueur de l'adresse logique. Pour IP la longueur de l'adresse logique est de 4 octets.
- Le cinquième champ souligne la nature même de l'opération du paquet.
    - Lorsqu'il s'agit d'une demande ARP le code est mis à 1. Dans ce cas, l'adresse IP de la source et celle du terminal de destination figurent parmi les champs qui suivent. Par contre, seule l'adresse MAC du terminal source s'y trouve mentionnée. Le champ de l'adresse MAC du terminal de destination est mis à 0 étant donné que cette adresse demeure inconnue.
    - Dans le cas d'une réponse ARP le code est égal à 2. Dans ce cas, toutes les adresses MAC et IP y sont mentionnées, surtout l'adresse MAC de l'adresse cible que l'on recherche. Il importe de distinguer que, dans la réponse ARP, l'ordre des adresses est inverse à celui des adresses de la demande ARP. L'adresse MAC recherchée se trouve donc dans le premier champ d'adresse MAC de la réponse ARP.
    - Pour une demande RARP le code est mis à 3.
    - Pour une réponse RARP le code est mis à 4.
- Les quatre derniers champs indiquent les adresses suivantes :

  Pour une demande ARP
    - Adresse MAC du transmetteur (6 octets)
    - Adresse IP du transmetteur (4 octets)
    - Adresse MAC cible (6 octets)
    - Adresse IP cible (4 octets)

  Pour une réponse ARP
    - Adresse MAC cible (6 octets)
    - Adresse IP cible (4 octets)
    - Adresse MAC du transmetteur original (6 octets)
    - Adresse IP du transmetteur original (4 octets)

10.2.2 Application du protocole ARP

Une station d'un réseau local désireuse d'établir une communication avec une autre station se trouvant dans le même réseau doit entamer la liaison par le biais de la couche Physique et doit connaître l'adresse MAC du terminal de destination. Tout comme nous l'avons vu, il est possible d'obtenir cette adresse à partir de l'antémémoire ARP. Si elle ne s'y trouve pas, une demande ARP est alors émise. La réponse permet de transmettre les trames et de mettre également à jour la mémoire « cache » ARP.

Si le terminal de destination ne se trouve pas dans le même sous-réseau, la station source devra alors acheminer ses trames par le biais d'un routeur. Si les adresses réseau IP ou les masques sont différents, la station de destination ne fait pas partie du même sous-réseau. Dans ce cas précis, la station source doit alors passer par le routeur pour établir la connexion physique.

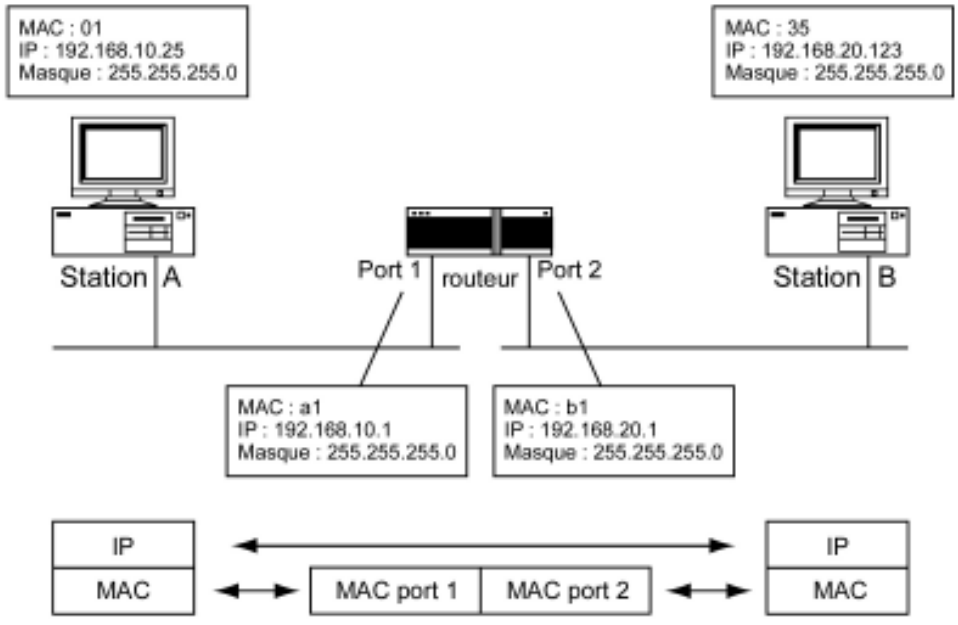


Figure 10.4 Illustration d'une des applications du protocole ARP

Prenons l'exemple de la figure 10.4. La station A désire transmettre des données à la station B. Les deux stations se trouvent dans des sous-réseaux distincts. La station A réalise que l'adresse IP de B qu'elle connaît, appartient à un sous-réseau différent. De ce fait, elle doit avoir recours à la liaison par le biais du routeur pour pouvoir acheminer ses trames.

La station A procède à la vérification de l'antémémoire ARP pour obtenir l'adresse MAC du routeur. La station A connaît l'adresse IP de son commutateur de passerelle (ou bien : de sa connexion du système hôte à la passerelle). Cette adresse fait partie de la configuration, celle du port 1 du routeur. Si cette adresse n'est pas connue ou ne se trouve pas dans l'antémémoire ARP, la station A met de côté les données qu'elle désire transmettre et émet une demande ARP pour l'obtenir.

Le contenu en hexadécimal de la trame comprenant une demande ARP revêt la forme suivante :

FF FF FF FF FF FF	adresse destination - diffusion
00 00 00 00 00 01	adresse source – station A
08 06	Type de contenu - ARP
00 01	Type d'adresse physique – Ethernet (1)
08 00	Type d'adresse logique - IP
06	Longueur de l'adresse physique
04	Longueur de l'adresse logique
00 01	Opération – demande ARP
00 00 00 00 00 01	Adresse MAC du terminal transmetteur
C0 A8 0A 19	Adresse IP du terminal transmetteur : 192.168.10.25
00 00 00 00 00 00	Adresse MAC du routeur – inconnue
C0 A8 0A 01	Adresse IP du routeur – 192.168.10.1
…..	Bourrage pour respecter la longueur minimale
…..	FCS

Le routeur reçoit cette trame par le port 1 et, reconnaissant son adresse IP, renvoie une réponse ARP pour donner l'adresse MAC de son port 1.

En hexadécimal, la teneur de la trame qui renferme la réponse ARP est :

00 00 00 00 00 01	Adresse destination – station A
00 00 00 00 00 A1	Adresse source – routeur port 1
0806	Type de contenu – ARP
00 01	Type d'adresse physique – Ethernet (1)
08 00	Type d'adresse logique - IP
06	Longueur de l'adresse physique
04	Longueur de l'adresse logique
00 02	Opération – réponse ARP
00 00 00 00 00 A1	Adresse MAC du transmetteur, le routeur
C0 A8 0A 01	Adresse IP du routeur : 192.168.10.1
00 00 00 00 00 01	Adresse MAC de la station A
C0 A8 0A 19	Adresse IP de la station A – 192.168.10.25
.	Bourrage pour respecter la longueur minimale
.	FCS

La station A obtient l'adresse MAC du port 1 du routeur. Elle peut donc reprendre ses données et les acheminer en utilisant l'adresse obtenue dans des datagrammes IP. Une trame de cette transmission serait en partie :

00 00 00 00 00 A1	Adresse MAC du routeur port 1
00 00 00 00 00 01	Adresse MAC du terminal A
0800	Type de contenu – protocole IP
…..	datagramme IP
C0 A8 0A 19	Adresse source – Terminal A 192.168.10.25
C0 A8 14 7B	Adresse cible – Terminal B 192.168.20.123
…..	

Nous notons que les adresses MAC sont des adresses locales du sous-réseau alors que les adresses IP sont des adresses de bout en bout.

Le routeur relaie les trames reçues par le terminal A sur le port 1 vers la destination, le terminal B par le port 2. Si le routeur ne connaît pas l'adresse MAC du terminal B, il stocke en mémoire les données et émet une demande ARP propre à cette adresse. Le reste de la procédure est identique à ce que nous avons vu précédemment.

10.2.3 ARP mandataire (Proxy)

La fonction ARP mandataire permet à un routeur de réagir à une demande ARP à la place d'une station se trouvant dans un autre sous-réseau. La station source ignore que le routeur prend la place de la station cible et que cette station se trouve de l'autre côté du routeur dans un sous-réseau autre que celui où elle se trouve. Cette situation survient lorsque la station source se base sur le masque par défaut qui est sous entendu dans les classes d'adresses et que, la destination qui se trouve dans un autre sous-réseau fait appel à la même classe d'adresse de réseau avec toutefois, un masque distinct.

La compréhension de cette situation n'en sera que mieux illustrée par l'exemple que nous allons en donner ci-après. La figure 10.5 illustre cette situation. Nous avons d'un côté du routeur une station A avec une adresse de classe B soit : 130.140.10.15. Cette station a recours à l'adressage classique par classe et pense donc que le masque est 255.255.0.0. La station B a une adresse 130.140.25.14/24 ce qui la place dans un sous-réseau distinct de la station A. Par contre, la station A pense que la station B se trouve dans le même sous-réseau 130.140.0.0.

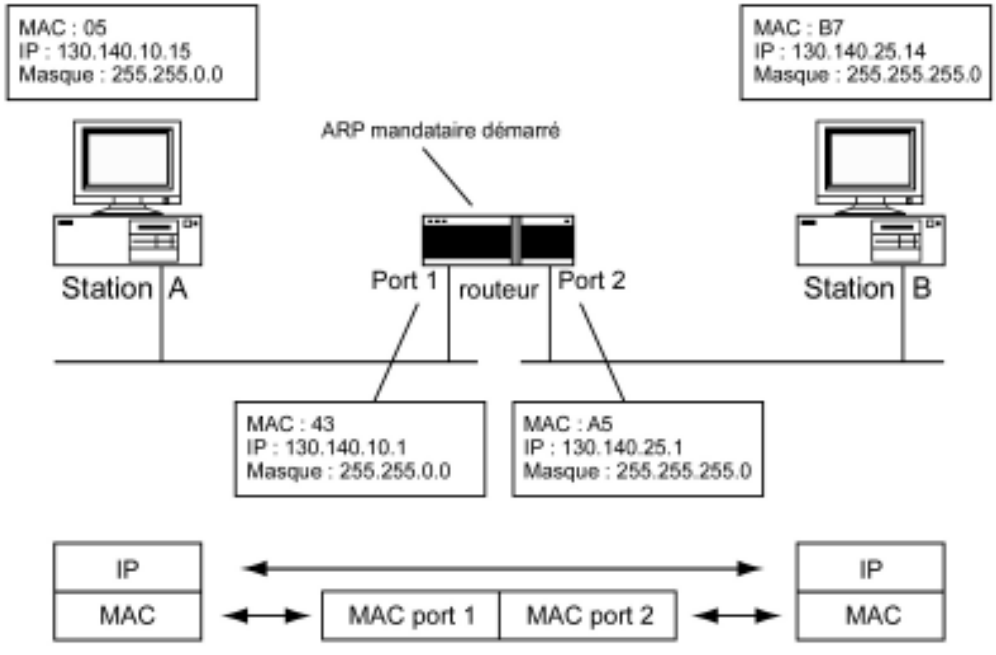


Figure 10.5 Utilisation d'un ARP mandataire

La station A émet une demande ARP à la station B si elle ne dispose pas de son adresse physique. La station A n'utilise donc pas l'adresse MAC du routeur dans le champ *adresse destination* de sa trame. Dans une telle situation il est clair que la station B ne recevra jamais la demande ARP émise par la station A et de ce fait, elle n'y répondra pas.

Si la demande ARP de la station A est reçue au port 1 du routeur, ce dernier qui aurait la fonction de ARP mandataire y répondra à la place de la station B sans toutefois donner d'indication à la station A du remplacement. Le routeur transmettra son adresse MAC en guise de réponse laissant croire qu'il s'agit de l'adresse physique de la station B. La suite de la procédure est en tous points semblable à ce qui a été décrit dans le cas normal.

Il existe une autre situation où la demande d'un ARP mandataire s'avère nécessaire. Il s'agit du cas où on fait appel au serveur NAT. La demande d'un ARP mandataire doit toujours être activée afin de permettre à toute demande d'adresse physique émanant du réseau de recevoir une réponse du serveur NAT.

Dans le cas de la figure 10.6, le serveur NAT réassigne l'adresse 195.148.147.26 à la station 130.140.55.18. Ceci est accompli par souci de sécurité. Le réseau peut adresser cette pseudo adresse et le routeur émet alors une demande ARP pour

obtenir son adresse MAC. Si le serveur NAT n'activait pas la demande d'un ARP mandataire en regard du port connecté au routeur, aucune réponse ne parviendrait au routeur.

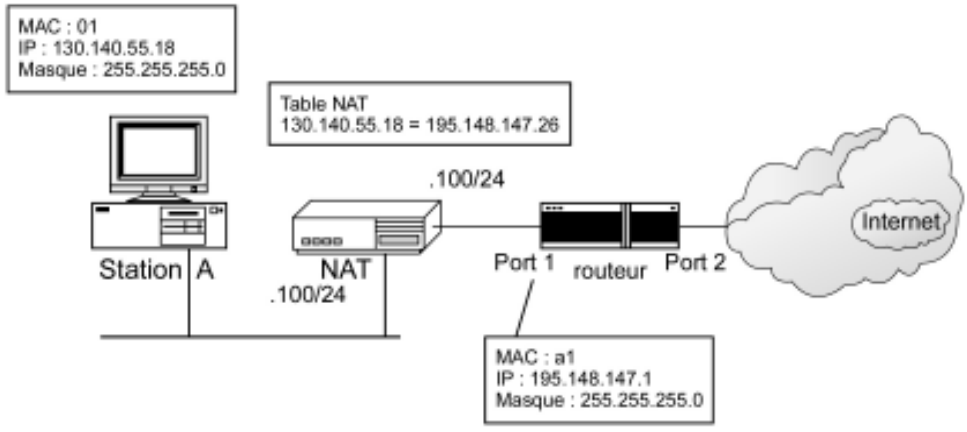


Figure 10.6 Demande d'un ARP mandataire sur le serveur NAT

Le serveur NAT active la demande d'un ARP mandataire et répond au routeur avec son adresse MAC à la place du terminal. Le serveur NAT pourrait être intégré au routeur. Cela permet de résoudre le problème.

10.2.4 ARP gratuit

La fonction ARP gratuit est utilisée par une station pour vérifier sa propre adresse. La station qui transmet la trame ARP gratuit qui est une simple demande ARP à la recherche de sa propre adresse MAC, n'attend pas de réponse. Si elle en reçoit une, cela nous signale qu'une autre station a recours à la même adresse et il y a de ce fait conflit d'adresse. Le recours au Protocole ARP gratuit est une méthodologie permettant d'identifier un problème potentiel.

Le Protocole ARP gratuit permet également de mettre à jour les tables ARP des autres stations du sous-réseau qui le reçoit. Soulignons que, dans le cas du Protocole ARP, quoique étant une demande émise pour l'obtention de l'adresse MAC, la station qui la transmet émet autant ses adresses MAC que IP.

10.3 RARP

Il existe une autre procédure qui permet à une station d'obtenir son adresse IP à partir de son adresse MAC. La procédure RARP (*Reverse ARP*) réalise une telle

opération. Le paquet RARP est le même que celui du ARP à une exception : dans le cas d'une demande RARP le code d'opération est égal à 3 alors que dans le cas d'une réponse il est égal à 4.

Pour garantir la procédure RARP il faut disposer d'un serveur dans le sous-réseau qui garde à jour une base de données incluant les adresses correspondantes MAC et IP.

La procédure RARP est surtout pratiquée par des stations sans disque. Lors de l'initialisation de la station la demande RARP est émise.

La demande RARP est codée conformément à la figure 10.3. L'adresse source MAC est connue. L'adresse source IP est demandée. De ce fait elle est mise à 0 dans le paquet de la demande. L'adresse de destination MAC se lit comme suit : FF FF FF FF FF FF en vue de pouvoir diffuser la demande vers tous les serveurs.

De nos jours, la procédure RARP est moins utilisée. Le protocole DHCP (*Dynamic Host Configuration Protocol*) la remplace assignant à la station une adresse IP. Lorsque nous traiterons du Protocole DHCP nous constaterons qu'il est tout à la fois plus pratique et plus complet que le RARP.

10.4 ICMP (*Internet Control Message Protocol*)

Comme nous l'avons vu (cf. chapitre 9), le protocole IP n'est pas fiable. Les datagrammes peuvent se perdre et il n'y a aucune disposition de prévue par ce protocole pour en aviser la station source. Le protocole ICMP a été élaboré et conçu pour combler cette lacune. Les routeurs ou hôtes font appel au protocole ICMP pour prévenir la source des problèmes rencontrés. Le protocole de contrôle de messages Internet ICMP fait partie de la couche réseau. Ce protocole permet de prévenir la source des erreurs ou des conditions particulières causées par le réseau Internet. Les messages sont activés par la le protocole IP. Le protocole ICMP procède à l'encapsulation de ses messages dans des datagrammes IP.

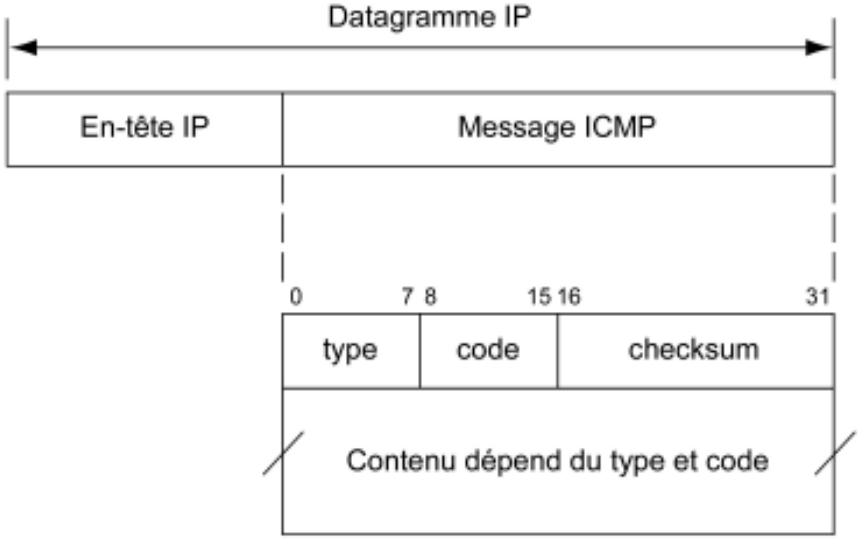


Figure 10.7 Format du message ICMP

Le ICMP utilisé avec la version 4 du protocole IP est encapsulé dans un datagramme IP dont certains champs de l'en-tête IP sont codés comme suit :

Version = 4
Type de service = 0
Protocole = 1 (indique que le contenu est un message ICMP)

Le format de ICMP est illustré dans la figure 10.7. Le premier champ du message ICMP est le type du message. C'est ce champ qui détermine le format du contenu du message.

Le message ICMP ne revêt son plein sens que lorsque le type et le code de ce message sont clairement établis. Le code, qui occupe le deuxième rang parmi les champs du message ICMP, fournit de plus amples détails sur la nature du message.

Nous constatons que le troisième champ est la somme de contrôle de 16 bits qui est calculée sur le contenu total du message ICMP en commençant par le champ type. Il sert de contrôle d'erreurs.

Il existe plusieurs types de messages ICMP. Le tableau 10.1 ci-dessous dresse la liste de toutes celles qui sont définies.

Tableau 10.1 Types des messages ICMP

Type du message	Description
0	Réponse écho
3	Destination inaccessible
4	Arrosage de la source
5	Redirection de route
8	Demande d'écho
11	Temps de Datagramme écoulé
12	Problème de paramètres du Datagramme
13	Demande d'estampille
14	Réponse d'estampille
15	Demande d'information
16	Réponse d'information
17	Demande de masque d'adresse
18	Réponse de masque d'adresse

Un message ICMP peut être une demande ou un message d'erreur. Le message d'erreur ICMP n'est jamais généré en réponse à :

1- un autre message d'erreur ICMP.
2- un datagramme destiné à une adresse de diffusion IP ou une adresse de multi distribution IP.
3- un datagramme transmis en diffusion de la couche Liaison.
4- un fragment autre que le premier.
5- un datagramme dont l'adresse source ne définit pas un seul hôte.

Le tableau 10.2 nous rapporte l'exemple de quelques messages ICMP. Le type définit le message mais le code en indique certaines particularités. Certains de ces messages n'ont qu'une seule combinaison de type et de code. Ainsi, par exemple, le message «Réponse écho» est de type et de code 0. D'autre part, certains des autres messages comportent plusieurs codes pour un seul type. L'exemple du message «Destination inaccessible» en est un cas. Lorsque le type est de 3 nous pouvons en déduire que le message est « Destination inaccessible ». Lorsque le code est à 0, ceci nous indique que le réseau est « inaccessible ». Lorsque ce code est à 1 cela signifie que l'hôte n'est pas accessible.

Tableau 10.2 Type et code de ICMP

Type	Code	Description	Demande	Erreur
0	0	Réponse écho	*	
3		Destination inaccessible		*
	0	réseau		*
	1	hôte		*
	2	protocole		*
	3	port		*
	…..			
5	0	redirection de route		*
9	0	Annonce de routeur	*	
11		Temps écoulé		*
13	0	Demande d'estampille		*
14	1	Réponse d'estampille	*	

10.4.1 Réponse écho (Type 0) / Demande écho (Type 8)- Commande Ping

Autant la demande que la réponse d'écho sont utilisées pour examiner le lien d'une communication entre deux points donnés. En règle générale cette opération est amorcée par la commande PING à partir de la station source qui désire procéder au test. La station réceptrice, si disponible, doit répondre à cette commande PING en lui renvoyant une réponse écho.

Un exemple d'opération PING se trouve sur un système d'exploitation du type DOS :

C:\>ping 142.137.26.200 déclenche la demande écho

Après un bref délai la réponse écho est reçue et l'affichage sur la station qui en a fait la demande se lit comme suit :

Reply from 142.137.26.200:bytes-32 time<10ms TTL=128
Reply from 142.137.26.200:bytes-32 time<10ms TTL=128
Reply from 142.137.26.200:bytes-32 time<10ms TTL=128
Reply from 142.137.26.200:bytes-32 time<10ms TTL=128

La station source qui déclenche la fonction PING transmet la demande d'écho avec des données optionnelles et elle chronomètre le temps de réponse. Une fois

la réponse reçue, la fonction PING affiche l'adresse de la station sollicitée, la longueur des données optionnelles (32 octets), l'indication à l'effet que la réponse est parvenue à moins de 10 millisecondes après l'envoi de la commande et la valeur de la Durée de Vie du datagramme IP (*Time To Live* : *TTL*).

Il importe de noter qu'une opération ARP est amorcée avant la demande écho ICMP. Le code du datagramme IP ainsi que ceux du message ICMP pour la demande et la réponse d'écho sont :

Couche liaison – (Data Link Control (DLC))

00 00 0C 07 AC 10	adresse MAC Destination
00 08 A1 01 A6 CD	adresse MAC Source
08 00	EtherType: IP
Protocole Internet (IP)	
45	version et longueur de l'en-tête
0100	Version 4 : ....
0101	20 octets – longueur de l'en-tête
0x00	Catégorie du service
00 3C	longueur totale - 60 octets
B4 8D	Identification - 46221
00 00	déplacement de fragment
80	durée de vie TTL = 128
01	Identification du Protocole - ICMP
XX XX	Checksum : Somme de contrôle
8E 89 10 D8	adresse source 142.137.16.216
8E 89 1A C8	adresse destination 142.137.26.200
Internet Control Message Protocol (ICMP)	
8	Catégorie: demande écho
0	Code: non utilisé
XX XX	CheckSum: Somme de contrôle (Correct)
02 00	Identificateur - 512
45 03	numéro de séquence: 17667
Data/FCS	
61 62 63 64 65 66	données optionnelles
67 68 69 6A 6B 6C 6D 6E 6F 70 71 72 73 74 75 76	
77 61 62 63 64 65 66 67 68 69	
XX XX XX XX	FCS : Champ de séquence de contrôle de trame.

La station réceptrice renvoie un message réponse écho ICMP. Le code de la réponse écho se lit comme suit :

Couche liaison – (Data Link Control (DLC))

00 08 A1 01 A6 CD	adresse MAC destination
00 00 0C 07 AC 10	adresse MAC source
08 00	EtherType: IP

Protocole Internet (IP)

45	version et longueur de l'en-tête
0100	Version 4 : ....
0101	20 octets – longueur de l'en-tête
00	Catégorie du service
00 3C	longueur totale
B4 8D	Identification
00 00	déplacement de fragment
FE	Durée de Vie
01	identificateur de protocole
XX XX	Checksum : Somme de contrôle
8E 89 1C C8	adresse source 142.137.26.200
8E 89 10 D8	adresse destination 142.137.16.216

Internet Control Message Protocol (ICMP)

00	Catégorie : réponse écho
00	Code
XX XX	Checksum : Somme de contrôle
02 00	identificateur
45 03	numéro de séquence

Data/FCS

61 62 63 64 65 66	données optionnelles
67 68 69 6A 6B 6C 6D 6E 6F 70 71 72 73 74 75 76	
77 61 62 63 64 65 66 67 68 69	
XX XX XX XX	FCS : séquence de contrôle de trame.

10.4.2 Destination inaccessible (Type 3)

Ce type de message, illustré dans la figure 10.8, est transmis pour faire état de l'inaccessibilité du terminal de destination dans les cas suivants :

- Lorsqu'un routeur ne sait comment atteindre la destination.
- Lorsqu'un protocole ou application n'est pas actif.
- Lorsqu'un datagramme assigne une route instable.
- Lorsqu'une route doit fragmenter la grandeur d'un datagramme mais qu'elle ne peut le faire parce que le fanion « ne pas fragmenter » est à 1.

Dans le cas où il y a « réseau inaccessible », le message ICMP de type 3 inclut le code 0. Ce message ICMP est transmis par le routeur lorsque ce dernier ne reconnaît pas, dans l'adresse du terminal de destination du datagramme à transmettre, une route valide ou un choix de route.

Dans le cas du message « hôte inaccessible », le code du message ICMP de type 3 se trouve être mis à 1. Cela nous signale que, bien que le routeur désigne une route acceptable vers le terminal de destination il ne peut établir de communication avec cet hôte.

Un exemple de message de type 3 survient lorsque la commande PING est entamée et que la station de destination ne répond pas. Dans ce cas, la commande de « demande écho » est transmise mais, étant donné que la station ne répond pas, le routeur transmet un message ICMP de type 3 avec un code mis à 1.

La représentation de cet exemple est :

Début de demande écho—
C:\>ping 142.137.26.140 amorce de la demande écho
Début de la réponse écho –

Pinging 142.137.26.140 with 32 bytes of data:
(Transmission de messages ICMP type 3 code 1)
Destination host unreachable.
Destination host unreachable.
Destination host unreachable.
Destination host unreachable.

Puisqu'il n'est pas possible de joindre le terminal hôte le message ICMP de type 3 et de code 1 est transmis.

Type (3)	Code (1)	Checksum
Non utilisé (doit être 0)		
En-tête IP incluant les options plus les premiers 8 octets de données du datagramme rejeté.		

Figure 10.8 Message « destination inaccessible »

Si le routeur achemine un datagramme vers sa destination et peut d'une part établir la route et de l'autre communiquer avec les stations, un problème pourrait empêcher le terminal source d'établir une communication avec le terminal de destination. Ce cas survient lorsque le service demandé par le système hôte

source à celui de destination ne peut en aucun cas être activé. De ce fait, et en raison de l'absence de port, ce service devient inaccessible. Ainsi, la station du système de destination transmet le message ICMP de type 3 et de code 3 ce qui signifie : « Port inaccessible ».

10.4.3 Arrosage de la source (Source Quench) (Type 4)

Ce message forme un contrôle de flux de base pour que le datagrammme parvienne à bon port. Lorsque les datagrammes se succèdent trop rapidement à une station réceptrice pour y être traités, ils sont mis de côté. Pendant toute la durée de ce processus, un message de type 4 est adressé à la station source pour chacun des datagrammes détruits. Ce message ICMP est une demande visant à modérer le taux de transmission des datagrammes. Malheureusement il n'existe aucun mécanisme permettant d'augmenter le taux de transmission si le récepteur le permettait.

10.4.4 Changement de route (Route Redirect) (Type 5)

Les informations de routage sont périodiquement échangées pour que nous puissions nous adapter aux changements du réseau et pour que la table de routage soit constamment tenue à jour. Lorsqu'un routeur détecte un hôte qui empreinte une route qui ne figure pas dans ses options, celui-ci émet un message ICMP de type 5 au terminal source tout en continuant de transmettre le datagramme vers le réseau du terminal de destination par la voie choisie. Ceci nous montre clairement que les messages ICMP de type 5 ne peuvent être transmis qu'aux systèmes hôtes directement connectés au réseau du routeur qui émet le message.

10.4.5 Durée de vie écoulée d'un datagramme (Datagram Time Exceeded) (Type 11)

Un routeur transmet un message ICMP de type 11 lorsqu'il doit détruire un datagramme dont la durée de vie est écoulée soit, lorsque le routeur reçoit un datagramme dont le TTL a atteint 0. Le routeur le détruit et émet un message ICMP de type 11.

10.4.6 Problème de paramètre d'un datagramme (Type 12)

Le message ICMP de type 12 (*Datagram parameter problem*) est transmis lorsque l'en-tête du datagramme ne peut être traité. Le datagramme est alors détruit et le message ICMP de type 12 est émis.

10.4.7 Demande et réponse d'estampille (Type 13)

La demande d'estampille ICMP (type 13), illustrée dans la figure 10.9, permet à un système donné d'en questionner un autre en regard du temps courant en millisecondes depuis minuit temps universel UTC (*Universal Time Coordinated*).

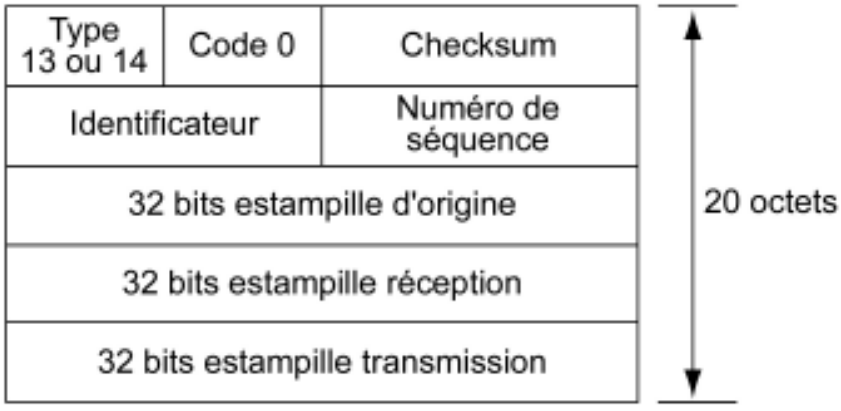


Figure 10.9 Demande ou réponse d'estampille

Le champ de l'« estampille d'origine » du message de demande d'estampille doit être rempli. Par contre, c'est au récepteur qu'il incombe de compléter le champ « estampille de réception » avec l'heure au moment de la réception du message de demande et inscrit le temps au moment de la transmission dans le champ « estampille de transmission », le tout figurant dans le message réponse d'estampille.

10.4.8 Calcul de la durée de rotation RTT (Round Trip Time)

La durée de rotation RTT (*Round Trip Time*) est égale au temps de réception de la réponse moins celui de la transmission de la demande (figure 10.10). La *différence* est définie comme étant le contenu du champ « estampille de réception » moins le contenu du champ « estampille d'origine ».

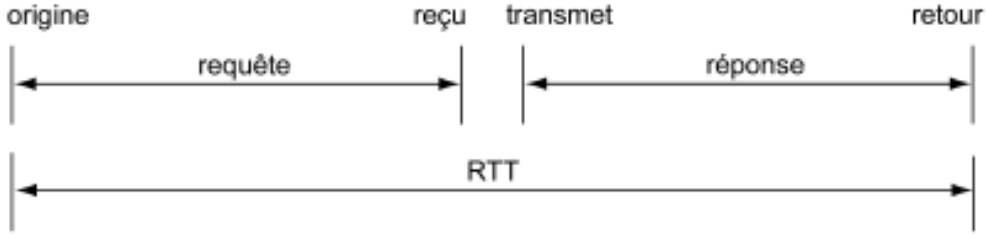


Figure 10.10 Représentation du RTT

L'horloge du transmetteur peut être ajustée en prenant la *différence* moins la moitié de la durée de rotation (RTT). Cet ajustement est pratiquement égal à 0 attendu que le temps de traitement est infime par rapport au temps de transmission. Soulignons cependant que la moitié de RTT est réservée à la transmission de la demande et que l'autre moitié l'est pour la transmission de la réponse. Ainsi, si l'opération précédente nous donne 0, cela revient à dire que le transmetteur atteste que son horloge est identique à celle du récepteur. Toutefois, il y a certains cas où la différence peut être négative, nommément lorsque l'horloge du récepteur accuse un retard par rapport à celle du transmetteur.

Le calcul du délai causé par le réseau peut aussi être jaugé. Ainsi, lorsque la station émettrice envoie une demande d'estampille en y indiquant le moment le message est transmis. La station réceptrice ajoute à sa réponse d'estampille le temps de réception. Le délai du réseau peut être calculé tout en ajustant leur horloge interne.

10.4.9 Demande et réponse d'information, (Types 15 et 16)

Une alternative au RARP tel que décrit précédemment repose sur le recours au message ICMP des Types 15 et 16. Une station peut demander son adresse Internet (elle ne connaît pas l'adresse IP qui lui a été assignée) en adressant un message ICMP de type 15 : demande d'information. Celle-ci peut être retournée dans un message ICMP de type 16 : réponse d'information. Le transmetteur indique alors dans son champ de données la partie réseau de l'adresse et le récepteur lui renvoie la réponse en remplissant la partie hôte de son adresse.

10.4.10 Demande et réponse de masque d'adresse, (Types 17 et 18)

Tout comme dans le cas des messages de demande et de réponse d'information, les stations peuvent transmettre des messages ICMP de type 17 pour demander quel est le masque de sous-réseau du réseau auquel elles sont connectées et le message ICMP de type 18 pour pouvoir leur envoyer la réponse.

10.5 Le protocole DHCP

Chaque station du réseau doit avoir son adresse IP pour communiquer. Sans cette adresse le protocole IP ne dispose d'aucun moyen pour fonctionner. L'assignation des adresses IP peut se faire manuellement. Cela peut susciter d'énormes problèmes, surtout lorsqu'il s'agit de gérer un grand nombre de

stations. Dans ce cas-ci, la gestion des adresses se fait par distribution. Il nous faut donc également contrôler cette attribution afin de ne pas donner lieu à un duplicata d'adresses.

Le protocole DHCP (*Dynamic Host Configuration Protocol*) est un protocole qui permet de procéder à la configuration dynamique des adresses IP. Il se fonde sur le modèle client serveur. La description de ce protocole figure essentiellement dans le RFC 2131. En amorçant l'application du DHCP sur sa station, et en supposant que le serveur dispose aussi d'une application serveur DHCP de démarrage, l'utilisateur peut être certain que son terminal sera rapidement configuré et fonctionner comme station d'accès à Internet.

Le protocole DHCP comporte des mécanismes d'assignation d'adresses. Celles-ci peuvent l'être automatiquement. Dans ce cas, l'adresse est attribuée en permanence au système hôte. Les adresses peuvent être réparties de façon dynamique. Ce protocole allouera, pendant un certain temps, une adresse IP à un client. L'adresse peut être adaptée manuellement par l'administrateur.
De la sorte, le serveur DHCP sera celui qui assignera les adresses aux clients. Plusieurs serveurs DHCP peuvent cohabiter dans un même réseau.

Le protocole DHCP se trouve dans la couche Application. Dans la hiérarchie, il est placé au-dessus du protocole de transport UDP. Ainsi, le client DHCP utilise le port UDP 68 alors que le serveur lui, utilise le port UDP 67.

10.5.1 Message DHCP

Le message DHCP est encapsulé dans un segment UDP. Les champs de l'en-tête DHCP tels qu'illustrés à la figure 10.11 sont identifiés comme suit et ils seront appliqués tels qu'indiqués dans le contexte où ils sont utilisés.

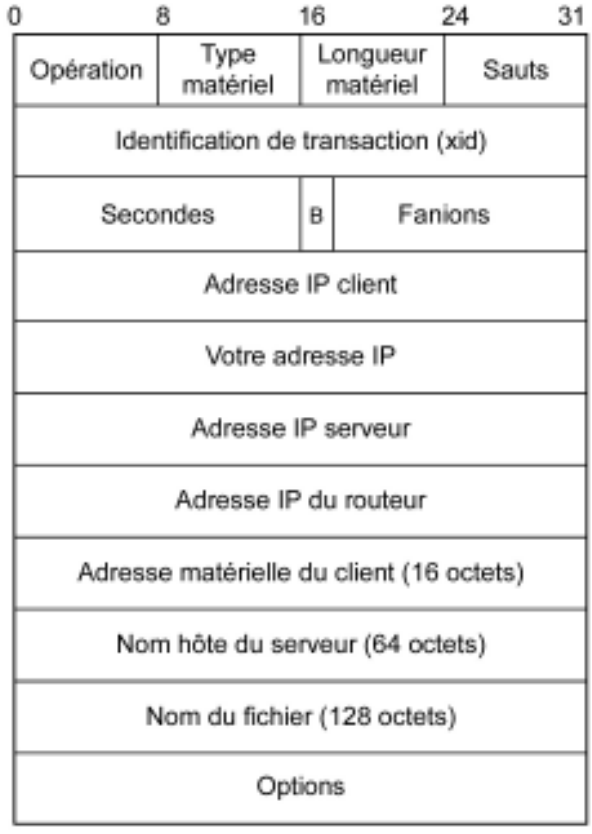


Figure 10.11 Format du message du protocole DHCP

Opération – Champ indiquant la nature de l'opération déclenchée par le message. Il peut prendre la forme soit d'une demande : boot (1) soit d'une réponse : boot (2).

Type de matériel – genre d'adresse de matériel (Ethernet ou anneau à jeton).

Longueur d'adresse matérielle – Longueur de l'adresse MAC.

Sauts – Normalement à 0 sauf lorsqu' utilisés par l'agent relais.

Identification de la transaction –Numéro aléatoire utilisé par la demande et par la réponse échangée entre le client et le serveur pour une même transaction.

Secondes – Secondes écoulées depuis la transmission de la demande du client.

Fanions – Indiquent si le client peut recevoir seulement des trames à diffusion individuelle unicasts (0) ou bien des trames multi distributions (1). Seul le bit le moins significatif est utilisé. Les 7 autres bits sont à 0.

Adresse IP client – Ce champ est à 0. Une fois la transaction terminée, le client y indiquera la validité de l'adresse IP client.

Votre adresse IP – Ce champ est aussi mis à 0 par le client durant la demande du client. Le serveur informe le client de son adresse IP lorsque celle-ci lui est fournie.

Adresse IP serveur – L'adresse IP du prochain serveur que doit utiliser le client pour sa configuration.

Adresse IP du routeur – L'adresse du relais susceptible d'être utilisée peut figurer dans ce champ.

Adresse matériel du client – Indication de l'adresse MAC du client.

Nom hôte du serveur – Nom du prochain serveur susceptible d'être utilisé par le client pendant la configuration.

Nom du fichier – Nom du fichier pouvant être demandé par le client au prochain serveur (pouvant contenir des options DHCP ou un programme de démarrage par exemple).

Options – La longueur du champ Options DHCP est variable. Cette section comprend une séquence d'options comportant des informations additionnelles comme le genre de message, le serveur DNS et bien d'autres encore. Nous analyserons certaines de ces options.

10.5.2 Options

Les options utilisées dans les messages DHCP peuvent être de longueur fixe ou variable. Toutes les options de longueur variable ont trois champs : le code de l'option, la longueur du champ suivant et la valeur.

Les quatre premiers octets du champ option sont connus sous le nom de « *magic cookie* ». Leur valeurs décimales sont : 99, 130, 83 et 99. Ceci définit le format de la section Options.

Nous retrouvons une option principale qui est « *type de message DHCP* ». Cette option nous permet de savoir quel est le genre du message. Le code de cette option est 53. Les valeurs d'une longueur de 1 octet sont réservées à certains messages comme :

DHCPDISCOVER	1
DHCPOFFER	2
DHCPREQUEST	3
DHCPDECLINE	4
DHCPACK	5
DHCPNAK	6
DHCPRELEASE	7

Une autre option utile est « *adresse IP demandée* ». Le code qui lui est attribué est égal à 50. Les valeurs d'une longueur de quatre octets contiennent l'adresse IP qui est assignée par le serveur.

Lorsque le serveur prescrit une adresse IP au client, il ne la lui réserve que pour un certain temps. Il établit un bail qu'il transmet aussi au client. L'option « *Temps de bail de l'adresse IP* » dont le code est égal à 51 renferme le temps en jours, heures, minutes, secondes, l'adresse est réservée au client. Ce dernier doit normalement demander un renouvellement de son adresse une fois qu'il atteint 50% de la fin de son bail. Si vers la fin du bail le client ne peut avoir de renouvellement, il devra renégocier une nouvelle adresse en adressant le message DHCPDISCOVER.

Pour connaître le temps de bail de l'adresse IP la commande suivante peut être utilisée sous DOS :

C:\>ipconfig /all

Il existe plusieurs autres options figurant dans les RFC 1048 et 2132. Nous ne mentionnerons qu'une autre de longueur fixe : l'option *Fin* qui ne comprend que l'octet d'une valeur égale à 255 et qui indique la fin du champ option.

10.5.3 Allocation dynamique des adresses réseau

Le mécanisme d'allocation dynamique des adresses réseau est simple. Il consiste en un échange de messages entre client et serveur. Dans sa forme la plus simple, l'échange se fait avec quatre messages tel qu'illustré à la figure 10.12. Le client et le serveur sont tous deux dans le même sous-réseau.

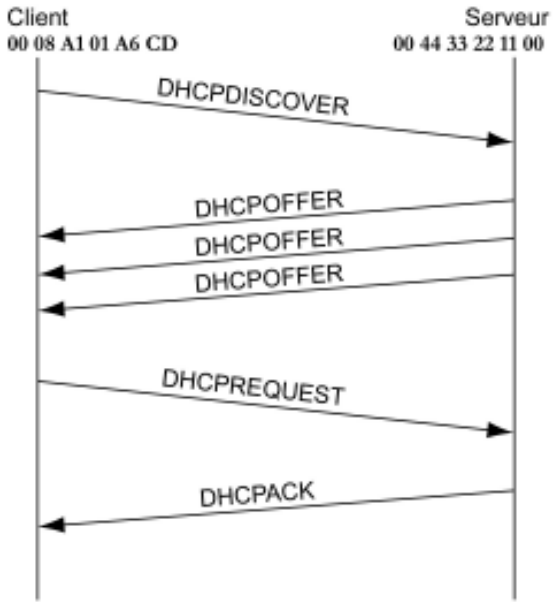


Figure 10.12 Interaction entre client et serveur DHCP

Le client qui a besoin d'une adresse IP, transmet un message DHCP au serveur pour demander qu'on lui en assigne une pour un certain temps. C'est une demande de bail. Ceci se fait par le biais de la transmission du message DHCPDISCOVER., Attendu que le client n'a pas d'adresse IP, il met 0.0.0.0 dans le champ de l'adresse source et inscrira 255.255.255.255 comme adresse de diffusion dans le champ de l'adresse destination du datagramme IP contenant le message DHCP. La trame suivante illustre les adresses utilisées dans ce cas de figure :

<u>Message DHCPDISCOVER</u>

Couche liaison – (Data Link Control (DLC))

FF FF FF FF FF FF	adresse MAC Destination
00 08 A1 01 A6 CD	adresse MAC Source
08 00	EtherType : IP

Protocole Internet (IP)

45	version et longueur de l'en-tête
0100....	Version 4 : ....
0101	20 octets – longueur de l'en-tête
00	Catégorie du service
00 4C	longueur totale - 76 octets

B4 8D	Identification - 46221
00 00	déplacement de fragment
80	durée de vie TTL = 128
11	Identification du Protocole – UDP (17)
XX XX	CheckSum : Somme de contrôle
00 00 00 00	adresse source 0.0.0.0
FF FF FF FF	adresse destination 255.255.255.255

Protocole UDP

44	Port source : 68 client
43	Port destination : 67 serveur
38	longueur 56 octets
XX	CheckSum : Somme de contrôle

Message DHCP

01 01 06 00	1er mot du message DHCP
00 34 23 54	Numéro d'id. de la transaction
00 00 00 00	
00 00 00 00	adresse IP client 0.0.0.0
00 00 00 00	Votre adresse IP 0.0.0.0
00 00 00 00	Adresse IP serveur 0.0.0.0
00 00 00 00	Adresse IP du routeur 0.0.0.0
00 08 A1 01 A6 CD	Adresse matériel du client
00 00 00 00	Nom hôte du serveur
00 00 00 00	Nom du fichier

OPTIONS

63 82 53 63	Magic cookie (99 130 83 99)
35 01 01	DHCP type : DHCPDISCOVER
FF 00 00	Fin du champ option

Il peut exister plus d'un serveur DHCP. Ainsi, puisque le message DHCPDISCOVER a été diffusé, tous les serveurs l'ont reçu et tous pourraient répondre par un message DHCPOFFER. Ce message qui est l'offre de bail a pour but de proposer une adresse IP et la valeur des paramètres demandés par le client. Dans le datagramme IP, l'adresse source est celle du serveur alors que l'adresse destination est 255.255.255.255. Ainsi la trame transportant ce message a la forme suivante :

Message DHCPOFFER

Couche liaison – (Data Link Control (DLC))

FF FF FF FF FF FF	adresse MAC Destination
00 44 33 22 11 00	adresse MAC Source – le serveur
08 00	EtherType : IP

Protocole Internet (IP)

45	version et longueur de l'en-tête
0100....	Version 4 : ....
0101	20 octets – longueur de l'en-tête
00	Catégorie du service
00 5C	longueur totale - 92 octets
8C 23	Identification - 35875
00 00	déplacement de fragment
80	durée de vie TTL = 128
11	Identification du Protocole – UDP (17)
XX XX	CheckSum : Somme de contrôle
C0 A8 64 01	adresse source 192.168.100.1 du serveur
FF FF FF FF	adresse destination 255.255.255.255

Protocole UDP

43	Port source : 67 serveur
44	Port destination : 68 client
48	longueur 72 octets
XX	CheckSum : Somme de contrôle

Message DHCP

02 01 06 00	1er mot du message DHCP
00 34 23 54	Numéro d'id. de la transaction
00 00 00 00	
00 00 00 00	adresse IP client 0.0.0.0
C0 A8 64 64	Votre adresse IP 192.168.100.100
00 00 00 00	Adresse IP serveur 0.0.0.0
00 00 00 00	Adresse IP du routeur 0.0.0.0
00 08 A1 01 A6 CD	Adresse matériel du client
00 00 00 00	Nom hôte du serveur
00 00 00 00	Nom du fichier

OPTIONS

63 82 53 63	Magic cookie (99 130 83 99)
35 01 02	DHCP type : DHCPOFFER

36 04 C0 A8 64 01	Id. du serveur DHCP 192.168.100.1
33 04 00 27 8D 00	Temps de bail de l'adresse 30 jrs
3A 04 00 13 C6 80	Temps de renouvellement de bail 15 jrs
3B 04 00 22 47 00	Temps de renouvellement avec nouveau serveur
01 04 FF FF FF 00	Masque de sous-réseau 255.255.255.0
FF	Fin du champ option

Le client reconnaît la réponse à sa demande DHCPDISCOVER en se référant au numéro de transaction qui se trouve dans le message DHCPOFFER. Tous les messages impliqués dans la même interaction comprennent le même numéro d'identification.

Le client peut recevoir plusieurs réponses. Il choisi l'adresse qu'il considère la plus appropriée et transmet au serveur qui la lui a transmise le message sélection de bail DHCPREQUEST. Ce message est en soi une demande de réservation de l'adresse indiquée dans le champ « votre adresse ». Les adresses source et destination du datagramme IP sont respectivement 0.0.0.0 et 255.255.255.255. Par contre, la valeur de l'adresse IP demandée (et proposée lors du message précédent) se trouve dans l'option numéro 50. La trame comprenant ce message est le suivant :

<u>Message DHCPREQUEST</u>

Couche liaison – (Data Link Control (DLC))

FF FF FF FF FF FF	adresse MAC Destination
00 08 A1 01 A6 CD	adresse MAC Source – le client
08 00	EtherType : IP

Protocole Internet (IP)

45	version et longueur de l'en-tête
0100....	Version 4 : ....
0101	20 octets – longueur de l'en-tête
00	Catégorie du service
00 5C	longueur totale - 92 octets
71 1A	Identification - 28954
00 00	déplacement de fragment
80	durée de vie TTL = 128
11	Identification du Protocole – UDP (17)
XX XX	CheckSum : Somme de contrôle
00 00 00 00	adresse source 0.0.0.0
FF FF FF FF	adresse destination 255.255.255.255

Protocole UDP

44	Port source : 68 client
43	Port destination : 67 serveur
5C	longueur 92 octets
XX	CheckSum : Somme de contrôle

Message DHCP

01 01 06 00	1er mot du message DHCP
00 34 23 54	Numéro d'id. de la transaction
00 00 00 00	
00 00 00 00	adresse IP client 0.0.0.0
00 00 00 00	Votre adresse IP 0.0.0.0
00 00 00 00	Adresse IP serveur 0.0.0.0
00 00 00 00	Adresse IP du routeur 0.0.0.0
00 08 A1 01 A6 CD	Adresse matériel du client
00 00 00 00	Nom hôte du serveur
00 00 00 00	Nom du fichier

OPTIONS

63 82 53 63	Magic cookie (99 130 83 99)
36 04 C0 A8 64 01	Id. du serveur DHCP 192.168.100.1
33 04 00 27 8D 00	Temps de bail de l'adresse 30 jrs
3A 04 00 13 C6 80	Temps de renouvellement de bail 15 jrs
3B 04 00 22 47 00	Temps de renouvellement avec nouveau serveur
01 04 FF FF FF 00	Masque de sous-réseau 255.255.255.0
35 01 03	DHCP type : DHCPREQUEST
32 04 C0 A8 64 64	Adresse IP demandée : 192.168.100.100
FF 00 00	Fin du champ option

Le serveur, qui reconnaît l'adresse demandée correspondant à l'adresse proposée du message DHCPOFFER, répond à cette demande en transmettant le message accord de bail DHCPACK. La trame comprenant le message DHCPACK est la suivante :

<u>Message DHCPACK</u>

Couche liaison – (Data Link Control (DLC))

FF FF FF FF FF FF	adresse MAC Destination
00 44 33 22 11 00	adresse MAC Source – le serveur
08 00	EtherType : IP

Protocole Internet (IP)

45	version et longueur de l'en-tête
0100....	Version 4 : ....
0101	20 octets – longueur de l'en-tête
00	Catégorie du service
00 5C	longueur totale - 92 octets
4D 90	Identification - 19856
00 00	déplacement de fragment
80	durée de vie TTL = 128
11	Identification du Protocole – UDP (17)
XX XX	CheckSum : Somme de contrôle
C0 A8 64 01	adresse source 192.168.100.1 du serveur
FF FF FF FF	adresse destination 255.255.255.255

Protocole UDP

43	Port source : 67 serveur
44	Port destination : 68 client
48	longueur 72 octets
XX	CheckSum : Somme de contrôle

Message DHCP

02 01 06 00	1er mot du message DHCP
00 34 23 54	Numéro d'id. de la transaction
00 00 00 00	
00 00 00 00	adresse IP client 0.0.0.0
C0 A8 64 64	Votre adresse IP 192.168.100.100
00 00 00 00	Adresse IP serveur 0.0.0.0
00 00 00 00	Adresse IP du routeur 0.0.0.0
00 08 A1 01 A6 CD	Adresse matériel du client
00 00 00 00	Nom hôte du serveur
00 00 00 00	Nom du fichier

OPTIONS

63 82 53 63	Magic cookie (99 130 83 99)
35 01 04	DHCP type : DHCPACK
36 04 C0 A8 64 01	Id. du serveur DHCP 192.168.100.1
33 04 00 27 8D 00	Temps de bail de l'adresse 30 jrs
3A 04 00 13 C6 80	Temps de renouvellement de bail 15 jrs
3B 04 00 22 47 00	Temps de renouvellement avec nouveau serveur
01 04 FF FF FF 00	Masque de sous-réseau 255.255.255.0
FF	Fin du champ option

Le client se trouve assigné une adresse IP qu'il utilisera dans toutes communications subséquentes comme adresse source des datagrammes qu'il transmet. Les datagrammes qui lui seront adressés auront également cette adresse dans le champ « adresse destination ».

10.5.4 Agent de relais DHCP

Puisque les adresses IP ne peuvent pas être utilisées durant une interaction DHCP, les messages DHCP sont propagés par le biais de la diffusion en empruntant la couche 2. Étant donné que les routeurs n'acheminent pas les diffusions émises par le biais de la couche 2, les messages DHCP ne pourront pas atteindre un serveur potentiel qui se trouverait à l'autre extrémité du routeur (figure 10.13).

Les agents de relais DHCP sont utilisés pour permettre que ce transfert se fasse. Chaque routeur comporte, en principe, l'agent de relais DHCP. Ce dernier reçoit un message DHCP diffusé et le retransmet en unicast au serveur DHCP. La même situation prévaut pour des réponses provenant du serveur.

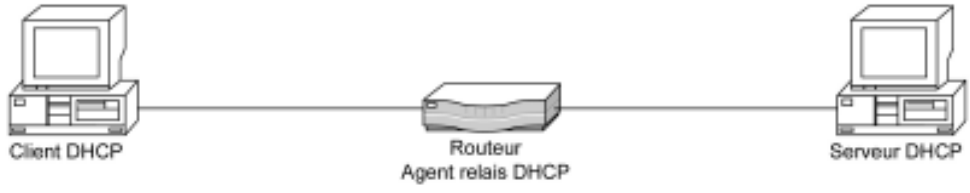


Figure 10.13 Agent de relais DHCP

La figure 10.14 nous permet d'illustrer ce processus. Le client commence par transmettre le message DHCPDISCOVER en ayant recours à la couche Liaison, avec l'adresse MAC de la source et l'adresse MAC de diffusion pour la station de destination. Les adresses IP sont respectivement : 0.0.0.0 et 255.255.255.255 pour la source et pour la station de destination. L'agent de relais DHCP distingue le message DHCPDISCOVER et le relaie au serveur DHCP qui se trouve dans un autre sous-réseau à l'autre extrémité du routeur. L'agent de relais fait alors appel à la couche Liaison, avec son adresse MAC comme adresse source et l'adresse MAC du serveur DHCP comme adresse de destination. En outre, les adresses IP tant source que destination sont respectivement, celles du routeur et du serveur DHCP.

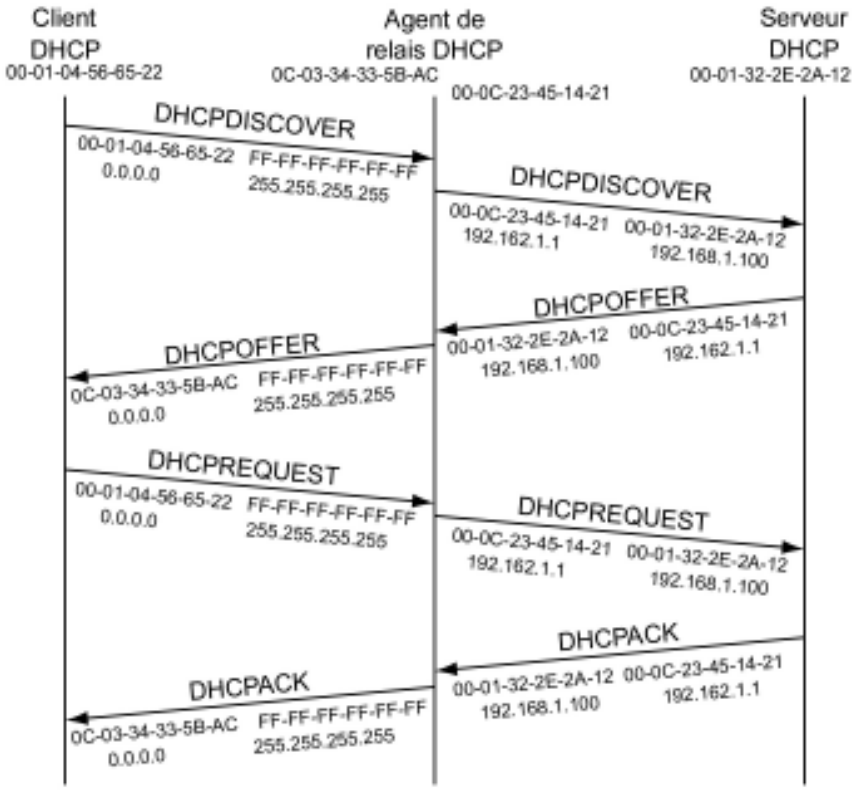


Figure 10.14 Exemple de transaction DHCP avec relais

Lorsque le serveur renvoie le message DHCPOFFER à l'agent de relais, il se sert des mêmes adresses que celles reçues par ce dernier mais en en inversant les sources et les destinations. Les mêmes adresses sont utilisées de part et d'autre de l'agent de relais DHCP pour les messages DHCPREQUEST et DHCPACK.

10.5.5 Autres messages DHCP

En plus des messages décrits précédemment, notamment DHCPDISCOVER, DHCPOFFER, DHCPREQUEST, et DHCPACK, il est possible d'avoir recours à d'autres messages qui pourraient s'avérer être d'une grande utilité dans certains cas. Certains de ces messages sont les suivants :

DHCPNAK – Ce message du serveur au client est utilisé lorsque le message DHCPREQUEST est rejeté.

DHCPDECLINE – Ce message peut être envoyé du client au serveur pour mentionner que l'adresse qui lui a été assignée (par

DHCPACK) est déjà utilisée par un autre client. Le client peut détecter l'existence de cette adresse en effectuant sa vérification par une demande ARP pour l'adresse en question.

DHCPRELEASE – Ce message est adressé par le client au serveur pour l'informer qu'il met fin au bail qui le lie à l'adresse IP.

10.6 Traduction des adresses du réseau – NAT

Dans certains cas, l'adresse utilisée par un terminal ou par un sous-réseau n'est pas celle qui devrait être appliquée dans Internet. Cela peut être une adresse privée ou illégale.

Une fonctionnalité particulière qui se trouve dans un routeur ou un pare-feu peut traduire ou plutôt remplacer l'adresse utilisée par le terminal, par une autre adresse reconnue dans Internet. Cette fonction se nomme « Traduction d'adresses réseau » connue en anglais sous l'acronyme de NAT (*Network Address Translation*).

Nous allons décrire quelques cas où la présence de la fonctionnalité NAT s'avère d'une très grande utilité. La première est le cas d'un abonné qui, auprès d'un fournisseur de réseau, n'a qu'un seul accès et donc une seule adresse IP, et qui veut permettre à un petit nombre de terminaux d'avoir accès simultanément à Internet par le biais de cet accès unique. C'est le cas type d'un simple abonné se trouvant soit à domicile soit dans une petite entreprise. Le fournisseur de réseau accorde l'accès à Internet par le biais d'un modem et il attribue une adresse IP à l'abonné. Tout comme l'illustre la figure 10.15, un routeur peut être installé au modem et permettre à un certain nombre de terminaux de s'y connecter point à point. Si les terminaux sont en mode DHCP, le routeur pourra attribuer une adresse interne propre à chaque terminal.

Un terminal qui a accès à Internet transmettra ses datagrammes avec l'adresse source qui lui aura été allouée, par exemple 192.168.100.100. Cette adresse étant privée, sera rejetée par Internet si le datagramme la comprenant avait été directement transmis. Le routeur comprenant la fonctionnalité NAT, remplace l'adresse source privée sur tous les datagrammes provenant du terminal avec l'adresse 192.168.100.100 par l'adresse que lui a assigné le fournisseur de réseau, notamment : 142.135.12.15.

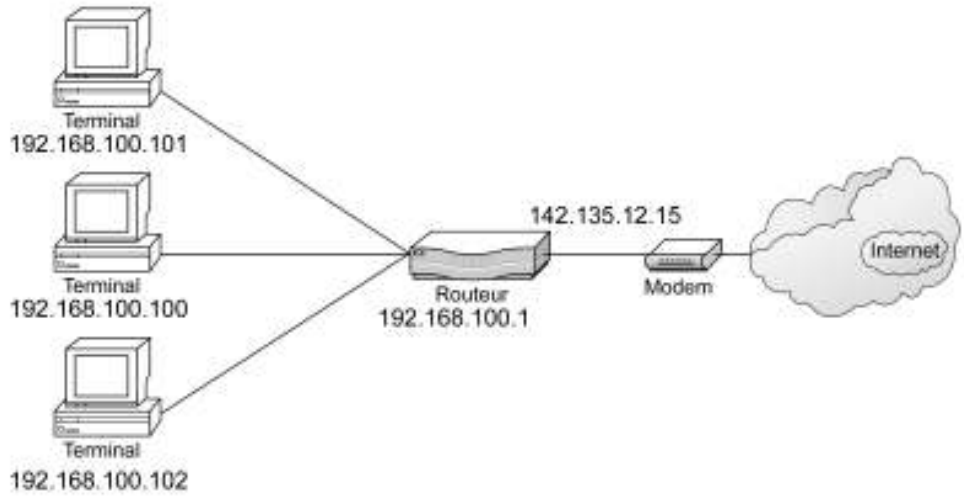


Figure 10.15 Traduction d'une adresse privée en adresse publique

L'adresse de destination des datagrammes de réponse provenant du réseau sera 142.135.12.15. Le routeur remplace toutes ces adresses de destination par l'adresse interne suivante : 192.168.100.100 et achemine les datagrammes vers le terminal ayant cette adresse.

Le routeur doit pouvoir desservir tous les terminaux du réseau interne en leur donnant accès à Internet avec une adresse unique publique. Le routeur peut faire la distinction entre le trafic provenant d'un terminal et destiné à n'importe quel terminal du réseau interne.

Dans une plus large mesure, une entreprise pourrait faire appel à la fonctionnalité NAT sur ses routeurs avec d'une part un certain nombre d'adresses publiques et de l'autre tous les terminaux de l'entreprise. Dans ces conditions on a recourt à NAT par simple mesure d'économie sans pour autant devoir en réduire l'accessibilité. Cela s'avère être fondé dans la mesure où le nombre d'adresses publiques est judicieusement choisi pour convenir au taux d'utilisation moyen d'Internet.

Tout comme dans le cas du réseau téléphonique, tous les terminaux d'une entreprise n'ont pas tous accès à Internet au même instant. Il n'est donc pas nécessaire d'avoir autant de liens d'accès directs à Internet que de terminaux. Ceci n'est pas sans rappeler pour beaucoup le principe du PBX ou *Autocommutateur téléphonique de réseau,* du réseau téléphonique d'une entreprise (*Private branch Exchange*). Aussi est-il plus économique d'avoir un nombre plus restreint d'accès à Internet et de ce fait d'adresses IP publiques que de terminaux.

Cependant, cela ne peut être possible que dans la mesure où NAT est activé dans les routeurs qui sont à la frontière du réseau.

Les adresses utilisées dans le réseau interne ne doivent pas nécessairement être des adresses privées. La seule condition est que chaque adresse se doit d'être unique dans le réseau interne.

Un autre cas de l'utilité de NAT survient lorsque l'hôte d'un serveur Web devient très encombré et ne peut suffire pour répondre aux demandes très nombreuses. Plus la demande est grande, plus le serveur ralentit et plus le temps de réponse devient trop long. Un utilisateur qui trouve que le temps de réponse est de quelques secondes considère cela intolérable et annule son accès. Dans ce cas, il faudrait plusieurs hôtes pour répondre à une demande croissante et réduire le temps d'attente. Le problème est que chacun de ces hôtes doit avoir une adresse distincte. Par contre l'adresse du site Web doit être unique. NAT permet de résoudre ce problème en attribuant d'une part une adresse individuelle à chaque hôte et de l'autre en conservant une adresse unique pour son accès.

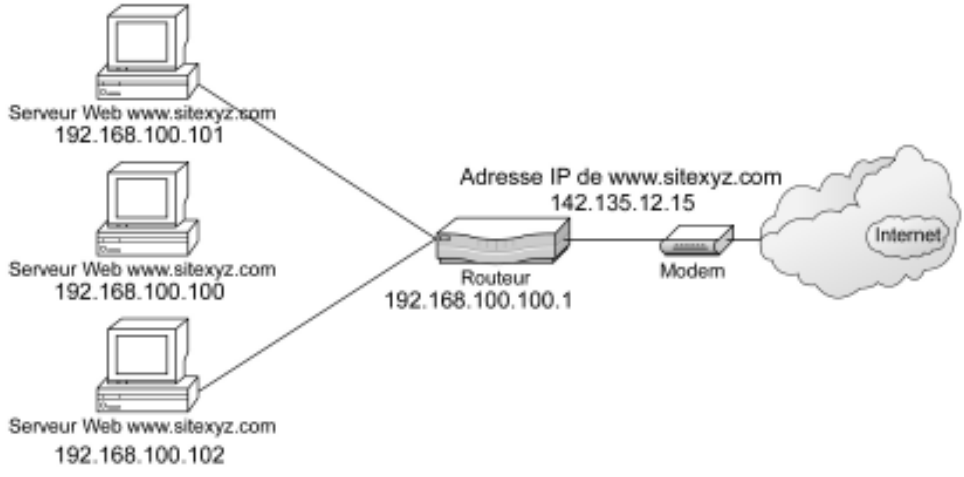


Figure 10.16 Accès serveur Web à plusieurs hôtes

Tel qu'illustré à la figure 10.16, le routeur avec la fonctionnalité NAT peut facilement répondre à la demande et combler ce besoin.

L'utilisation d'adresses privées est chose courante. Les organisations trouvent cela beaucoup plus pratique surtout lorsque le nombre de terminaux augmente. Tout comme l'indique la figure 10.17, il est bien plus simple de privilégier le groupe d'adresses 10.0.0.0 puisqu'il nous permet d'avoir un très grand nombre de terminaux. Il est donc assez courant de voir une double traduction d'adresses

lorsque les réseaux communicants utilisent le même groupe d'adresses. Même si cela n'est pas le cas, cela est vrai pour tous les réseaux communiquant entre eux et utilisant des adresses privées.

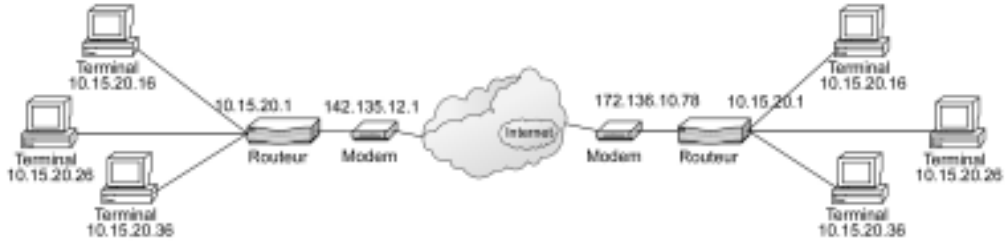


Figure 10.17 Sous-réseaux utilisant les mêmes adresses privées

L'utilisation de NAT contribue à réduire l'utilisation du nombre d'adresses IP sur Internet. Puisque les entreprises utilisent des adresses privées pour leurs terminaux et quelques adresses publiques pour l'accès à leur réseau, l'allocation de nouvelles adresses IP ne suit pas la croissance des terminaux qui utilisent Internet. L'allocation des adresses IP s'en trouve ralentie grâce au recours à NAT.

Avec l'avènement de la version IPv6 certains réseaux privés pourraient devoir transiter par des adresses IPv6 pour des accès externes. Dans de telles circonstances, il faudrait changer toutes les adresses IP de chacun des hôtes de l'organisation. Cependant, grâce a un NAT spécial pouvant traduire l'adresse IPv4 en une adresse IPv6, et vice versa, les réseaux privés basés sur IPv4 n'auront à procéder à aucun changement sauf en ce qui a trait aux équipements soutenant le NAT en question.

10.6.1 Fonctionnement de NAT

Le NAT peut fonctionner de différentes façons; la plus simple étant la traduction d'adresses. L'adresse source du terminal qui est une adresse privée est traduite par une des adresses publiques IP que le routeur garde en mémoire. Ces adresses sont attribuées au sous-réseau. La figure 10.18 illustre une telle traduction. Le terminal 192.168.100.102 transmet un datagramme concernant l'adresse IP destination 142.127.16.3. Ce datagramme est traité par le routeur qui change l'adresse source à 142.135.12.15, une adresse qu'il réserve au sous-réseau. Le routeur conserve cette correspondance dans une table, ce qui va lui permettre d'acheminer vers le terminal les datagrammes qui lui parviennent du réseau.

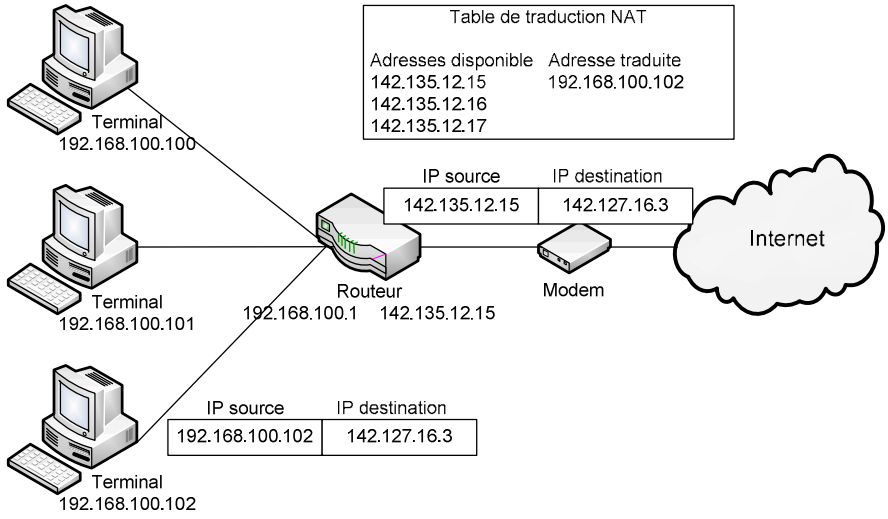


Figure 10.18 Cas de traduction d'adresses une par une

Les adresses assignées au sous-réseau peuvent être permanentes. Dans ce cas, on est en présence d'un NAT statique. Il s'agit de l'assignation d'adresses faite par le fournisseur du réseau.

Les adresses peuvent aussi être assignées de manière dynamique. Le fournisseur peut donc attribuer des adresses au routeur par le biais du DHCP. Une fois les adresses obtenues, le routeur fonctionne de la même façon que précédemment.

Il importe de mentionner que le champ « somme de contrôle (*checksum*) » de l'en-tête IP doit être recalculé attendu que le contenu des champs adresses a changé.

Ce processus est fort simple lorsque nous nous trouvons en présence d'autant d'adresses assignées qu'il y a de terminaux dans le sous-réseau. Comment NAT fonctionne-t-il si le fournisseur de réseau n'attribue qu'une seule adresse et que le routeur dessert plusieurs terminaux? La question peut devenir bien plus spécifique : comment le routeur pourra établir la correspondance entre les datagrammes parvenant au réseau avec ceux réservés à un terminal en particulier? Comment peut-il savoir à quel terminal le datagramme doit parvenir?

Dans ce cas NAT agit différemment. Il est question d'avoir recours au port de la couche Transport, connu sous l'acronyme NAPT (*Network Address Port Translation*) ou PAT (*Port Address Translation*). Tel qu'illustré à la figure 10.19, il n'y a qu'une adresse publique (dans le cas général il y aurait, dans le sous-réseau, moins d'adresses publiques que de terminaux), le routeur doit consulter l'en-tête de la couche Transport et en extraire l'adresse du port utilisée par la source.

L'adresse de départ est traduite par l'adresse que le fournisseur lui assigne et, le port utilisé par la source à l'origine est changé pour un autre port unique au routeur.

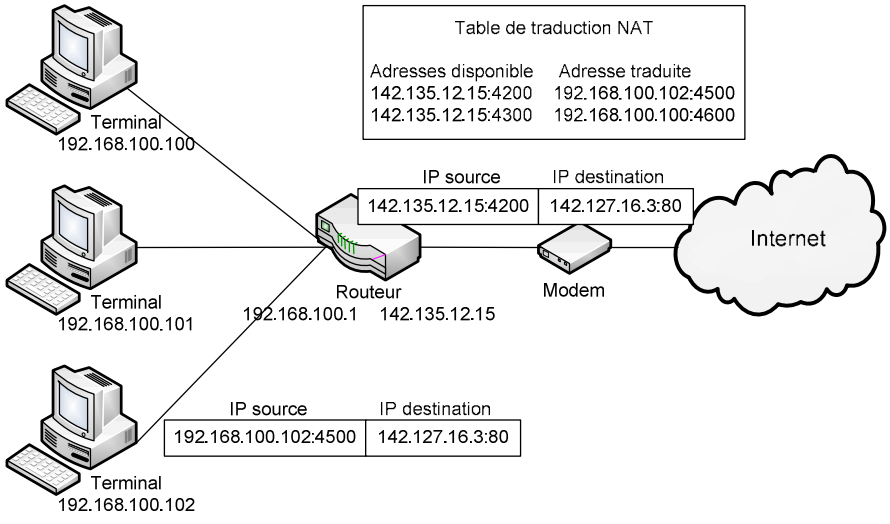


Figure 10.19 Traduction NAT d'une adresse en plusieurs adresses

Ainsi, lorsque le terminal 192.168.100.102 utilisant le port 4500 transmet un datagramme le routeur utilise NAT pour traduire l'adresse 142.135.12.15 et lui alloue le port 4200. Ce mécanisme rend l'échange unique. Si un autre terminal venait à transmettre un datagramme à son tour une nouvelle entrée serait créée et figurerait dans la table de traduction NAT. Si le routeur recevait un datagramme adressé à l'adresse IP 142.135.12.15 et au port 4200, il consulterait au préalable la table de traduction NAT et changerait l'adresse de destination du datagramme lui attribuant l'adresse IP 192.168.100.102 et le port 4500 puis, il transmettrait le tout au terminal.

10.7 Système de noms de domaine DNS

Les adresses IP sont des *adresses* reconnues par les machines. Les humains préfèrent des noms pour les terminaux hôtes. Dans le milieu du protocole TCP/IP, le DNS (*Domain Name System*) désigne une base de données répartie fournissant l'adresse IP et le nom de l'hôte de même que leur interrelation.

Toute application peut faire appel à une fonction de librairie standard pour trouver l'adresse IP correspondant à un nom d'hôte. Inversement, on peut dire qu'une fonction est disponible pour réaliser la recherche à partir d'une adresse IP donnée et trouver le nom qui lui correspond.

L'accès au DNS se fait en passant par le résolveur (*resolver*). De temps à autres, il arrive au résolveur de contacter le serveur de noms de domaine pour obtenir des *adresses* requises lorsque, par exemple, un fureteur demande une page. Le DNS figure dans les RFC 1034 et 1035.

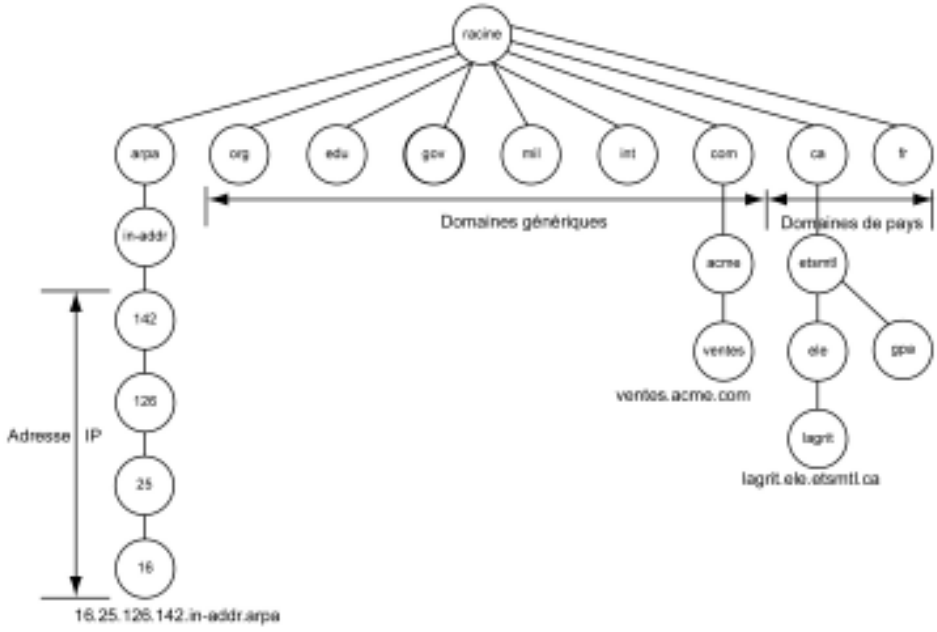


Figure 10.20 Arborescence de domaines

La structure du DNS, partiellement illustrée dans la figure 10.20, est faite en arborescence et comporte une série de noms placés de façon hiérarchique (Cf. RFC 1700). Chaque nœud de l'arbre représente un domaine qui compose en soi, une partie de la base de données de l'ensemble de tous les noms. Hiérarchiquement, le nœud le plus élevé constitue ce que l'on a coutume de désigner par la « racine ». Il ne porte aucun nom précis. Cependant, aux paliers inférieurs, chaque nœud pourrait être relié à un serveur différent. Aussi, le nom de l'*adresse* correspond également au nom de chaque nœud qui se trouve sur une même branche mais dans un ordre inverse. Chacun s'en trouve séparé par un point, tout comme dans l'exemple suivant : *lagrit.ele.etsmtl.ca* ou *ventes.acme.com*. Chaque nom se doit d'être unique.

Les domaines du niveau supérieur (*Top-Level Domain-TLD*) sont subdivisés en trois :

- arpa est un domaine spécial utilisé seulement pour obtenir le nom à partir d'une *adresse*. L'adresse IP est intervertie et suivie de *.in-addr.arpa*.

En guise de réponse, cela devrait nous renvoyer le nom de domaine de l'*adresse*.

- Les domaines finissant par trois lettres sont des noms de domaines génériques. Ils sont également connus comme domaines organisationnels soit :
    - .org organisation
    - .edu éducation
    - .gov gouvernement US
    - .mil militaire US
    - .int organisation international
    - .com commercial et,
- Les domaines composés de deux caractères et basés sur les codes des pays conformément à la norme ISO 3166. Il s'agit dans ce cas là de domaines soit de pays soit géographiques.

Aucune entité unique ne peut gérer toutes les entrées de l'arbre DNS. La *Internet Assigned Number Authority* (*IANA*) gère cet arbre notamment en regard des domaines des niveaux supérieurs. De plus, et pour les autres zones comme ARIN (*American Registry for Internet Numbers)* il en délègue la responsabilité à d'autres entités pour l'Amérique du nord et pour bien d'autres pays, RIPE NCC (*Réseaux IP Européens et pour Network Coordination Center*), et APNIC pour l'Asie (*Asian Pacific Network Information Center*). Ces zones sont des arbres sous-jacents à l'arbre DNS.

Chaque entité responsable d'une zone peut y héberger plusieurs serveurs de noms de domaine. Sitôt qu'un nouvel hôte fait partie de ladite zone, l'entité administrative DNS de la zone en question lui assigne un nom et une adresse IP. Ces deux données figureront désormais dans la base de données hébergeant le serveur de noms de domaine.

Lorsqu'un serveur de noms de domaine est sollicité pour une demande d'information comme une adresse IP, et qu'il ne trouve pas cette information dans sa base de données, il doit prendre contact avec un autre serveur de noms de domaine pour en obtenir l'information. S'il ne connaît pas d'autres serveurs de noms de domaine ou qu'il ne peut pas obtenir l'information demandée, il se tourne alors vers le serveur racine de noms de domaine (*root name server*). Ce dernier lui enverra l'*adresse* du nouveau serveur à contacter. Actuellement, il existe dans le monde 13 serveurs racines de noms de domaine. Tous les serveurs primaires doivent connaître l'*adresse* de chacun de ces serveurs racines de noms de domaine.

Le DNS a recours à l'antémémoire. Lorsqu'un serveur de noms de domaine reçoit une information relativement à une *adresse* et à un nom, il la stocke dans son antémémoire afin qu'elle lui soit disponible et qu'il puisse répondre rapidement à une autre demande évitant ainsi un délai créé par la sollicitation auprès d'un autre serveur.

10.7.1 Format du message DNS

Le protocole DNS, qui fait partie de la couche Application, incorpore des messages utilisés pour solliciter des *adresses* ou pour envoyer une *réponse* à la réclamation faite.

Le protocole DNS se fonde sur le concept client serveur où des *questions* et des *réponses* entre hôtes et serveurs de noms de domaine sont échangées. Les messages qui comportent ces *questions* et les *réponses* sont transportés sur le réseau IP. Le DNS exploite les protocoles UDP ou TCP et, dans un cas comme dans l'autre, il a recours au port 53. Toutefois, le DNS fait plus souvent appel au protocole UDP.

La figure 10.21 indique le format du message DNS. Les champs de ce message sont les suivants :

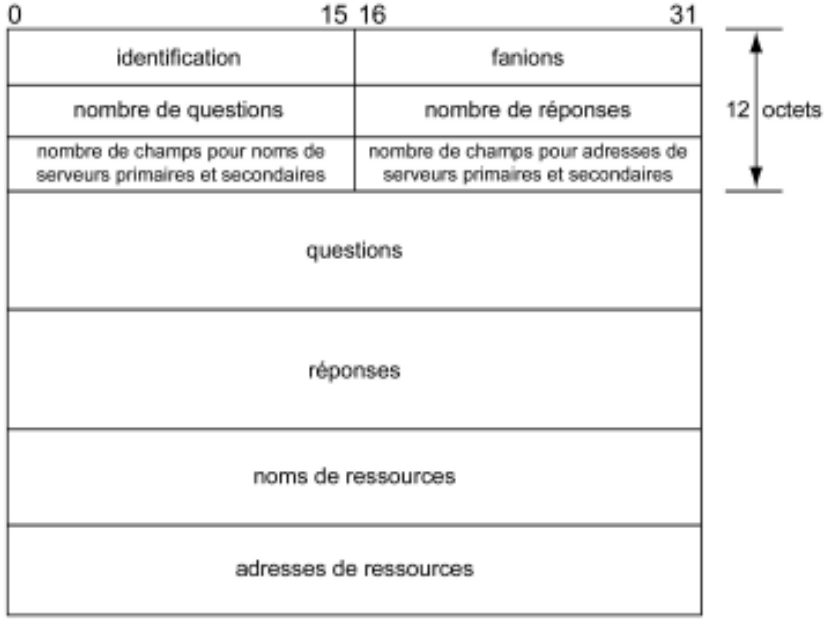


Figure 10.21 Format du message DNS

Identification – Le champ *Identification* est un numéro établi par le client et utilisé durant la transaction pour la *réponse* par le serveur.

1	4	1	1	1	1	3	4
QR	opcode	AA	TC	RD	RA	0 0 0	rcode

Figure 10.22 Format du champ fanion

Fanions – la figure 10.22 nous présente le format du champ fanion.

- Si QR est mis à 0 cela signifie que le message est une *demande*, mais s'il est mis à 1, on se trouve alors en présence d'une *réponse*.
- Lorsque Opcode se trouve à 0 cela nous signale que la *demande* est normale ; le client veut obtenir l'adresse IP à partir du nom. Lorsque Opcode est à 1, cela signifie qu'on se trouve en présence d'une *demande inverse*. Avec l'adresse IP un client veut obtenir le nom. Lorsque Opcode est à 2 on se trouve face à une demande d'état émanant du serveur.
- AA – (*authoritative answer*) Quand AA se trouve à 1, cela nous indique que le serveur de noms de domaine - fournissant l'information dans la réponse - est un serveur *primaire* ou *secondaire*.
- TC – (*truncated*). Lorsque ce champ se trouve à 1, cela nous signale qu'avec le protocole UDP, le maximum permis pour la réponse, soit 512 octets, ne suffit pas car ce nombre d'octets ne représente que le début de la *réponse*.
- RD – (*Recursion desired*). Lorsque ce bit est mis à 1, cela signifie que le client indique au serveur que ce dernier peut entamer procéder à la recherche de l'information telle qu'indiquée dans la *question* auprès d'autres serveurs.
- RA – (*Recursion available*). Lorsque ce bit est à 1, le serveur mentionne qu'il peut procéder à la récursivité.
- rcode – (*return code*). Ce champ nous signale l'état de la réponse du serveur. Lorsqu'il est à 0 cela signifie qu'il n'y a pas d'erreurs, par contre lorsqu'il se trouve à 1, cela nous prévient que le format de la *demande* est incorrect. Lorsqu'il se trouve à 2, il y a erreur du serveur. Enfin lorsqu'il se trouve à 3 il y a erreur de nom.

nombre de questions – Normalement ce champ est réservé à une seule question. Dans le cas d'une interrogation, le contenu de ce champ est à 1 et les trois autres compteurs se trouvent à 0.

nombre de réponses – Cela nous mentionne le nombre de champs contenant des réponses. Lorsqu'un serveur adresse une réponse, la valeur de ce champ est à 1 et les trois autres compteurs à 0.

nombre de champs pour noms de serveurs primaires et secondaires – Il s'agit d'une indication relative au nombre de champs indiquant le nom de chaque serveur *primaire* ou *secondaire* où se trouve l'information.

nombre de champs pour adresses de serveurs primaires et secondaires – Il s'agit des mêmes informations que celles figurant dans le champ précédent. Toutefois, dans ce cas-ci au lieu d'avoir des noms nous avons les adresses des serveurs primaires ou secondaires disposant d'informations.

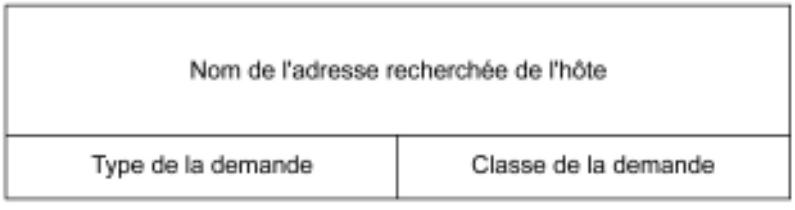


Figure 10.23 Contenu du champ question

questions – Les questions (normalement une par paquet) respectent le format illustré à la figure 10.23. Le nom figurant dans le champ question est celui dont l'*adresse* est recherchée. Il comporte des séquences de un ou plusieurs caractères précédés d'un compteur indiquant le nombre de caractères pour chacune des séquences. Le nom se termine par un octet de valeur 0. Exemple: pour demander l'*adresse* du nom suivant : ele.etsmtl.ca, le contenu du champ questions comprendra les caractères suivants, chacun contenu dans un octet :

3 e l e 6 e t s m t l 2 c a 0

3 est le compteur de l'octet faisant état du nombre de caractères de la première séquence : ele (3 octets).

6 le compteur de la deuxième séquence : etsmtl.

2 le compteur de la troisième séquence : ca;

et 0 signale la fin du champ comportant le nom.

Tout comme nous en traiterons plus loin, chaque question et chaque réponse comporte une catégorie. Les différentes catégories sont les suivantes :

Nom de la catégorie	Valeur	Description
A	1	adresse IP
NS	2	serveur de noms de domaine
CNAME	5	autre nom du système hôte (forme canonique)
SOA	6	gérant de zone (*Start of Authority*)
PTR	12	Pointeur (Pointeur courant)
HINFO	13	information hôte (*host info*)
MX	15	enregistrement d'échange de courriel (*mail exchange record*)
AAAA	28	adresse IPv6
*ou ANY	255	demande de tous les enregistrements (*request for all records*)

Cette dernière catégorie est utilisée seulement lors d'une *demande* émise par le client.

La plupart des interrogations sont du type A, c'est-à-dire une demande d'adresse IP. NS est utilisée pour obtenir l'adresse d'un serveur primaire ou secondaire.

La classe de la *demande* se trouve normalement mis à 1 et fait état de ce qu'il s'agit d'une adresse Internet.

Les trois derniers champs soit : réponses, noms des ressources (*Authoritative Resource Records*) et adresses des ressources (*Additional Resource Records*) partagent le même format connu sous l'acronyme de RR (*Resource Record*).

Le format de ces champs est illustré à la figure 10.24. Le nom de domaine figure toujours dans le premier champ dont la longueur varie. Il est représenté de la même façon que celle décrite dans le cas du champ question. La catégorie est aussi identique à celle utilisée dans le cas du champ question. Toutefois, certaines catégories sont exclusives aux messages propres à la *demande*. Le champ « Durée de vie » ou TTL (*Time to Live*) mentionne le temps alloué à l'information. Le champ «longueur de données» fait état de la longueur du champ dans lequel les données se trouvent : « données de ressources ».

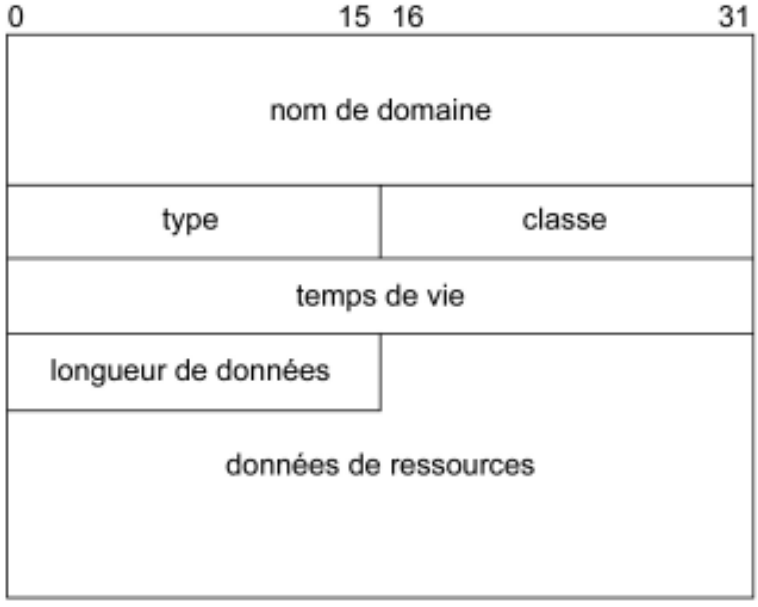


Figure 10.24 Format du message de réponse ou RR

10.7.2 Opération du DNS

Les services d'un serveur DNS sont demandés par un client lorsque ce dernier ne trouve pas l'adresse désirée dans son répertoire local de nom (*resolver*). La *demande* se fait par le biais de l'envoi d'un message de «*demande*» tel qu'illustré à la figure 10.25.

Si le système de serveur de noms de domaine répond directement c'est qu'il a autorité sur le domaine en question. Sinon il consulte son antémémoire pour voir si l'*adresse* ne s'y trouve pas.

Dans le cas contraire, la résolution de nom se fait par un accès descendant (*top-down*), au-delà même du serveur local. Le serveur local s'adresse directement au serveur de noms de domaine racine. Ce dernier adresse une *demande* aux serveurs DNS en commençant par les domaines du niveau supérieur TLD.

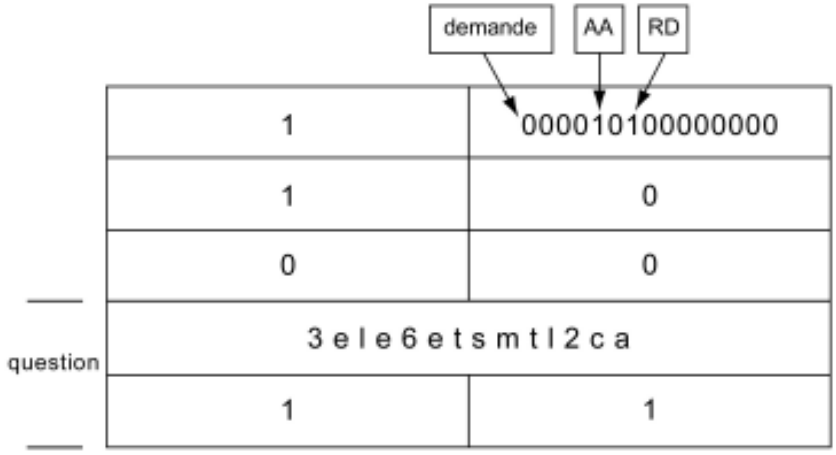


Figure 10.25 Demande d'adresse IP pour ele.etsmtl.ca

Sitôt le bon serveur trouvé et l'*adresse* obtenue, l'information est acheminée vers le serveur local qui, à son tour, la transmet au client par le biais d'un message tel qu'illustré à la figure 10.26. Il la mémorise aussitôt dans son antémémoire en vue d'une *demande* future.

Dans notre exemple, la *demande* concerne l'adresse IP *ele.etsmtl.ca*. La figure 10.25 illustre le message transmis. L'identification est alors mise à 1. Pour sa part, le bit RQ se trouve à 0, indiquant ainsi qu'il s'agit d'une *demande*. Une seule *question* est transmise : quelle est l'adresse de *ele.etsmtl.ca*? La r*éponse* provenant d'un serveur est illustrée à la figure 10.26. L'identification se trouve également à 1 pour s'assurer qu'il s'agit bien là de la *réponse* à la *question* transmise. Le bit RQ est mis à 1 et nous indique qu'il s'agit d'une *réponse*. La *question* est alors retransmise et la *réponse* obtenue est : 142.137.3.65.

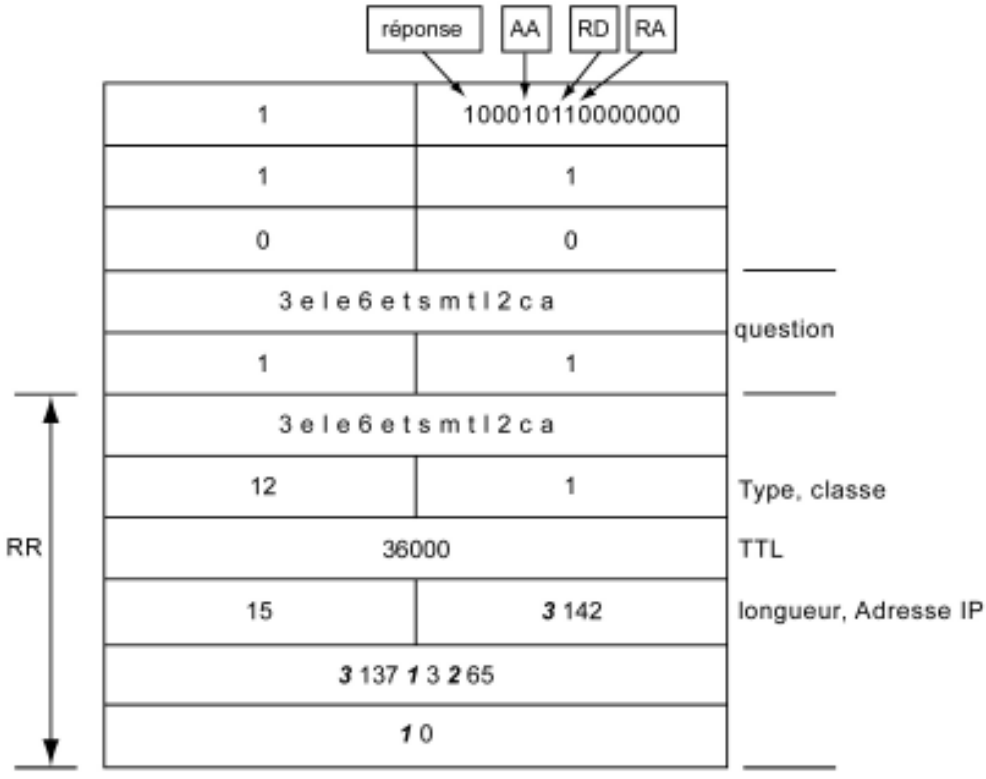


Figure 10.26 Réponse d'adresse IP pour ele.etsmtl.ca

10.7.3 Commande nslookup

La commande *nslookup* est un outil d'analyse DNS disponible sur certains systèmes Windows et sur Unix. Tout client connecté à Internet peut faire appel à cette commande qui montre l'information obtenue d'un serveur *primaire* par un serveur *secondaire*.

Il s'agit de procéder à l'analyse de l'enregistrement des ressources (*Resource record*) et à l'analyse des défaillances de système (*Start Of Authority* ou *SOA*). Cet enregistrement nous signale où trouver la meilleure source d'informations d'un domaine donné. *nslookup* comprend plusieurs séquences qui sont définies par différentes commandes.

Le programme *nslookup* est le premier à être amorcé. Le genre d'enregistrement des ressources est clairement indiqué par la séquence « set type=SOA ». Cette commande permet à *nslookup* de prévoir l'interrogation du serveur par défaut. Le

nom de domaine pour lequel l'*adresse* est recherchée est ensuite inscrit (*ele.etsmtl.ca*).

La réponse du serveur par défaut vl-mo-dn000-fac1.mo.videotron.ca nous signale que le serveur *primaire* par défaut de *ele.etsmtl.ca* est *goldfinger.ele.etsmtl.ca.*

Le résultat de la commande nous est illustré par l'exemple suivant :

```
C:\> nslookup
Serveur par défaut : vl-mo-dn000-fae1.mo.videotron.ca
Address : 24.200.243.234

>set type=SOA
>ele.etsmtl.ca
Serveur : vl-mo-dn000-fae1.mo.videotron.ca
Address : 24.200.243.234

Réponse ne faisant pas autorité :
Ele.etsmtl.ca
        primary name server = goldfinger.ele.etsmtl.ca
        responsible mail addr = hostmaster.ele.etsmtl.ca
        serial      = 2003121600
        refresh     = 3700 (1 hour 1 min 40 secs)
        retry               = 1800 (30 mins)
        expire      = 604800 (1 day)

etsmtl.ca nameserver = aniline.ad.etsmtl.ca
etsmtl.ca nameserver = ambre.ad.etsmtl.ca
etsmtl.ca nameserver = world.etsmtl.ca
aniline.ad.etsmtl.ca          internet address = 142.137.3.65
```

L'adresse IP de l'accès principal du système hôte demandée est également obtenue.

Bien que le DNS soit une application au sens de l'architecture TCP/IP, il est capital d'avoir eu à en traiter dans le présent chapitre parce qu'il a été pensé et conçu, tout comme nous l'avons vu, pour soutenir le protocole IP.

Le chapitre 13 « Les applications sur Internet » décrira des applications qui sont des services supportés par les couches inférieures et indépendants de ces couches.

10.8 IP Mobile

Tout équipement connecté à un réseau comprend une adresse IP. Cette adresse lui est assignée par son fournisseur de service, soit par DHCP ou par adresse fixe. Dans le cas d'assignation d'adresse par DHCP, cette dernière change périodiquement et n'est jamais la même. Par contre, l'adresse fixe allouée à un terminal ne change pas. Elle reste permanente. L'équipement détenant cette adresse est joignable en tout temps. Certains éléments de réseaux ont besoin de cette caractéristique par la nature du service qu'ils offrent. C'est le cas de serveurs ou encore de téléphone IP. Si un appareil téléphonique IP, par exemple, est déplacé hors de son domaine, c'est-à-dire hors de l'accès de son point d'attache fixe à son réseau administratif, il ne pourrait pas être accessible. Il ne pourra pas utiliser son adresse IP permanente dans son nouveau réseau visité et devra adopter une nouvelle adresse IP qui lui sera assignée selon l'adresse réseau de ce nouvel accès.

Toutefois, si le terminal adopte l'option IP Mobile (*Mobile IP*), il pourrait se déplacer au sein ou en dehors de son réseau et être toujours accessible même sur un réseau étranger tout en gardant son adresse IP fixe originale. Ceci le rendrait donc joignable partout ou il se trouverait.

Il existe deux versions de norme, notamment *Mobile IPv4* (RFC 3344) et *Mobile IPv6*. Les deux versions fonctionnent de la même façon, mais sont naturellement utilisées respectivement dans les réseaux IPv4 et IPv6. La version *Mobile IPv6* bénéficie des avantages de IPv6 tels qu'une sécurité améliorée et un routage plus performant.

La norme Mobile IP permet au terminal, doté de cette option, d'utiliser en pseudonyme une adresse IP du réseau visité tout en gardant son lien et son adresse permanente avec son réseau administratif. Ce dernier relaierait vers sa nouvelle destination tous les datagrammes adressés à son adresse permanente.

Afin de mieux comprendre l'architecture de IP Mobile, il est important de définir les éléments qui la constituent.
- Le terminal itinérant, à qui une adresse IP fixe est assignée originalement, est connu sous le nom de nœud mobile (*mobile node*) ou MN. C'est le terminal qui doit recevoir les appels qui sont adressés à son adresse permanente. Ce terminal se trouve avoir une autre adresse qui lui est assignée par l'agent visité (Foreign Agent - FA). Cette adresse se nomme COA pour *Care Of Address*.
- Le correspondant (*Correspondant Node*) ou CN est le terminal source qui désire adresser le MN.

- Le réseau d'attache (Home Network) qui assigne l'adresse IP fixe au MN comprend un nœud ou routeur qui connaît l'adresse permanente du MN ainsi que sa nouvelle adresse provisoire (COA). Ce nœud ou routeur est l'agent d'attache (*Home Agent*) ou HA.
- En contrepartie, le réseau visité où se trouve le MN, comprend un nœud ou routeur qui lui assigne provisoirement une nouvelle adresse IP, le COA. Ce nœud ou routeur est l'agent visité (*Foreign Agent*) ou FA.

La figure 10.27 illustre un cas de figure d'un appel IP utilisant le IP Mobile. Le CN transmet des datagrammes au MN en utilisant son adresse permanente. Le HA intercepte cet appel, sachant que le MN ne se trouve pas dans son réseau d'attache. Le HA connaît l'adresse provisoire COA de MN. Il réachemine chaque datagramme en les encapsulant avec un nouvel en-tête dont l'adresse destination est le COA. Il crée ainsi un tunnel.

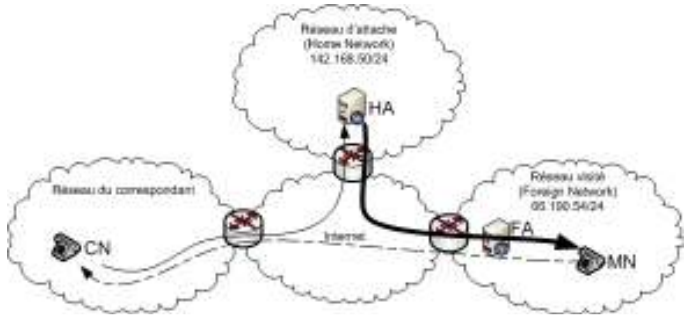


Figure 10.27 Architecture de IP Mobile

Les datagrammes reçus dans le réseau visité peuvent être traités de deux différentes façons.
Dans le premier cas, l'adresse COA est l'adresse du FA. Ce dernier dé-encapsule ces datagrammes reçus et les transmet au MN. Le MN reçoit donc les datagrammes d'origines. Cette approche est connue sous le nom de *FA based COA*.
Dans le deuxième cas, le FA assigne originalement une adresse IP provisoire au MN et l'identifie comme COA. Les datagrammes adressés à cette adresse sont directement remis au MN qui les dé-encapsule et traite le contenu qui est le datagramme d'origine. Cette approche est connue sous le nom de *Colocated COA*. Dans cette méthode, le MN reconnaît avoir deux adresses, son adresse permanente et son COA.
Le MN peut répondre au CN, et lui transmettre des datagrammes directement sans passer par le HA. Cette dernière approche n'est pas toujours possible surtout si le réseau visité comprend des pare-feu ou des filtres qui détruisent les datagrammes n'ayant pas une adresse source une adresse de ce réseau. C'est le

cas de notre MN qui a son adresse source d'origine. Lorsque la transmission directe du MN au CN ne peut se faire, le MN peut avoir recours à la transmission par le biais de son HA en utilisant son adresse COA. Le HA acheminerait par la suite ces datagrammes au CN. Ceci est connu sous le nom de *reverse tunneling*.

Lorsque le MN se retrouve dans un réseau visité, il diffuse un message de reconnaissance (*Agent Discovery*) à la recherche du FA ou répond au message *Agent Advertisement* transmit périodiquement par le FA. Le FA fait connaître au MN son adresse COA. Ce dernier (ou le FA) le transmet au HA pour s'enregistrer. Le HA retourne une réponse et l'authentifie. Ces échanges avec le HA se font dans la couche transport avec des segments UDP.

Le MN s'enregistre chaque fois qu'il change de réseau pour faire connaître sa nouvelle adresse COA au HA.

Plusieurs approches ont été proposées pour améliorer le IP Mobile. Il existe en effet plusieurs problèmes d'inefficacité dans la solution rapportée dans la norme. Il y a tout d'abord l'en-tête additionnel qui crée une surcharge de trafic. La dépendance de MN sur le FA ne permet pas de méthode d'optimisation de chemin. La méthode la plus simple serait de permettre au CN de connaître l'adresse COA du MN et de lui adresser directement les datagrammes. Cette solution engendre un nouveau problème lorsque le MN se déplace dans un nouveau réseau. En effet, ce dernier n'a pas moyen d'aviser le CN. Une autre solution serait d'impliquer le FA comme intermédiaire dans ce cas de mobilité.

10.9 Exercices

10.1

Une station a pour adresse MAC 04 0C 00 45 11 22. Elle a des données à transmettre à la station dont l'adresse IP est : 192.168.54.128/24. La station émettrice ne connaît pas l'adresse MAC cible pour communiquer avec la station de destination. Elle doit donc adresser une demande ARP. Toutefois, la station source ne connaît pas non plus sa propre adresse IP.

L'adresse IP de la source est selon le serveur RARP 192.168.54.47/24. L'adresse MAC de la station cible est 00 00 05 04 02 15.

a) Quelles procédures la station émettrice doit-elle amorcer?
b) Quel est le contenu utilisé par chaque trame?

10.2

Une station désire transmettre des données à une autre station. La station émettrice possède les adresses suivantes : MAC = 00 00 09 C4 45 18, IP=192.168.14.189/24. L'adresse IP de la station cible est : 192.168.45.200/24 et son adresse MAC est 00 00 09 62 05 21.

Un routeur sépare les deux sous-réseaux où se trouvent les stations dont il est question. Le routeur utilise les adresses hôte 192.168.14.1 et 192.168.45.1 dans les deux sous-réseaux. Les adresses MAC du routeur sont : du coté du sous-réseau du transmetteur 00 00 09 15 35 28 et du coté de la station cible 00 00 09 55 24 74.

La station émettrice ne connaît pas l'adresse MAC du routeur et le routeur ignore l'adresse MAC de la station cible.

a) Donnez la topologie du réseau et indiquez le contenu de chaque demande et réponse ARP utilisées avant transmission des données.
b) Indiquez quelles adresses MAC et IP ont été utilisées lors de la transmission des données dans les trames et dans les datagrammes?

10.3

Un gestionnaire de réseau désire développer un petit programme pour déterminer la durée de rotation (RTT) moyenne de paquet transmis de sa station à une autre station sur Internet.
Décrivez quelles seront les étapes à suivre par le programme pour obtenir l'information.

10.4

Le protocole DHCP permet au serveur de proposer une adresse IP au client qui veut bénéficier de ce service.

a) Indiquez exactement dans quel message DHCP et dans quel champ de ce message se trouve l'adresse proposée.
b) Le client accepte cette adresse. Indiquez dans quel message DHCP et dans quel champ de ce message se trouve l'adresse qui est acceptée.

10.5

Un usager a obtenu une seule adresse IP par son fournisseur de service Internet (ISP). Cet usager a quatre mini-ordinateurs qu'il désire brancher sur Internet. Il utilise un routeur qui comprend le service NAT. L'adresse IP publique qui lui a été transmise par son ISP est 142.144.5.5.
Décrivez comment la traduction NAT est effectuée et indiquez les entrées de la table de traduction NAT pour les quatre stations si des adresses privées sont utilisées.

10.6

Une demande d'adresse est effectuée par une station à un serveur DNS. La station désire obtenir l'adresse du nom « microsoft.com ».

a) Décrivez les messages de demande et de réponse qui sont échangés entre le client et le serveur DNS.
b) À l'aide du programme Ethereal vérifiez votre réponse.

(Ethereal est un programme de surveillance de protocole. Il donne en détail tous les PDUs échangés entre la station d'où il est activé et son accès réseau. Ethereal est disponible gratuitement sur Internet.)

10.7

Des solutions plus efficaces que celle proposée dans la norme IP Mobile pourraient être appliquées. Prenons le cas de la communication directe entre le CN et le MN. Décrivez comment le protocole pourrait fonctionner. Si le MN venait à changer une deuxième fois de réseau, comment pourrait-on envisager une solution en impliquant le FA du premier réseau visité ?

11 Les protocoles de routage

11.1 Introduction

Le transport des paquets d'une station source vers une station de destination à travers un réseau fait appel à une fonction plus complexe que celle de la couche liaison qui transfert des trames d'un point donné à un autre point adjacent. La couche réseau fonctionne selon la topologie du sous-réseau de communication, et doit opter pour les chemins les plus directs, tout en évitant les congestions. Si la destination se trouve dans un réseau différent, la couche réseau doit en tenir compte.

Les protocoles de routage sont indispensables à tout réseau. Ils permettent à tous les routeurs de déterminer le meilleur chemin pour faire parvenir les données à la station de destination.

Le réseau Internet est un ensemble de plusieurs réseaux autonomes interconnectés. Le réseau autonome comprend des protocoles de routage homogènes.

Il existe des protocoles de routage interne utilisés par les routeurs au sein du même domaine de routage. Ainsi, chaque routeur sait quel est l'état des autres routeurs, de même que les chemins y menant.

Les tables du routage, qui figurent dans chacun des routeurs, reflètent l'état des routeurs même et celui du chemin à emprunter pour y parvenir. Ces tables sont mises à jour grâce au protocole de routage qui permet l'échange de cette information entre tous les routeurs relevant du même système autonome.

On distingue deux catégories de protocoles de routage internes : les protocoles axés sur le vecteur de distance (*Distant Vector*), et ceux orientés sur l'état de lien (*Link State*). Le concept de vecteur de distance repose sur l'algorithme de

Bellman-Ford[1] alors que celui d'état de lien est fondé sur l'algorithme de Dijkstra[2].

Le protocole axé sur le vecteur de distance établit son chemin de la station source jusqu'à la station de destination en diminuant le nombre de sauts des nœuds intermédiaires. L'information relative au routage est échangée entre les routeurs adjacents. Les routeurs transmettent initialement le coût de leurs propres liens. Chaque routeur évalue, séparément des autres routeurs, le meilleur chemin pour parvenir à destination. Il en fait part aux routeurs adjacents qui, à leur tour, jugent du meilleur chemin à prendre pour se rendre jusqu'à l'hôte de destination en bénéficiant cette fois-ci des informations recueillies par les autres routeurs. Ce processus est itératif. L'information se propage à travers le réseau. Dans la mesure où il n'y a pas de changement d'état des liens et des routeurs, les tables du routage finissent par se stabiliser. Le moindre changement génère une autre vague de mises à jour des tables de routage. Le protocole RIP (R*outing Information Protocol*) est un exemple de protocole de routage axé sur le vecteur de distance.

Le protocole orienté sur l'état de lien établit le chemin depuis la source jusqu'au terminal de destination en optant pour la voie la moins onéreuse. Les coûts afférents à chacun des liens sont connus des routeurs auxquels ils sont reliés. Ainsi, le coût du chemin menant au terminal de destination est la somme des coûts de tous les liens traversés. Chaque routeur transmet le coût de tous ses liens. Cette information est donc envoyée à tous les routeurs. Chacun d'entre eux est alors informé de l'état de tout le réseau et pourra identifier le chemin le plus court (ou le moins coûteux) vers toutes les destinations. Le protocole OSPF (*Open Shortest Path First*) est un exemple de protocole axé sur l'état de lien.

Les réseaux autonomes peuvent avoir recours à des protocoles distincts de routage. Ceux-ci sont aussi gérés par divers gestionnaires du réseau et il est impensable que la gestion de routage d'un réseau puisse relever d'une administration autre que la sienne. C'est la raison pour laquelle il existe des protocoles de routage externes. Ceux-ci sont utilisés entre réseaux autonomes. Le protocole BGP (*Border Gateway Protocol*) en est un exemple.

---

[1] R. E. Bellman, « *Dynamic Programming* » Princeton University Press, Princeton N.J., 1957.
et L.R. Ford, D. R. Fulkerson, « *Flows in Networks* », Princeton University Press, Princeton N.J., 1962.
[2] E. W. Dijkstra, « *A note on Two Problems in Connection with Graphs* », Numerische Mathematic, vol.1, pp. 269-271, 1959.

11.2 Le protocole de routage RIP

Le protocole RIP (*Routing Information Protocol*) appartient à la catégorie du « vecteur de distance ». Il a été élaboré à même l'algorithme de Bellman-Ford. Chaque routeur faisant appel au protocole RIP est composé d'une table de routage qu'il met à jour dès réception des messages RIP transmis par les routeurs adjacents. Chaque routeur connaît son environnement immédiat. Il est au courant de la condition de chaque lien et réseau et peut de ce fait y associer un coût ou une métrique représentant une pénalité par rapport aux autres voies disponibles. Par défaut, le coût de chaque lien est de 1. On alloue une métrique de 16 à un réseau qui devient inaccessible.

La table de routage couvrant le réseau ou sous-réseau est mise à jour périodiquement dès réception des messages RIP.

Pour mieux comprendre le fonctionnement du protocole RIP, nous allons analyser un réseau composé de 5 nœuds et de 6 liaisons. Comme celui indiqué à la figure 11.1.

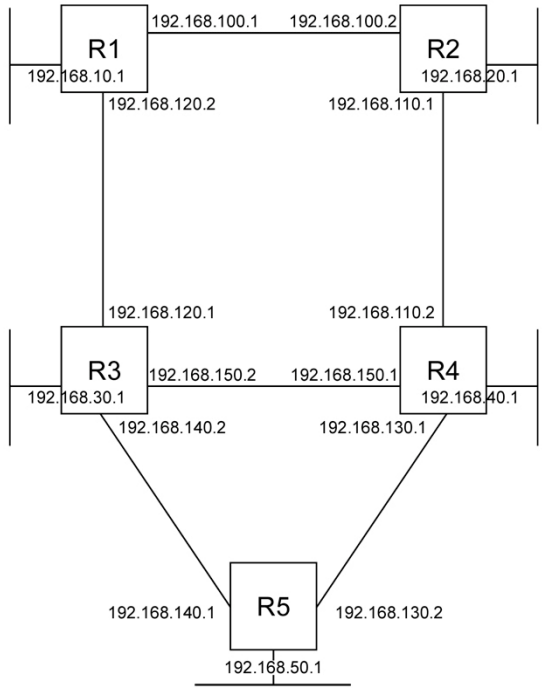


Figure 11.1 Réseau autonome à 5 nœuds

Chaque nœud ou routeur est composé d'une table de routage. À l'origine, les tables de chaque nœud ne comprennent que des entrées servant de destination (locale) et dont la métrique indiquent 0. Ceci est illustré dans la figure 11.2.

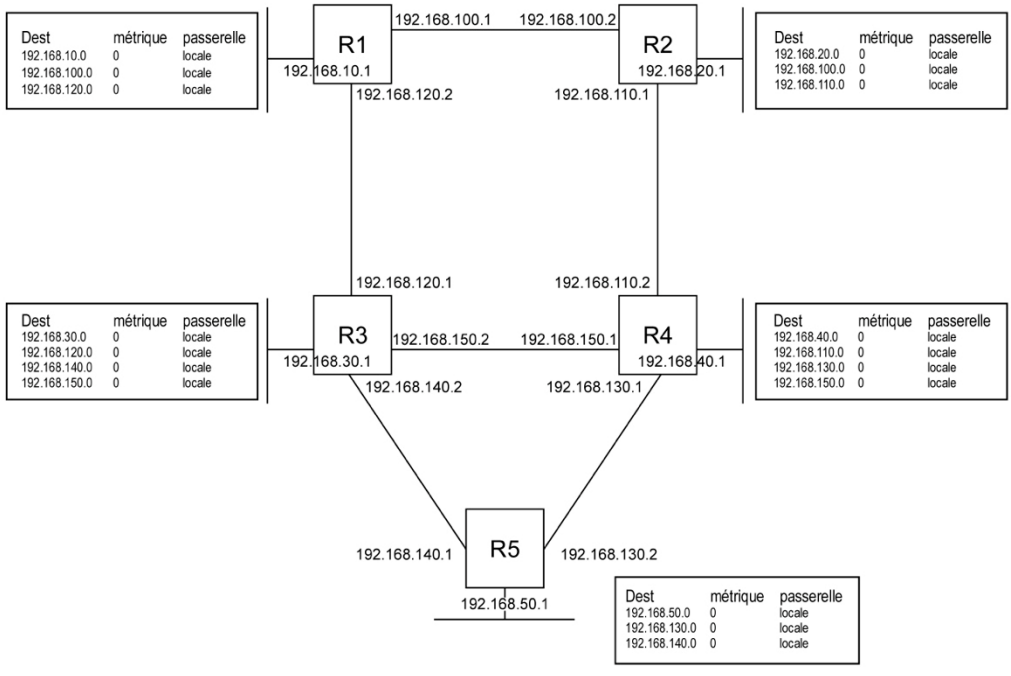


Figure 11.2 Tables initiales de routage du protocole RIP

Le nœud R1 transmet des messages RIP aux nœuds R2 et R3. Ces messages renferment l'adresse IP du réseau, et la métrique qui est 0. Grâce à cette métrique, le nœud R1 indique que le segment lui est directement accessible. Ainsi, les nœuds R2 et R3 font une mise à jour de leur propre table de routage en incrémentant la métrique de la valeur des vecteurs reçus. Chaque table de routage comprend l'adresse réseau, la métrique, l'adresse IP de la prochaine passerelle menant à ce réseau de destination (et donc à toutes stations y faisant partie), un fanion pour indiquer si l'information a récemment changé ainsi que des temporisateurs.

On peut concevoir que le même processus se passe avec chacun des autres nœuds. Les entrées des tables du routage viennent ajouter de nouvelles données qui nous mènent aux nœuds adjacents.

Après les premiers messages, qui ne comprenaient à l'origine qu'une entrée, d'autres messages nouveaux ceux-là et comprenant le contenu des tables de routage sont revues entre nœuds adjacents. Il en va ainsi, lorsque le nœud R2

transmet des messages RIP aux nœuds R1 et R4 en leur remettant le contenu de sa table de routage. Les nœuds R1 et R4 font alors une mise à jour de leur propre table de routage en incrémentant le coût des vecteurs reçus (figure 11.3).

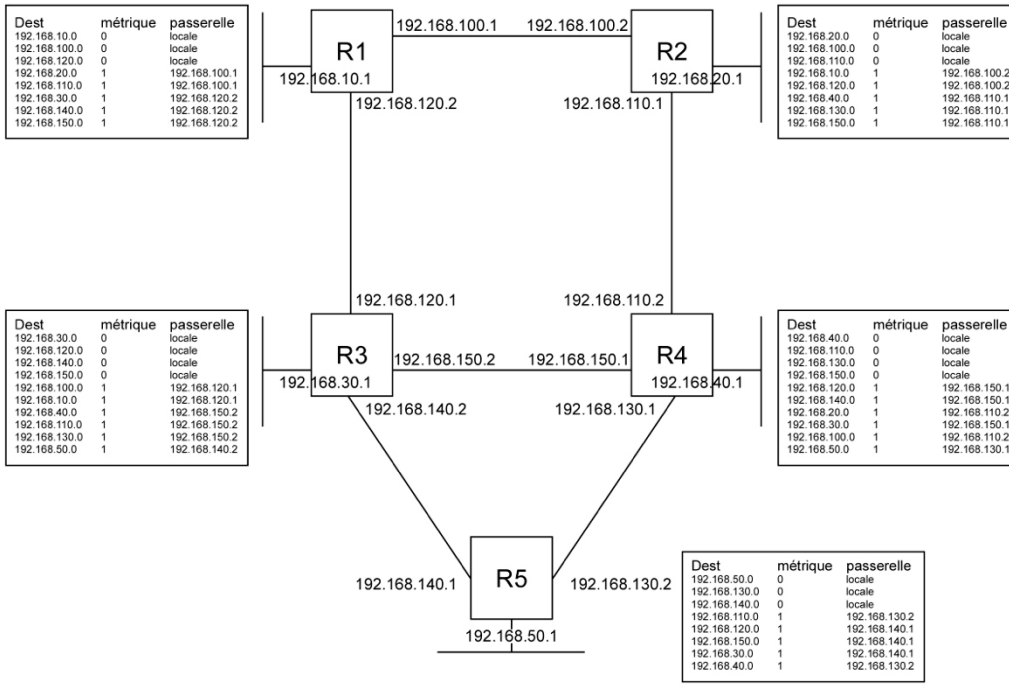


Figure 11.3 Propagation de la table du routage

Le nœud R3 transmet aussi des messages RIP aux nœuds R1 et R4 en leur remettant le contenu de sa table de routage. Les nœuds R1 et R4 font alors une mise à jour de leur propre table en incrémentant le coût des vecteurs reçus. Toutefois, le nœud R4 pourrait décider de ne conserver que l'information relative au coût du nœud R1 qu'il a obtenue du nœud R2 plutôt que celle obtenue du nœud R3 puisque les deux valeurs sont les mêmes. Le seul changement réside dans l'adresse de la passerelle. La figure 11.4 en illustre clairement l'exemple.

Les nœuds ignorent les informations les concernant et qu'ils reçoivent des autres nœuds. De plus, tous les nœuds ignorent les informations reçues sur les réseaux dont les valeurs sont soit plus grandes soit égales à celles déjà stockées dans les tables du routage.

La table de routage de chacun des nœuds tient finalement compte de tous les autres nœuds et elle est sans cesse mise à jour. La figure 11.4 nous donne un bon aperçu des tables de routage à un moment de stabilité.

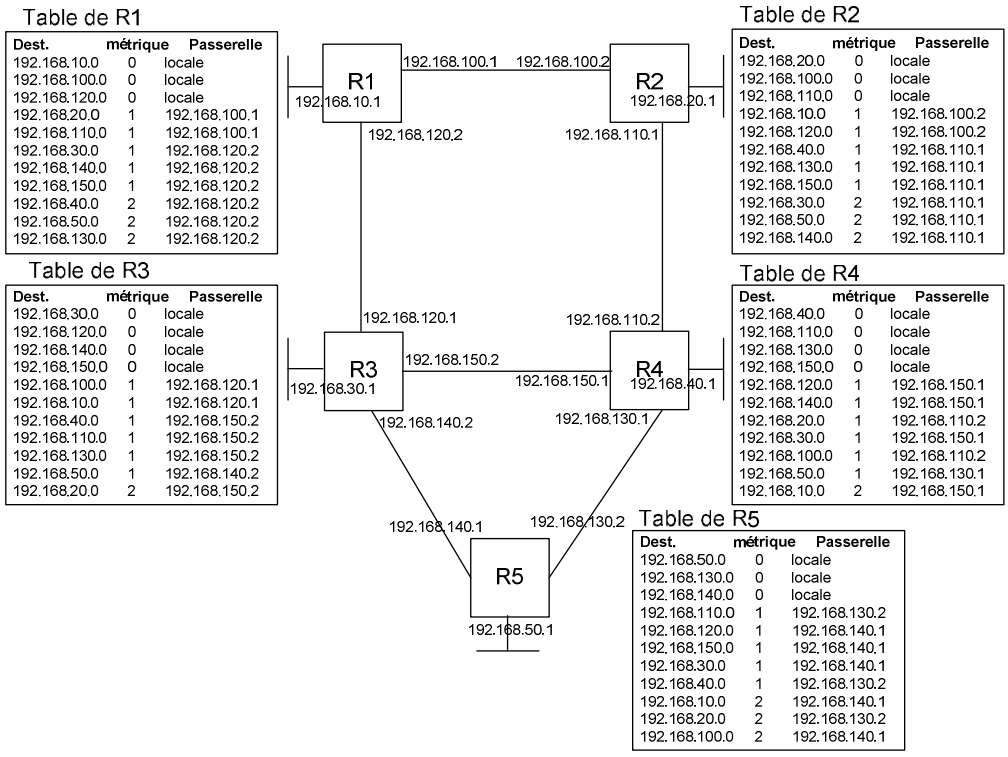


Figure 11.4 Stabilisation des tables du routage RIP

11.2.1 Temporisateurs RIP

Le protocole RIP comprend divers temporisateurs qui vont lui permettre de régir les tables de routage. Les compteurs sont :

- *Mise à jour (Update)* : Il s'agit de la fréquence en secondes des envois d'informations relatives au routage (valeur par défaut = 30 secondes)
- *Invalide* : Ce compteur désigne le nombre de secondes avant qu'une route soit proclamée invalide en raison de l'absence du message *Update* (valeur par défaut = 180 secondes)
- *Retrait (flush)* : Ce compteur désigne le nombre de secondes avant qu'une route ne figure définitivement plus dans la table de routage (valeur par défaut = 240 secondes)

11.2.2 Format de message RIP version 1

Le protocole RIP est clairement énoncé et défini dans le RFC 1058. Les messages RIP ont recours aux segments UDP sur le port 520. L'information est diffusée par les routeurs toutes les 30 secondes.

Le message de la version 1 du protocole RIP comprend les informations figurant dans les tables de routage. Ces informations sont : l'adresse IP de réseau et la métrique ou coût pour l'atteindre à partir du nœud qui transmet l'information. Dans la version 1 le masque de réseau du message ne figure pas. Aussi, on présume que le réseau fonctionne selon les classes d'adresses.

Les informations pour 25 réseaux peuvent être transmises dans un message de la version 1 du RIP.

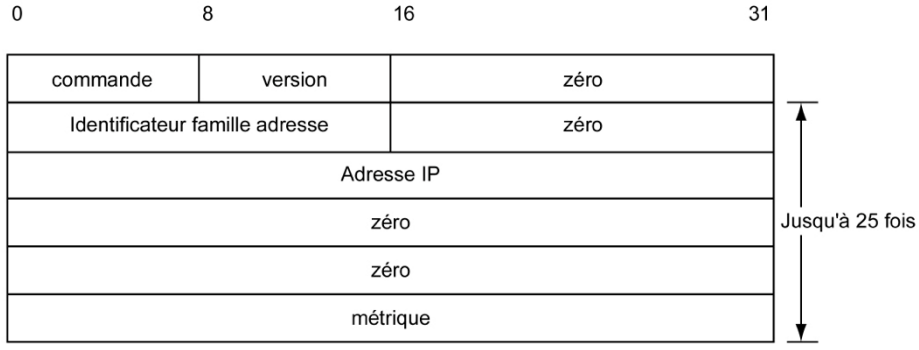


Figure 11.5 Format du message de la version 1du RIP

Les champs compris dans le message et illustré dans la figure 11.5 sont les suivants :

Commande – Ce champ indique la nature du message. On distingue deux commandes :

- Requête – ceci est une requête demandant la transmission d'une partie ou de toute la table de routage.
- Réponse – Message comprenant une partie ou toute la table de routage. Ce message est généralement envoyé suite à une requête.

Pour la version 1 du protocole RIP, le champ version est codé à la valeur 1.

L'identificateur famille d'adresse est là pour indiquer le protocole qui se sert de l'adresse. Seul IP a toutefois été identifié pour être utilisé avec RIP. Ce champ est codé à 2 pour IP.

Les informations qui suivent sont celles que l'on retrouve partiellement dans les tables de routage. Il s'agit de l'adresse IP du réseau et de la métrique qui lui est associée.

Un processus RIP silencieux ne transmet pas de messages RIP. Par contre, il écoute les messages échangés entre les autres nœuds. Son utilisation par un hôte qui exerce la surveillance de réseau en est un exemple.

11.2.3 Format du message RIP dans la version 2

RIP étant le protocole de routage le plus répandu, il a été jugé nécessaire de procéder à une mise à jour de la version 1 en y ajoutant le masque réseau ainsi que le processus d'authentification des messages de routage.

La version 2 de RIP est traitée dans les RFC 1387 et 1388. Elle permet le routage de sous-réseaux (véhicule le masque de sous-réseau). Elle identifie les routes externes empruntées par un EGP (*Exterior gateway protocol*). Les versions 1 et 2 sont compatibles. La version 1 utilise la diffusion pour envoyer des messages RIP. La version 2 fait appel à la multi distribution (*multicast*) à l'adresse 224.0.0.9. La version 2 permet aussi de procéder à l'authentification des informations reçues.

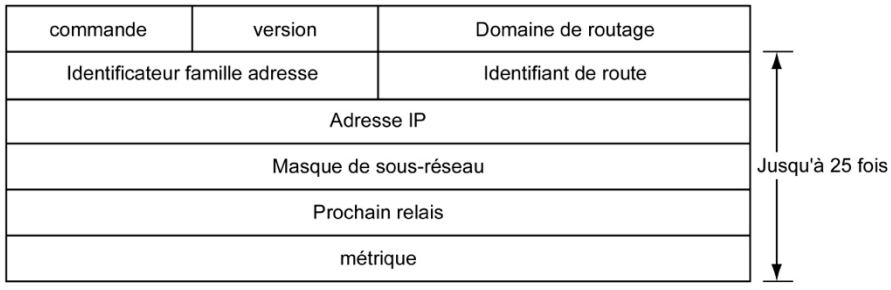


Figure 11.6 Format du message de la version 2 du RIP

Le format du message de la version 2 du RIP est illustré à la figure 11.6. Le champ version est codé à la valeur 2.

Sur un seul et même lien, plusieurs processus RIP peuvent coexister. Chacun de ces processus est autonome l'un par rapport à l'autre et exige sa propre table de routage. Le numéro du domaine de routage indique le numéro du processus de routage auquel la mise à jour appartient. Ce champ associe cette mise à jour à un processus de routage propre au routeur de réception.

L'authentification peut-être utilisée dans le cas de chacun des messages de la version 2 du protocole RIP. Cette procédure est incontournable pour éviter à des nœuds indésirables de transmettre des messages RIP et de les diffuser dans tout

le réseau. Ainsi, lorsque l'authentification est utilisée, les 20 octets qui suivent le premier 32 bits du message revêtent le format tel qu'illustré à la figure 11.7.

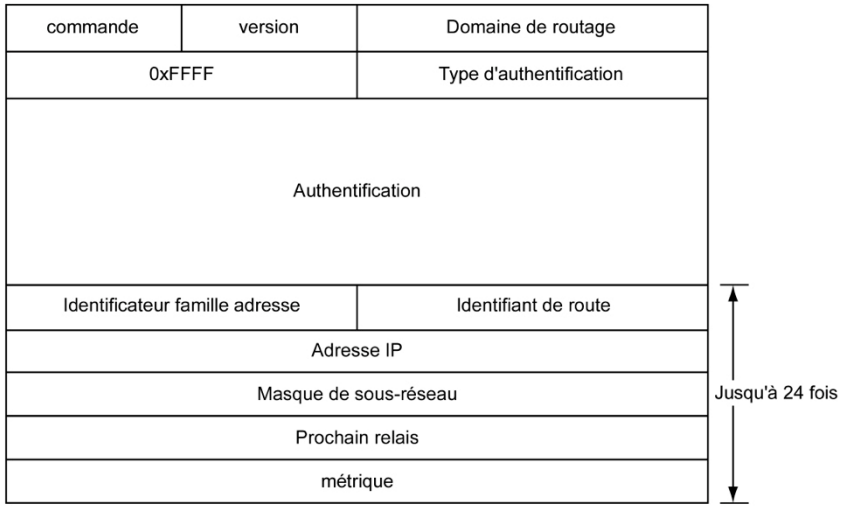


Figure 11.7 Format du message de la version 2 du RIP

L'authentification est indiquée par le code 0xFFFF dans le champ identificateur famille d'adresse. Le genre d'identification indique le codage utilisé. Les 16 octets suivants incluent le code d'authentification. Les champs suivants fournissent les informations du routage comme indiqué précédemment.

11.2.4 Problèmes de convergence

Le fait que le protocole de routage RIP ne permette aux routeurs de ne connaître que ses voisins peut donner lieu à des situations créant des problèmes de convergence, qui surviennent dans des cas particuliers.

Nous illustrons l'une de ces situations pour mieux saisir la portée du problème. Les réseaux illustrés à la figure 11.8 sont reliés par des routeurs. Ceux-ci s'envoient des informations RIP.

Tous les routeurs sont tenus de faire une mise à jour de leur table de routage et chacun de ces routeurs sait quelle passerelle emprunter pour atteindre l'hôte dans le réseau H1. Le routeur R2 a, dans sa table, une entrée qui lui indique la passerelle vers R1. Ainsi, R2 acheminera par la passerelle indiquée dans sa table toutes les transmissions destinées à ce réseau. Le routeur R4 sait que, pour atteindre ce même réseau, il devra le faire par le biais du routeur R2 pour un coût de 2 (selon la métrique indiquée dans sa table). Finalement, le routeur R5 a accès

à ce réseau par le biais du routeur R4 conformément à sa table de routage avec une métrique dont la valeur est égale à 3.

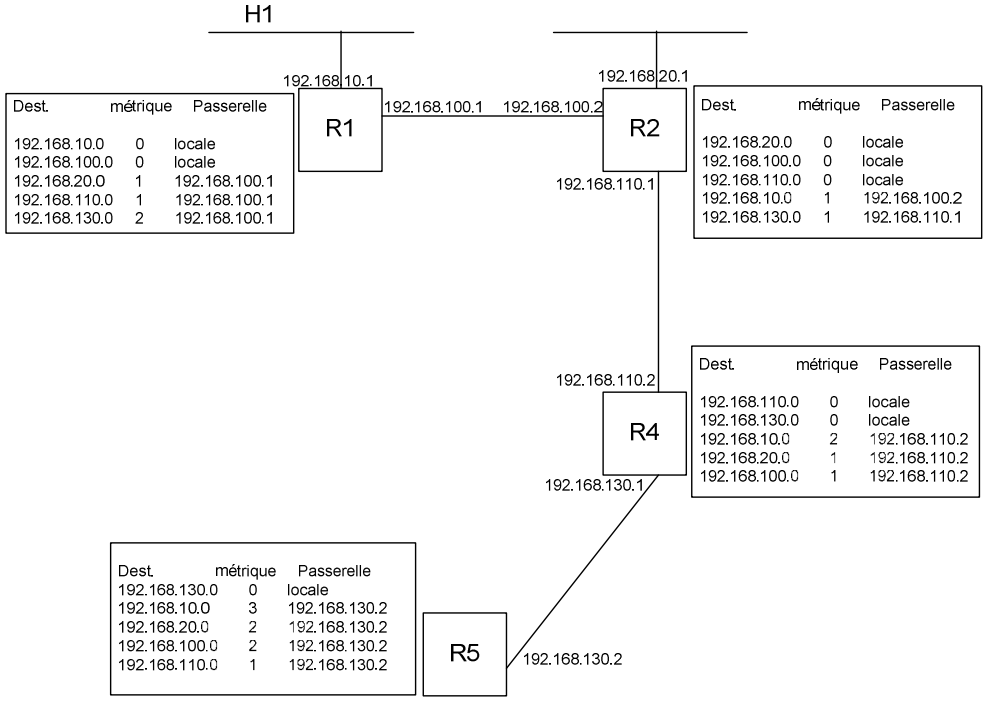


Figure 11.8 Scénario propice au problème de convergence

Advenant le cas où le réseau H1 en question soit inaccessible, le routeur R1 diffusera l'information avec une métrique égale à 16 pour ce réseau. Tel que déjà mentionné, la métrique de 16 signifie l'infini, (∞).

Le routeur R2 reçoit périodiquement les informations pour procéder à la mise à jour des routeurs R1 et R4 de sa table du routage. Lors de la mise à jour suivante, le routeur R2 prendra l'information du réseau H1 non de R1, mais de R4 attendu que la métrique fondée sur les données de R4 est égale à 3.

Le routeur R2 émettra ses nouvelles informations incluant celles reçues du routeur R4 en rapport avec le réseau H1. Le routeur R4 recevant une nouvelle mise à jour de R2 changera son entrée pour le réseau H1 et lui assignera une métrique égale à 4 dans sa table du routage. Pour ce réseau, l'entrée convergera fort probablement vers la métrique 16, soit vers l'infini. Étant donné que cette convergence sera trop lente, les routeurs ne pourront pas réagir et parer au problème de perte de connexion.

11.2.5 Horizon coupé (Split Horizon)

Une solution à ce problème a été mise de l'avant par le biais de l'algorithme connu sous le nom de Horizon coupé. Cet algorithme ne permet pas aux routeurs de transmettre les informations concernant un réseau par le biais du port d'où elles proviennent. Ainsi, dans notre exemple, le routeur R4 ne transmettra pas au port 192.168.110.2 l'information dont il dispose au sujet du réseau H1 puisque c'est le routeur R2 qui les lui a transmises à l'origine. Cette méthode est connue sous le nom de « *Split Horizon with poisoned reverse* ».

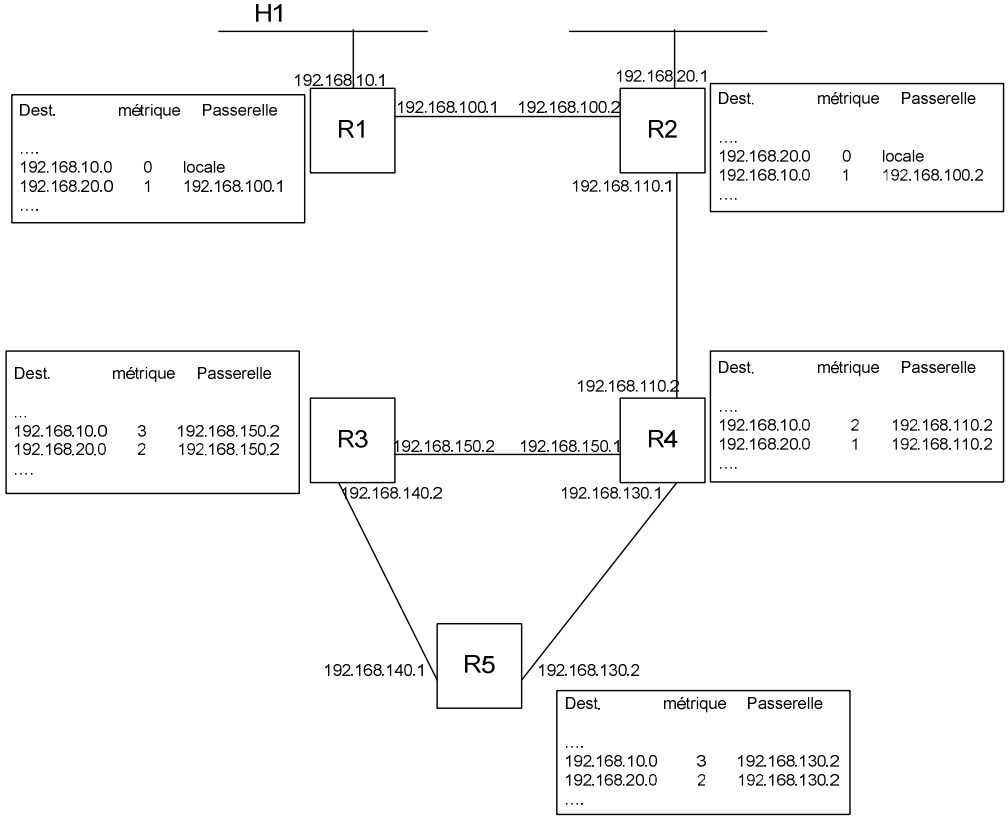


Figure 11.9 Scénario propice au problème de l'horizon coupé

Il est encore possible de rencontrer un problème de convergence avec l'algorithme de l'horizon coupé. Imaginons la situation suivante illustrée dans la figure 11.9. Le réseau H1 devient inaccessible. Le routeur R1 diffuse cette information à R2 en lui transmettant la métrique 16 pour ce réseau. Le routeur R2 ne pouvant obtenir une mise à jour du routeur R4 à cause de l'algorithme de l'horizon coupé va y associer la métrique 16 pour lui indiquer l'infini (∞). Le routeur R5 ne pourra alors pas envoyer au routeur R4 l'information relative au

réseau H1 puisque R4 sert de passerelle pour l'atteindre. Par contre, il pourra transmettre l'information au routeur R3. Ce dernier à son tour pourra en aviser le routeur R4 qui en déduira qu'il devra passer par R3 pour atteindre le réseau H1. La convergence aura vraisemblablement lieu, mais lentement. Nous pouvons remarquer que cette situation survient dans les cas où une boucle existe.

La méthode de déclenchement de la mise à jour (*Triggered updates*) pourrait accélérer le processus de convergence. La règle qui est alors ajoutée aux routeurs réside en ce que lorsque la métrique d'une route est changée, il faut transmettre des messages de mise à jour le plus rapidement possible plutôt que d'y procéder toutes les 30 secondes. Ceci nous permet de transmettre des messages en permanence, et ce, jusqu'à ce que les valeurs des tables du routage restent inchangées soit, jusqu'à ce qu'il y ait convergence.

11.3 *Open Shortest Path First* (OSPF)

OSPF (*Open Shortest Path First*) est un protocole de routage conçu pour un réseau IP plus étendu. Il parvient à contourner les problèmes de convergence du protocole de routage RIP traité dans la section précédente. L'OSPF est énoncé dans le RFC 2178 avec une mise à jour dans le RFC 2328. Ces documents décrivent plus spécifiquement la version 2 de l'OSPF.

Ce dernier est un protocole de routage fondé sur l'algorithme d'état de lien. Les routeurs OSPF s'échangent l'information sur l'état des liens plutôt que de s'informer sur le routage proprement dit. Ces routeurs vont calculer les chemins les plus courts à partir de l'information reçue sur l'état des liens. Cette information est emmagasinée dans une base de données, le LSDB (*Link State Database*).

Chaque routeur prévient ses voisins sur l'état des liens auxquels il est relié. Ces annonces connues sous le nom de LSA (*Link State Advertisement*) sont envoyées à tous les routeurs du réseau. Les routeurs fondent leur stratégie de routage sur la topologie du réseau et procèdent aux calculs, selon l'algorithme de Dijkstra, pour établir le chemin le plus court c'est-à-dire le moins onéreux.

Contrairement au RIP, l'OSPF fonctionne directement au-dessus du protocole IP. Dans le datagramme IP il est défini comme étant le protocole 89. De plus, le champ TOS (*Type of Service*) du datagramme doit être 0. Le fait de ne pas dépendre d'un protocole de transport comme UDP ou TCP le rend beaucoup plus fiable. L'OSPF prend aussi en considération le masque du réseau qui définit l'adresse sous-réseau.

Différemment du RIP qui transmet périodiquement toutes les 30 secondes ses informations en ayant recours au protocole UDP, l'OSPF utilise les messages de multi distribution aux adresses 224.0.0.5 et 224.0.0.6 et épargne ainsi de la bande passante. L'inondation des paquets transmis à tous les routeurs voisins est aussi employée par le protocole OSPF lorsqu'un lien vient à rompre par exemple. Le numéro de séquence figurant dans le paquet OSPF nous permettra de savoir si ce dernier a été reçu plus d'une fois.

La table de routage RIP indique le coût d'accès à un réseau par le nombre de sauts. Le nombre de sauts y est donc limité à 15, vue que la valeur 16 indique l'infini. Par contre, le coût dans le cas des tables de routage de OSPF est généralement basé sur une métrique par défaut calculé selon la formule suivante :

Coût = 10^8/débit en bits/sec

Le nombre de sauts n'est donc pas un problème.

L'environnement réseautique a un impact sur le comportement du protocole de routage OSPF. C'est pourquoi nous présenterons tout d'abord comment OSPF voit cet environnement, et la nature des paquets transmis avant de se pencher sur un exemple d'échange de paquets pour la configuration de table de routage.

11.3.1 Les aires de OSPF

OSPF est utilisé pour des réseaux plus étendus. Toutefois, lorsque le réseau est trop imposant, le LSDB prévoit un grand nombre d'entrées. Une base de données trop importante en ralentira son accès. Ce cas est loin d'être désirable surtout pour un routeur dont on sait que le temps de prise de décision est ce qu'il y a de plus crucial. Dans le cas d'un réseau trop étendu, la solution envisagée repose sur sa répartition en plusieurs sous-réseaux ou « *aires* ».

Chaque aire comporte un nombre de routeurs défini et est identifiée par un numéro unique de 32 bits qui le particularise par rapport aux autres aires du même réseau.

Lorsqu'on a eu recours aux aires, il se trouve une aire dorsale qui représente l'aire centrale et à laquelle toutes les autres aires du réseau sont directement connectées. Le numéro d'identification de l'aire dorsale est toujours 0.0.0.0. Notez que ce numéro ne représente pas une adresse IP. L'ensemble de toutes les aires définies pour ce réseau représente un système autonome (AS).

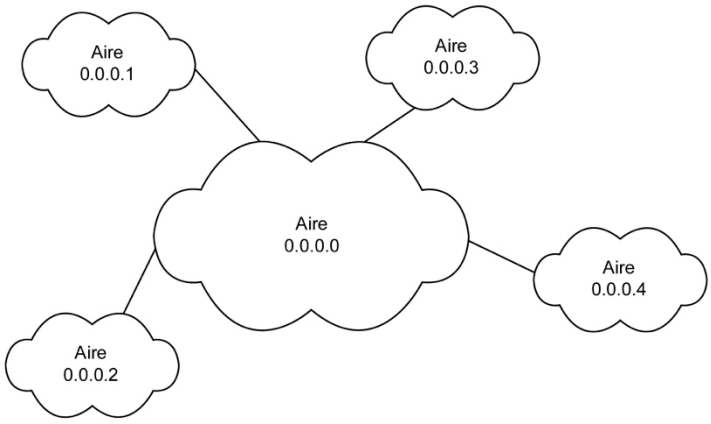


Figure 11.10 Aires dans l'OSPF

Tout comme l'illustre la figure 11.10, chaque aire est reliée à l'aire centrale par un lien. Chaque aire a aussi au moins un de ses routeurs qui est relié à une autre aire ou à l'aire centrale. Ce routeur est qualifié de routeur de bordure d'aire ou d'ABR (*Area Border Router*).

Chaque aire maintient son propre LSDB. Tous les routeurs de l'aire ont le même LSDB et s'échangent des messages pour que leurs bases de données soient en tout point et en tout temps identiques. Le LSDB d'une aire diffère de celui d'une autre. Un routeur relié à d'autres aires, notamment le ABR, doit aussi comprendre le LSDB de chacune des aires à laquelle il est raccordé.

D'autres aires qualifiées de terminales (*Stub Area*) ne font pas transiter des paquets provenant d'une autre aire vers l'aire dorsale. Dans ce cas, toutes les routes menant à l'extérieur de l'aire n'utilisent qu'un seul routeur : le routeur de sortie.

Il existe aussi des aires caractérisées de « pas si terminales » (*Not so Stubby Area*). Tel est le cas d'une aire terminale qui reçoit l'information de routage d'une autre aire (et pouvant faire appel à un différent protocole de routage). Ainsi, l'aire « *pas si terminale* » ne peut pas relayer les informations de routage à l'aire dorsale en indiquant qu'elles provenaient d'une autre aire, et de ce fait elle va les lui transmettre tout comme si elles émanaient d'elle-même

Le concept de contiguïté (*Adjacency*) entre routeurs voisins est primordial pour l'OSPF. Cette contiguïté est établie grâce à l'échange des paquets « Hello ». Ces derniers comprennent les champs « area ID », authentification, masque réseau,

intervalle Hello, et « *Hello dead interval* » qui doivent être identiques pour chacun des deux routeurs adjacents.

11.3.2 Types de réseaux OSPF

L'OSPF est appliqué à plusieurs genres de réseaux. Le classement de ces réseaux se fait d'après leur topologie. Les caractéristiques de OSPF sont distinctes pour chacune d'elles. Brièvement, les catégories de réseaux que l'on peut trouver sont les suivantes :

- Point à point
- Diffusion
- Sans diffusion à accès multiples
- Point à multipoint et
- Lien virtuel

Le **réseau point à point** combine deux routeurs directement connectés par un lien commun. Les routeurs doivent établir leur contiguïté en s'échangeant des paquets et en ayant recours à l'adresse multicast suivante : 224.0.0.5.

Le **réseau à diffusion**, tout comme par exemple les réseaux à compétition ou à jeton que nous avons déjà décrits, se compose de routeurs adjacents les uns par rapport aux autres. Un « routeur désigné » ou DR (*Designated Router*) est sélectionné pour le réseau à diffusion, de concert avec tous les autres routeurs.

Les routeurs de ce réseau échangent leurs informations (LSA) avec le DR seulement et jamais directement avec les autres routeurs. Les informations sur l'état des liens (LSA) en passant par le DR minimisent le nombre d'échanges nécessaires à la concordance des bases de données (LSDB). Si les échanges devaient être accomplis entre tous les n routeurs directement, ils seraient au nombre de n(n-1)/2. Tandis que si le DR recevait toutes les informations directement, il y aurait n-1 échanges soit une économie en termes d'utilisation de largeur de bande.

Un routeur désigné de réserve ou BDR (*Backup Designated Router*) est aussi retenu advenant le cas où le DR venait à tomber en panne. De la sorte, le BDR assumerait la relève dans la mesure où il s'aperçoit que le DR ne diffuse plus d'informations.

Étant donné que tous les routeurs sont adjacents au DR et au BDR, nous pouvons être certains que les bases de données (LSBD) de chaque routeur seront identiques.

Les routeurs simples font appel à l'adresse multicast 224.0.0.6 pour transmettre les informations au DR, alors que ce dernier s'adresse à son tour aux routeurs en utilisant l'adresse multicast 224.0.0.5.

Le **réseau sans diffusion à accès multiples** ou NBMA (*Nonbroadcast Multiaccess*) est un réseau maillé tel l'ATM, le X.25 ou le relais de trames. Les routeurs de ce type de réseau sont dotés d'un DR et d'un BDR pour échanger l'information sur l'état des liens (LSA) et ainsi se prévaloir de LSDB identiques.

Le **réseau point à multipoint** est employé dans un réseau qui n'est pas complètement maillé. Ce fait en rend l'accès au DR bien plus complexe. Dans ce cas-là, la contiguïté est établie tout comme dans celui du point à point, soit entre chacun des liens. Face à une telle situation, le DR n'est tout simplement pas adopté.

Le **lien virtuel** est un réseau permettant, à une aire qui n'est pas directement connectée à l'aire dorsale, d'établir une contiguïté avec ce dernier. Ce lien virtuel se trouve entre deux ABRs ou routeurs en bordure d'aire, ces derniers étant reliés à une autre aire, l'aire de transit. Un lien point à point est alors établi entre les deux ABRs. Notons que l'un des ABRs se trouve sur l'aire dorsale. Ce cas est illustré à la figure 11.11.

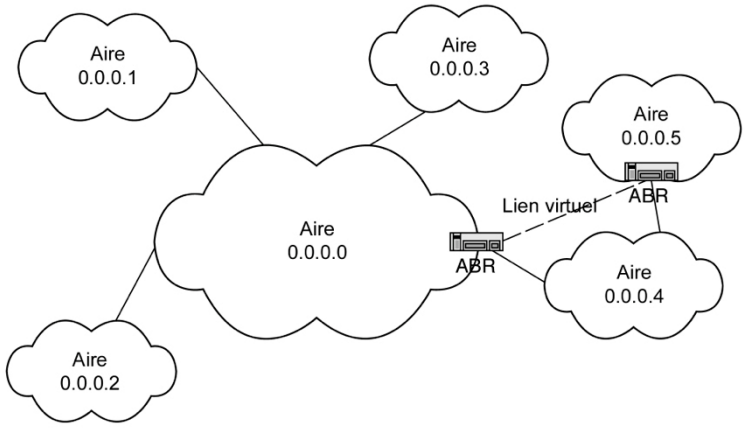


Figure 11.11 Lien virtuel

11.3.3 Les paquets de l'OSPF

Cinq types de paquets sont nécessaires au protocole OSPF. L'en-tête des paquets illustré à la figure 11.12 est global pour tous les types et se compose de 24 octets.

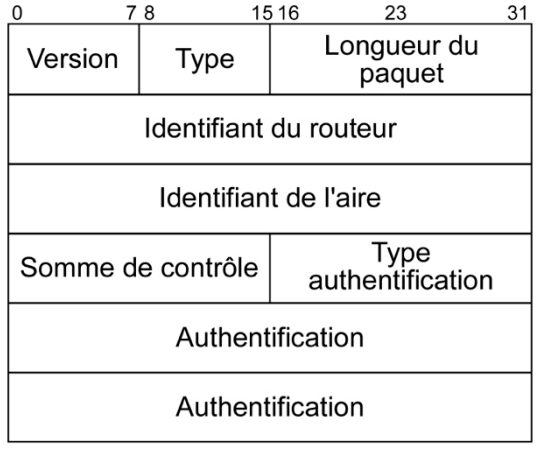


Figure 11.12 En-tête du paquet de l'OSPF

Les champs de l'en-tête sont les suivants :

- Version – la version 2 du protocole OSPF est celle qui a cours à l'heure actuelle.
- Type – tout comme nous l'avons mentionné, il existe 5 types de paquets OSPF. Il s'agit de :
    - 1 – Hello
    - 2 – Description de base de données (*Database Description*)
    - 3 – Demande d'état de liens (*Link State Request*)
    - 4 – Mise à jour d'état de liens (*Link State Update*)
    - 5 – Accusé de réception d'état de liens (*Link State Acknowledgment*)

  Chacun de ces types de paquets sera détaillé plus loin.

- Longueur du paquet – On entend par là la longueur en octets de tout le paquet incluant l'en-tête.
- Identifiant du routeur – Il s'agit ici de l'identifiant du routeur de la source du paquet.
- Identifiant de l'aire – Ce champ identifie l'aire à laquelle appartient ce paquet.
- Somme de contrôle – Champ connu sous le nom anglais de *Checksum*. La somme de contrôle couvre tout le paquet incluant l'en-tête exception faite du champ authentification.
- Type d'authentification – Ce champ identifie la procédure d'authentification qui est utilisée dans le champ suivant.

- Authentification – Les échanges de protocole OSPF sont authentifiés par ce champ.

11.3.3.1 Les paquets Hello

Les paquets Hello sont échangés périodiquement sur toutes les interfaces afin d'établir et de maintenir des rapports de contiguïté entre routeurs voisins. Ces paquets sont diffusés ou propagés sur tous les liens (ou réseaux) afin de permettre la découverte dynamique de routeurs voisins. La figure 11.13 illustre le format du paquet Hello.

Les champs caractéristiques de ce paquet, se trouvant à la suite de l'en-tête, sont les suivants :

- Masque réseau –Masque qui sert à définir l'adresse réseau.
- Intervalle Hello – Cet intervalle indique le nombre de secondes que le routeur peut attendre avant de procéder à la transmission du prochain paquet Hello.
- Options – C'est le même champ qui se retrouve dans le paquet description de base de données. Cinq des huit bits sont utilisés. Ce champ permet à un routeur de rejeter un routeur voisin si les capacités définies par ce dernier ne sont pas compatibles.
- Priorité de routeur – Ce champ permet à un routeur d'être désigné en réserve sauf si ce champ est égal à 0.
- Intervalle mort du routeur – Le contenu de ce champ nous indique le nombre de secondes d'attente avant d'affirmer qu'un routeur est silencieux.
- Routeur désigné – Ce champ nous signale l'identité du routeur désigné pour ce réseau d'après le transmetteur. S'il n'y en a pas, le champ est égal à 0.0.0.0.
- Routeur désigné de réserve – Ce champ renseigne sur l'identité du routeur désigné de réserve pour ce réseau d'après le transmetteur. S'il n'y en a pas, le champ est égal à 0.0.0.0.
- Voisin – Identifiant de chaque routeur duquel un paquet Hello valide a été reçu dans le temps « Intervalle mort du routeur ».

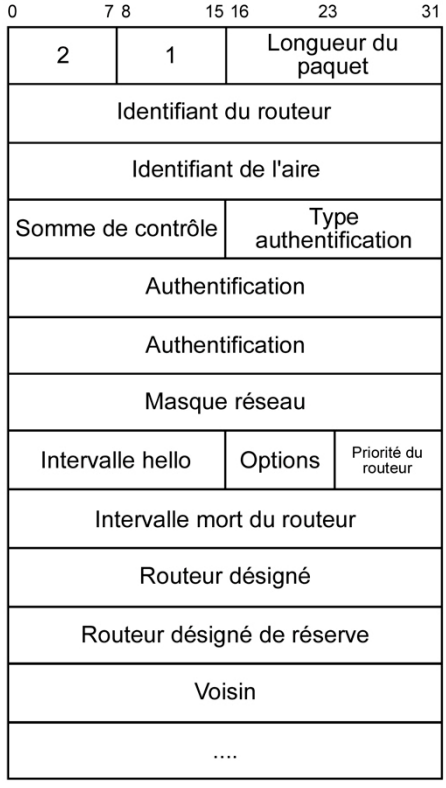


Figure 11.13 Représentation du paquet Hello

11.3.3.2 Les paquets Description de base de données (Database Description)

Les paquets « Description de base de données » sont échangés entre des routeurs établissant une contiguïté. Un bon nombre de ces paquets doivent être échangés pour décrire la base de données de l'état de liens dans son intégralité. La procédure suivie pour cet échange est l'invitation à émettre. L'un des routeurs impliqués dans l'échange est qualifié de « maître », l'autre « d'esclave ». Le routeur maître transmet une interrogation sélective sous la forme du paquet de « Description de base de données (*Database Description*) ». Le routeur esclave y répond par un paquet réponse de « Description de base de données (*Database Description*) ». Ces deux paquets portent le même numéro de séquence qui les relie. La figure 11.14 illustre le format du paquet.

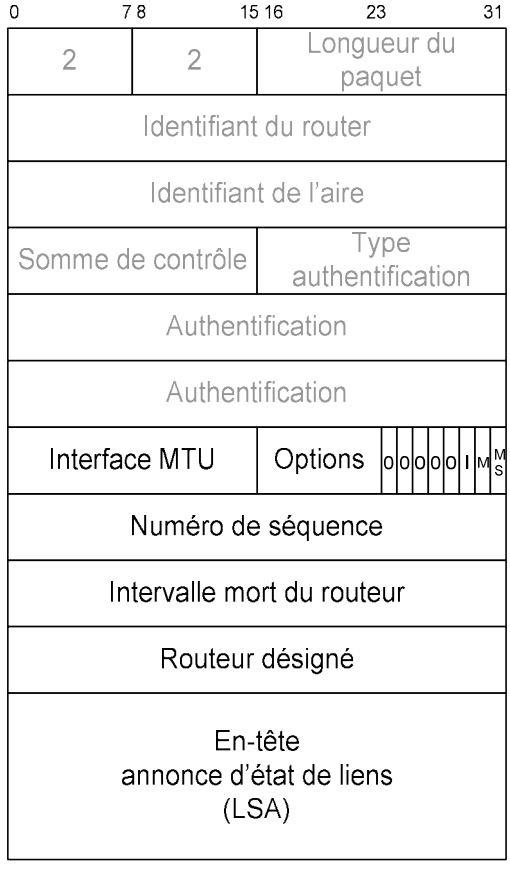


Figure 11.14 Paquet « Description de base de données »

Les champs spécifiques de ce paquet se trouvant à la suite de l'en-tête sont :

- Interface MTU – Ce champ nous signale la grandeur maximale du datagramme IP pouvant être transmis sans fragmentation.
- Options – Ce champ est identique à celui du paquet Hello.
- Bit I – (Bit Init) Lorsque ce bit est à 1 il nous signale que le paquet est le premier d'une série de paquets « Description de base de données ».
- Bit M – (Bit More) Lorsque ce bit est à 1 il nous mentionne que d'autres paquets du même type suivent.
- Bit MS – (Master/Slave) Lorsque le routeur maître transmet ce paquet, le bit MS est à 1, autrement lorsque le routeur est esclave le bit est mis à 0.
- Numéro de séquence – Ce compteur nous permet de conserver la séquence dans laquelle les paquets ont été transmis. La valeur initiale de ce compteur est aléatoire. On sait, grâce au bit I, qu'il s'agit là de la valeur initiale.

- En-tête LSA – Ce champ comprend la liste des éléments de la base de données de l'état de liens. Le format de ce contenu est détaillé plus loin dans ce chapitre.

11.3.3.3 Le paquet « Demande d'état de lien » (Link State Request)

Après les échanges initiaux des paquets « Description de base de données » entre routeurs voisins, l'un de ces derniers pourrait bien devoir procéder à une mise à jour d'une partie de sa base de données. Aussi, c'est grâce au paquet « Demande d'état de lien » ou *Link State Request*, tel qu'illustré à la figure 11.15, que le routeur adresse sa requête spécifique quant aux portions désirées de la base de données.

La demande de chaque annonce d'état de lien LSA (*Link State Advertisement*) est faite selon le type, l'identifiant et le routeur annonceur. Dans le paquet de demandes, seule l'identification des LSA est faite et non le contenu. La demande est faite pour la plus récente des mises à jour.

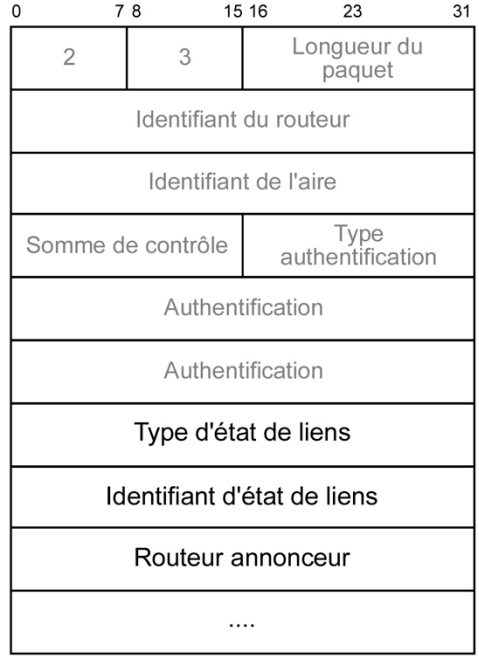


Figure 11.15 Paquet « Demande d'état de liens »

11.3.3.4 Le paquet « Mise à jour d'état de liens » (Link State Update)

En réponse au paquet « Demande d'état de liens », le paquet « Mise à jour d'état de liens » de la figure 11.16 est transmis à tous les routeurs voisins et comprenant la liste des mises à jour d'état de liens demandés. Chaque entrée LSA débute par un en-tête suivi des informations selon le type de LSA.

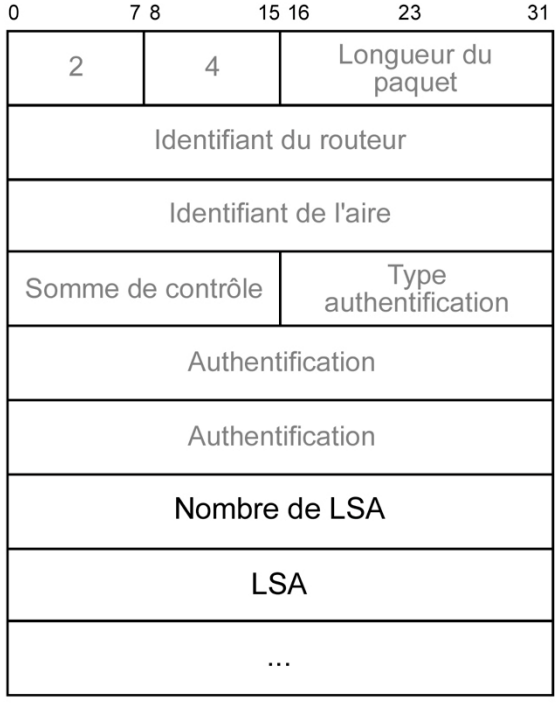


Figure 11.16 Paquet « Mise à jour d'état de liens »

11.3.3.5 Le paquet « Accusé de réception d'état de liens » (Link State Acknowledgment)

Chaque paquet de « Demande d'état de liens » est normalement acquitté par le paquet « Accusé de réception d'état de liens » (*Link State Acknowledgment*). Un seul de ces paquets peut acquitter plusieurs LSA. Seul l'en-tête du LSA acquitté est rapporté par ledit paquet. La figure 11.17 illustre le format du paquet « Accusé de réception d'état des liens ».

Lorsqu'un paquet « Demande d'état des liens » doit aussi être transmis en sens inverse, ce paquet peut servir au titre d'accusé de réception. Il n'est donc pas indispensable de transmettre le paquet « Accusé de réception d'état de liens ».

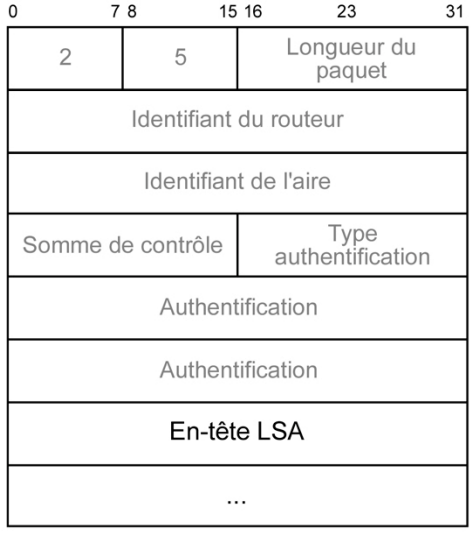


Figure 11.17 Paquet « Accusé de réception d'état des liens »

11.3.3.6 Annonce d'état de liens LSA (Link State Advertisement)

L'information relative à l'état de liens transportés dans les paquets Mise à jour de liens par exemple est représentée sous un format précis. En effet, grâce à cette information, chaque routeur décidera de son routage. Il existe 5 catégories d'annonce d'état de liens ou de LSA :

Type 1	Routeur-LSA
Type 2	Réseau-LSA
Type 3	Sommaire-LSA pour la description des routes vers réseaux IP.
Type 4	Sommaire-LSA pour la description des routes vers les routeurs en bordure.
Type 5	AS-External-LSA pour la description des routes vers des destinations externes au système autonome (AS).

Chaque routeur se trouvant dans une aire génère un type Routeur-LSA compris dans ses paquets. Ce LSA décrit l'état des liens du routeur le connectant à l'aire. Le LSA inonde l'aire en question lorsqu'il est transmis. Tous les liens du routeur sont décrits dans un même paquet.

Lorsqu'un routeur est choisi comme routeur désigné (*Designated Router*) il génère un type Réseau-LSA. C'est le cas de réseau à diffusion et NBMA. Le LSA décrit tous les routeurs (masque et identifiant) qui sont rattachés au réseau incluant le routeur désigné.

Les routeurs qui se trouvent en bordure d'une aire génèrent l'une des deux catégories de Sommaire-LSA pour décrire des destinations inter-aires. Le type 3 Sommaire-LSA est utilisé lorsque la destination est un réseau IP. Lorsque la destination est un routeur de bordure de système autonome (AS), c'est le type 4 Sommaire-LSA qui est utilisé.

Les routeurs en bordure des systèmes autonomes AS (*Autonomous Systems*) génèrent des AS-external-LSA.

Tous les LSA possèdent le même format d'en-tête d'une longueur de 20 octets. La figure 11.18 en donne une illustration.

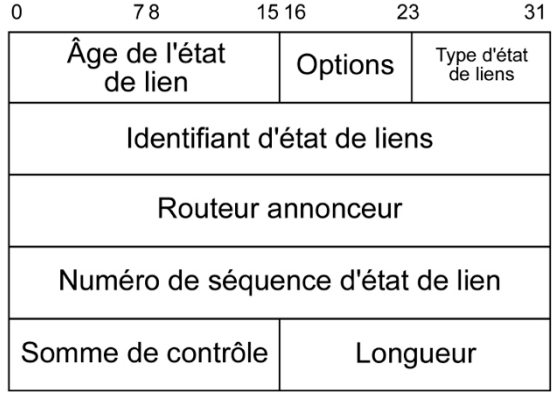


Figure 11.18 En-tête d'Annonce d'état du lien (LSA)

Les champs propres à cet en-tête sont les suivants :

- Âge de l'état de liens – Ce champ nous signale le temps écoulé en secondes depuis la création de ce LSA.
- Options – Ce champ est identique à celui du paquet Hello.
- Type d'état de liens – Indique le type de LSA utilisé.
- Identifiant d'état de liens – Ce champ dépend du genre de LSA.
- Routeur annonceur – Ce champ sert d'identifiant au routeur qui a généré le LSA.
- Numéro de séquence d'état de liens – Dans ce champ, les instances successives sont séquentiellement numérotées.

- Somme de contrôle – Ce champ est connu sous le nom anglais de *Checksum*. Le calcul est réalisé sur le LSA à l'exception du champ Âge de l'état de liens.
- Longueur – Ce champ établit la longueur du LSA en y incluant l'en-tête.

Le numéro de séquence d'état de liens que l'on retrouve dans l'en-tête de chaque LSA pourrait créer une certaine confusion. Le souci est dû au fait que le champ du numéro de séquence a un nombre de bits limité. Il ne faudrait pas que le même numéro de séquence puisse occasionner une confusion dans la transmission des paquets. Une des méthodes qui réglerait ce problème consiste à faire appel à la méthode du vieillissement du paquet, l'autre est celle que l'on qualifie de «méthode à sucette ».

La méthode du vieillissement vise à établir un âge maximum que l'on attribue au LSP (*Link State Packet*). Un routeur recevant ce LSP copie l'âge dans sa base de données en le décrémentant périodiquement. Lorsque le compteur atteint 0, le routeur purge sa base de données de ces LSPs et demande au réseau d'en faire autant en mettant l'âge à 0 et en diffusant ses LSPs.

La «méthode à sucette» est une meilleure solution pour les routeurs qui sont réinitialisés en cours de route. Les numéros de séquence créés par le routeur réinitialisé et transmis dans les LSPs devront être différenciés de ceux employés couramment dans la base de données des routeurs récepteurs. Cette distinction ne peut pas être clairement établie avec la méthode du vieillissement. L'espace de séquence à sucette[3] tel qu'illustré à la figure 11.19 nous en donne un bon exemple.

[3] R. Perlman. « *Fault-Tolerant Broadcast of Routing Information* », Computer Networks, Vol.7, 1983, pp.395-405.

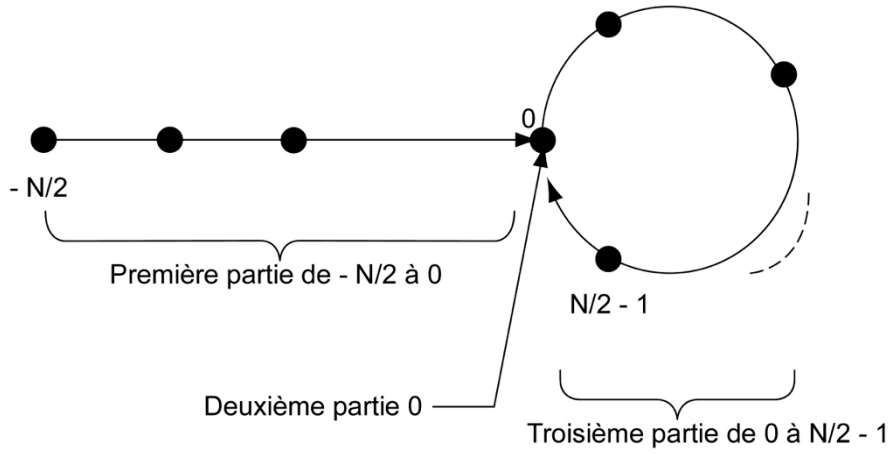


Figure 11.19 Méthode qualifiée « d'espace de séquence à sucette »

Dans cette méthode l'espace de séquence de grandeur N est subdivisé en trois : La première, négative et allant de –N/2 à 0, la seconde se trouvant dans le numéro de séquence 0, et la troisième allant de 0 à N/2 – 1.

Lorsqu'un routeur est initialisé, il démarre avec un numéro de séquence négatif -N/2 attribué au premier LSP. Ce routeur numérote les LSPs suivants en leur imputant les valeurs -N/2+1, -N/2+2 etc. Lorsque les numéros de séquence deviennent positifs ils sont incrémentés modulo N/2. Cela revient à dire que les numéros de séquence vont de 0 à N/2 – 1 et recommence de 0 à N/2 – 1.

Un LSP ayant un numéro de séquence *a* est considéré comme étant plus âgé qu'un LSP dont le numéro de séquence est *b* si :

- $a < 0$ et $a < b$, ou
- $a > 0$, $a < b$, et $b - a < N/4$ ou
- $a > 0$, $b > 0$, $a > b$, et $a - b > N/4$

Les chiffres négatifs tout particulièrement – N/2 représentent des numéros de séquence parmi les plus anciens.

Ainsi, lorsqu'un routeur reçoit un LSP avec un numéro de séquence négatif d'un autre routeur qui vient d'être réinitialisé, le numéro de séquence figurant dans sa base de données étant plus récent, il le conserve et l'envoie aux autres routeurs (y compris celui que lui a transmis le LSP avec un numéro de séquence négatif). Cela s'avère être fondé pour tous les numéros de séquences reçus plus anciens que celui apparaissant dans la base de données du routeur.

Un simple exemple illustre ce mécanisme. Si N=8, le routeur qui est réinitialisé entame ses numéros de séquence à partir de -4 et continue jusqu'à 0. Après cela, le compte sera amorcé de 0 jusqu'à 3 puis l'on reprendra de 0 à 3 sans arrêt.

Un routeur qui reçoit un LSP avec le numéro de séquence -4 et qui a dans sa base de données pour cette même entrée le numéro de séquence 2; va conserver ce dernier et l'adresser à tous les autres routeurs. Nous donnons une illustration de ce cas à la figure 11.20.

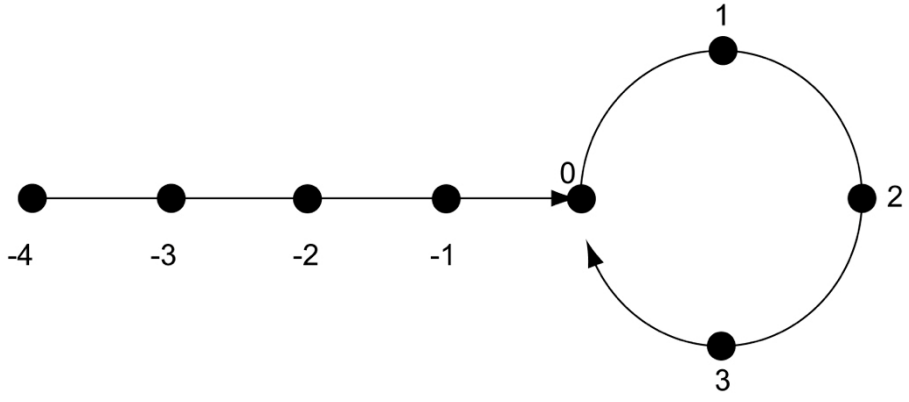


Figure 11.20 Exemple de méthode « espace de séquence à sucette »

11.3.4 La machine d'état des routeurs voisins

Le protocole OSPF amorce une procédure identifiée par l'expression de « protocole Hello ». Ce protocole est responsable du démarrage de l'échange d'états de liens et il définit la communication entre voisins.

Le diagramme d'état de la figure 11.21 illustre les étapes suivies au cours de l'exécution de ce protocole.

- On débute par l'état *DOWN* où aucune contiguïté n'existe entre les routeurs voisins.
- L'état *Attempt* ne prévaut que pour les voisins rattachés au type de réseau sans diffusion à accès multiples ou NBMA. Ceci nous signale qu'il y a peu d'activités de détectées et que le paquet Hello est transmis aux routeurs voisins à un intervalle égal à *HelloInterval*.
- *INIT* nous prévient qu'un paquet Hello a été reçu, mais que la communication avec le routeur voisin n'est pas encore bidirectionnelle.
- *2-Way* est l'état nous informant que la communication entre les routeurs voisins est bidirectionnelle.

- *ExStart* apparaît comme étant la première étape lors de la création de la contiguïté entre les routeurs voisins. C'est à ce stade que l'on apprend lequel des routeurs sera le routeur « Maître », en se fondant d'une part sur le numéro d'identification le plus élevé, et de l'autre, sur le numéro de séquence que le « Maître » assigne.

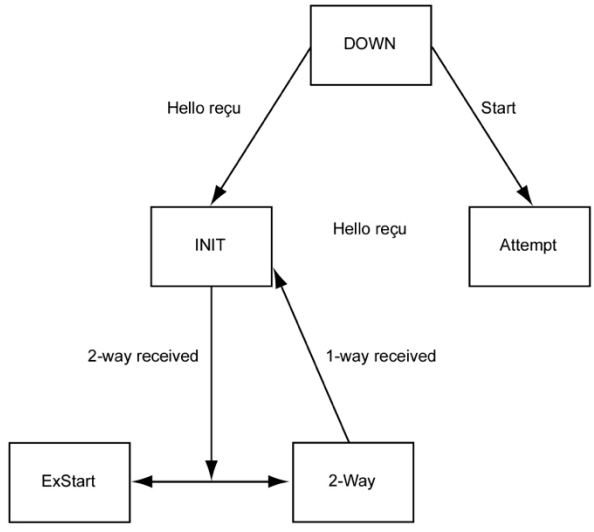


Figure 11.21 Changement d'état du routeur voisin (Protocole Hello)

Une fois la contiguïté amorcée, la négociation est considérée comme étant établie. L'étape suivante illustrée à la figure 11.22 réside dans l'échange de base de données.

- L'état *Exchange* indique que les routeurs voisins s'échangent des paquets « Description de base de données ».
- *Loading* est l'état où des paquets « Demande d'état de liens » sont transmis au routeur voisin pour obtenir des mises à jour du LSA.
- *Full* est l'état qui nous apprend que les routeurs voisins ont établi une contiguïté complète.

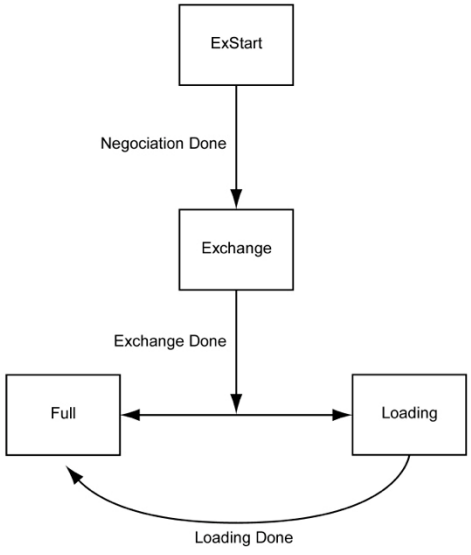


Figure 11.22 Changement d'état du routeur voisin

11.3.5 Exemple d'établissement de contiguïté

L'exemple suivant représenté à la figure 11.23 figure dans le RFC 2328. Il décrit l'établissement de la contiguïté entre deux routeurs. Ces derniers se trouvent dans un réseau du type diffusion (*broadcast*). Le routeur R2 est le routeur désigné pour ce réseau et son numéro d'identification est plus élevé que celui du routeur R1.

Les diagrammes d'états tels que traités dans la section précédente sont appliqués à notre exemple et chaque état y est clairement mentionné au fur et à mesure de son utilisation.

Tout commence au moment où les routeurs R1 et R2 ne sont pas visibles l'un par rapport à l'autre et sont de ce fait, tous deux, à l'état *DOWN*. Le routeur R1 devient opérationnel et diffuse des paquets Hello sur le réseau. Dans ce paquet, il nous signale qu'il ne connaît pas le routeur désigné (DR), ni ne voit de routeur voisin.

Le routeur R2 reçoit les paquets Hello de R1. Il passe alors à l'état *Init* et transmet à son tour des paquets Hello en s'identifiant comme DR et en signalant que son voisin est le routeur R1.

Lorsqu'il reçoit ces paquets, le routeur R1 passe à l'état *ExStart*. Il transmet initialement (bit I) un paquet « Description de base de données » (D-D) en y indiquant par le bit M que d'autres suivront, et par le bit MS qu'il est le « Maître ». Il propose aussi un numéro de séquence pour les paquets D-D.

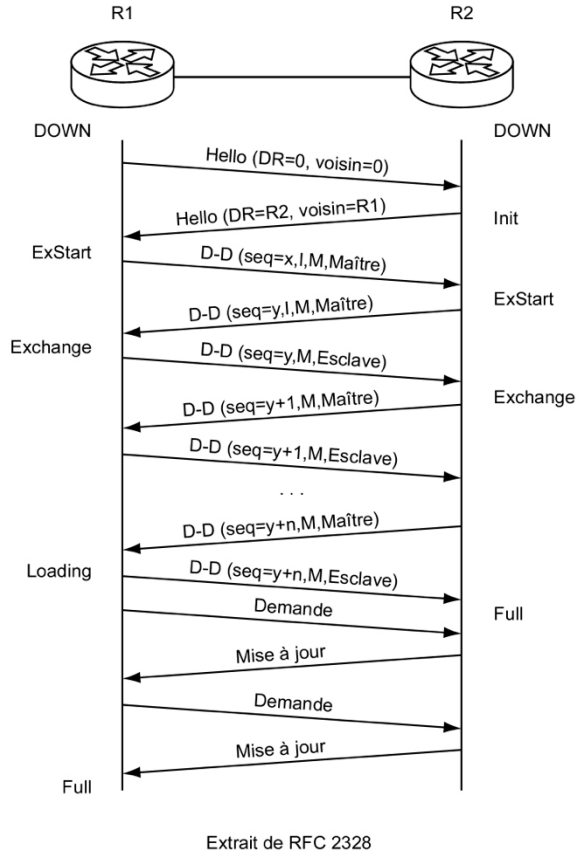


Figure 11.23 Établissement de la contiguïté entre deux routeurs

Lorsque le routeur R2 reçoit le paquet D-D du routeur R1, il passe à son tour à l'état *ExStart* et lui renvoie un nouveau paquet D-D mais en s'identifiant comme « Maître », puisque son numéro d'identification est plus élevé que celui de R1. Il y inclut également le numéro de séquence choisi.

Le routeur R1 cède le contrôle à R2, passe à l'état *Exchange*, transmet un paquet D-D en s'identifiant comme «esclave» et adopte le numéro de séquence proposé par R2. En répétant le même numéro de séquence, le paquet de R2 est directement acquitté.

Sur réception du paquet D-D de R1, le routeur R2 passe aussi à l'état *Exchange*. Il transmet alors le paquet D-D suivant en incrémentant le numéro de séquence.

Cet échange de paquets D-D se poursuit jusqu'au temps où toutes les entrées de la base de données sont résolues. Soulignons que le bit M de tous les paquets D-D intermédiaires est égal à 1.

Le routeur R2 parvient à l'état *Full* attendu qu'il a complété la mise à jour de sa table du routage. Quant au routeur R1, il peut faire des demandes complémentaires pour mettre à jour sa propre table de routage. Ce processus s'accomplit grâce à la transmission des paquets « Demande d'état des liens ». Le routeur R2 qui reçoit ces paquets répond par des paquets « Mise à jour d'état des liens ». Ces derniers peuvent être acquittés par R1 en renvoyant des paquets « Accusé de réception d'état des liens » ou en transmettant une nouvelle demande. Lorsque le routeur R1 a achevé sa mise à jour, il atteint l'état *Full*.

11.3.6 Fonctionnement de OSPF

OSPF peut fonctionner de façon autonome à l'intérieur d'une aire. Chaque routeur de cette aire fait connaître à ses voisins et, par leur biais, aux autres routeurs, sa table de routage. C'est par la connaissance de la vue d'ensemble du réseau que chaque routeur calcule les chemins les plus courts vers chacun des autres routeurs. L'algorithme utilisé pour ce calcul est celui de Dijkstra (section 11.5).

Les routeurs du réseau sont initialement activés au même moment. Chaque routeur comprend deux mémoires. La première, *ROUTAGE*, comprend les chemins les plus courts connus vers chacun des routeurs de l'aire. La première entrée dans *ROUTAGE* est celle du routeur lui même avec un coût de zéro. La deuxième mémoire *ITÉRATION*, comprend les coûts connus mais pas nécessairement plus petits pour atteindre chaque autre routeur. Ainsi, les premières entrées dans *ITÉRATION* sont les coûts des liens auxquels le routeur est directement connecté. Chaque routeur diffuse au départ, cette information aux autres routeurs. Ainsi chaque routeur connaît l'état de liens du réseau.

Considérons la figure 11.24 comme cas de figure. Les premiers messages diffusés par chaque routeur correspondent à chaque ligne de la table suivante :

De	à	Coût	à	Coût	à	Coût	à	Coût	à	Coût
1	2	2	3	5	4	1				
2	1	3	3	3	4	3				
3	1	8	2	6	4	3	5	1	6	5
4	1	7	2	2	3	3	5	1		
5	3	1	4	1	6	4				
6	3	8	5	4						

L'information comprise dans cette table sera donc connue de tous les routeurs de l'aire.

Dans chaque routeur, le contenu de chacune des mémoires *ROUTAGE* et *ITÉRATION* s'effectuera de façon itérative.

L'application de l'algorithme par le routeur 1 sera prise à titre d'exemple.

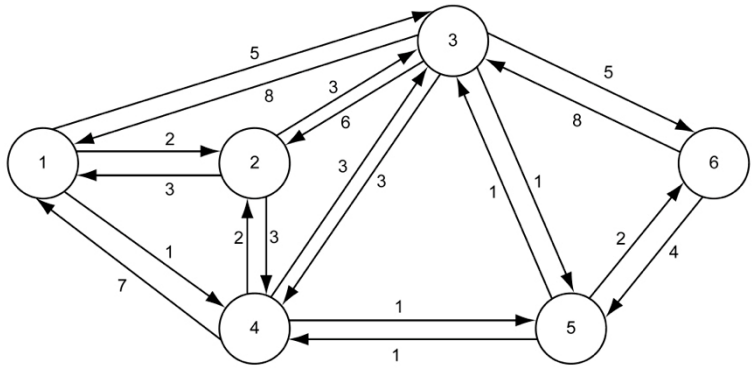


Figure 11.24 Aire OSPF

Dans la **première itération** nous avons les valeurs suivantes dans chacune des deux mémoires:

ROUTAGE

Routeur	Coût	Next
1	0	

ITÉRATION

Routeur	Coût
2	2
3	5
4	1

Notons que *ROUTAGE* donne le chemin vers un routeur en indiquant le coût ainsi que le prochain routeur pour l'atteindre. Dans cette première entrée il s'agit du routeur source. Il n'y a donc pas de prochain routeur.

Le plus petit coût dans la mémoire *ITÉRATION* correspond au routeur 4. Celui ci est donc déplacé dans la mémoire *ROUTAGE*.
Le nouveau contenu de la mémoire *ITÉRATION* comprendra le coût total pour atteindre les voisins du routeur 4. Si le coût total pour atteindre un routeur est plus petit à partir du routeur 4, il remplacera l'entrée précédente. Ainsi, à la **deuxième itération** du routeur 1 le contenu des mémoires est :

ROUTAGE			*ITÉRATION*	
Routeur	Coût	Next	Routeur	Coût
1	0		2	2
4	1	4	3	4
			5	2

Notons que le coût pour atteindre le routeur 3 est maintenant 4 au lieu de 5. Ceci est déduit par le fait que le chemin de 1 à 3 est moins couteux en passant par 4.

À la **troisième itération** l'entrée du routeur 2 est déplacée de la mémoire *ITÉRATION* à *ROUTAGE* parce qu'elle a le plus petit coût (le routeur 5 aurait pu tout aussi bien être choisi).

ROUTAGE			*ITÉRATION*	
Routeur	Coût	Next	Routeur	Coût
1	0		3	4
4	1	4	5	2
2	2	2		

À la **quatrième itération** l'entrée du routeur 5 est déplacée de la mémoire *ITÉRATION* à *ROUTAGE* parce qu'elle a le plus petit coût. La mémoire *ITÉRATION* comprendra les voisins de 5 et remplacera les entrées existantes si le coût est plus petit.

ROUTAGE			*ITÉRATION*	
Routeur	Coût	Next	Routeur	Coût
1	0		3	3
4	1	4	6	4
2	2	2		
5	2	4		

Le processus est complété une fois que la mémoire *ITÉRATION* est vide. À la **sixième itération** nous aurons donc :

ROUTAGE			*ITÉRATION*	
Routeur	Coût	Next	Routeur	Coût
1	0			
4	1	4		
2	2	2		
5	2	4		
3	3	4		
6	4	4		

C'est le contenu de la mémoire *ROUTAGE* qui représente la table de routage OSPF.

Il est évident que l'aboutissement de ce processus ne peut avoir lieu que si tous les routeurs sont démarrés en même temps. Ceci est important parce que les coûts de tous les liens du réseau doivent être connus de tous.

Naturellement, ceci n'est pas toujours possible. C'est pourquoi un des routeurs est élu *nœud désigné*. Le routeur qui est mis en service après la stabilisation des routes doit s'adresser au *nœud désigné* pour obtenir ses mises à jour.

La table de routage est sujette à des mises à jour périodique pour indiquer l'état de changement de liens.

11.4 *Border Gateway Protocol* (BGP)

RIP et OSPF sont des protocoles de routage internes. Ils sont utilisés pour acheminer des datagrammes dans un réseau autonome (AS) contrôlé essentiellement par un fournisseur de service. Lorsqu'il s'agit de transférer des datagrammes d'un réseau autonome à l'autre, géré par différents fournisseurs de service, on a recours à un protocole de routage plus adapté, notamment Border Gateway Protocol (BGP). Le BGP version 4 est défini dans le RFC 4271 avec des ajustements dans les RFC 4274 et 4276.
BGP est un protocole inter-AS utilisé à travers la toile. Sans BGP, Internet ne serait pas aussi répandu et efficace. C'est par contre un protocole complexe.
Chaque système autonome comprend des routeurs de bordure. Ces routeurs, comme leur nom l'indique, sont en bordure du AS et offre une passerelle d'un AS à un autre. Le protocole de routage utilisé entre ces routeurs est le BGP et plus précisément le eBGP, c'est-à-dire le protocole BGP externe. Les informations de routage sont échangées à travers une connexion TCP (voir

chapitre 12) semi-permanente utilisant le port 179. Cette connexion est établie entre pairs, comme toute connexion TCP, impliquant les deux routeurs de bordure adjacents, chacun appartenant à un AS différent. D'autre part, chaque AS comprend des routeurs internes qui ne sont pas de bordure et qui s'échangent des informations de routage BGP par le biais de connexion TCP. Dans ce cas, les sessions iBGP établies entre les routeurs de l'AS sont des sessions BGP internes. L'application de iBGP est pertinente lorsque les AS sont interconnectés. Ceci est illustré dans la figure 11.25.
Notons que TCP établit des sessions de bout en bout entre pairs. Ainsi chaque pair de routeurs qui s'échangent les informations BGP le fait à travers une session TCP. Le routeur intermédiaire extrait les informations d'une session en amont et les transferts à la session établie en aval avec un autre routeur. C'est ainsi que les informations de routage seraient réparties à travers Internet.

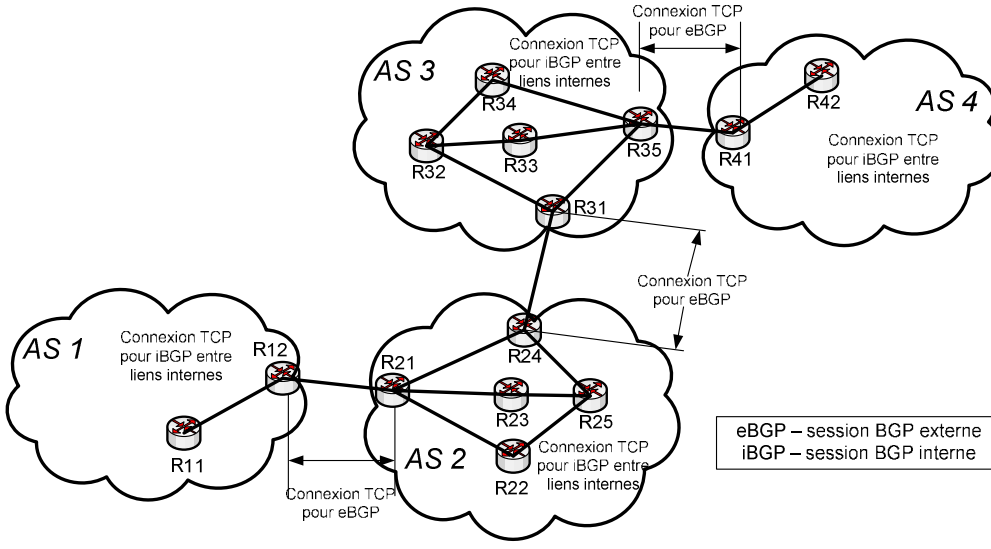


Figure 11.25 Environnement BGP

Les tables de routage propre à BGP sont réparties et distribuées de AS en AS. Le contenu de ces tables de routage représente les destinations globales en termes de préfixes CIDR. Ces préfixes représentent les sous réseaux tels que définis au chapitre 9. L'agrégation des routes peut donc permettre de réduire l'entrée des tables de routage BGP. Un routeur qui serait la passerelle de réseaux et de plusieurs de ses sous réseaux pourrait utiliser un préfix réseau plus petit suffisant pour tous les représenter.

Il est à noter qu'un AS avise un autre AS des sous réseaux qui lui sont accessibles. Cette information est passée par le biais d'une connexion TCP pour

eBGP. Le AS qui la reçoit, la distribue à ses propres routeurs en utilisant le protocole iBGP. Ainsi, dans le cas de figure illustré dans la figure 11.24, le routeur R24 établi une session TCP avec le routeur R31 et lui transmet les routes qui lui sont disponibles pour accéder ses sous réseaux et ceux d'autres AS qui lui ont été transmis. Le routeur R31 à son tour distribuera cette information à ses routeurs avec le protocole iBGP. Le routeur R35 pourra ainsi transmettre cette information au AS4 en établissant une session eBGP par le biais d'une connexion TCP.

Les informations distribuées par le protocole BGP comprennent aussi un nombre d'attributs. Ainsi un préfixe qui est indiqué dans la table de routage avec ses attributs représente une route. Les attributs les plus importants sont AS-ROUTAGE et NEXT-HOP.

Il est important de noter que les systèmes autonomes (AS) sont généralement dotés d'un numéro de système autonome ou « Autonomous System Number » (ASN) qui lui est assigné par l'ICANN (Internet Corporation for Assigned Names and Numbers). Ces ASNs sont uniques (RFC 1930). Certains AS n'ont pas de ASN. Ce sont les AS qui ne transportent que le trafic qui leur est destiné ou pour lequel ils sont la source. Ces AS sont connus sous le nom de *stub AS*.

L'attribut AS-ROUTAGE comprend la liste des routes représentées par le préfixe des sous réseaux et le ASN auquel ils appartiennent. La première route mentionnée est la plus proche suivie des indications de routes graduellement distantes. Lorsqu'il existe plusieurs chemins vers un sous réseau, la route préférée est indiquée. Le choix de la route est fait suivant le processus connu sous le nom de « *hot-potato routing* ». Dans ce contexte, le routeur qui a un choix de routes vers une destination transmet le paquet le plus rapidement possible (concept de hot-potato) au routeur de bordure dont l'accès est le moins coûteux en termes de la pondération qui est utilisée. D'autre part, il peut y avoir des politiques qui sont appliquées dans le choix de route basée sur des critères administratifs. Dans ce cas c'est l'administrateur de réseau qui applique ses règles dans le choix de routes.

L'attribut NEXT-HOP représente l'adresse IP de l'interface d'accès du routeur de bordure qui transmet l'information de routage.

Il est clair que cette section ne fait qu'un survol d'un protocole très complexe qu'est BGP.

11.5 *Algorithmes de routage*

11.5.1 Algorithme de Dijkstra

Les routeurs ayant recours au protocole OSPF doivent calculer le chemin le plus court à partir de la table de routage telle qu'établie. Ce calcul se fait d'après l'algorithme de Dijkstra qui est du type d'algorithme de routage d'état de lien (*Link State Routing Algorithm*). C'est un algorithme itératif. À titre d'exemple, commençons par un réseau illustré à la figure 11.26 dont chaque embranchement, et chaque direction comprend un coût associé. Ce coût peut être fonction de la distance ou de la congestion.

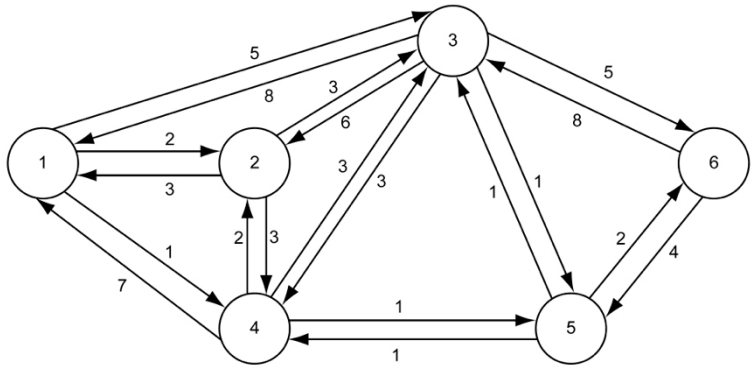


Figure 11.26 Réseau utilisé à titre d'exemple

L'objectif visé est de calculer le chemin le plus court en partant d'un nœud donné vers n'importe lequel des autres nœuds du réseau. Ce processus se fait à partir de plusieurs itérations.

Ainsi il s'agit de :

- Créer une arborescence à partir des connexions du réseau
- Définir les branches les moins coûteuses c'est-à-dire le trajet complet le moins coûteux.

Aux fins du présent exemple, nous allons identifier un nœud de départ, le nœud 1, donc *S=1*. Nous définissons aussi un vecteur *M* qui inclut seulement les nœuds que l'on peut atteindre par un lien sélectionné. Pour la première itération (figure 11.27) *M* = {*1*} (1 étant le nœud sélectionné dans le vecteur). Les branchements possibles sont identifiés par des lignes pointillées et le coût de ces lignes est identifié par *Dx*.

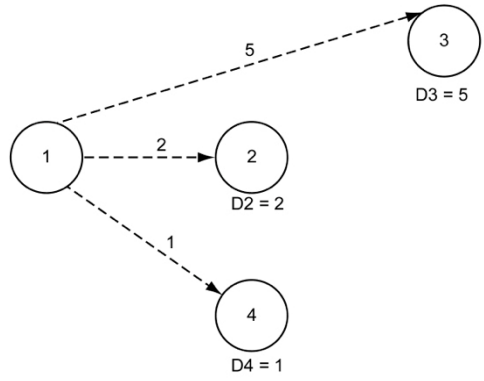


Figure 11.27 Première itération (Dijkstra)

Des trois lignes sélectionnées précédemment nous prenons la moins chère (du nœud 1 à 4) et nous traçons une ligne ferme. Puis nous cherchons à savoir s'il existe une autre ligne pour atteindre les nœuds adjacents de la façon la moins chère. S'il en existe, on remplace les lignes déjà envisagées, on les retient et on élimine celles qui seront remplacées. Pour la deuxième itération, $M = \{1,4\}$ (figure 11.28). Ceci inclut seulement les nœuds que l'on peut atteindre par le lien sélectionné soit, les nœuds 1 et 4.

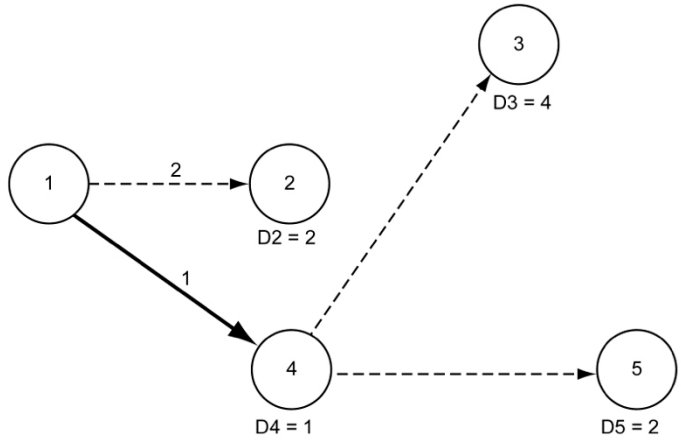


Figure 11.28 Deuxième itération (Dijkstra)

À la troisième itération (figure 11.29) le nœud 2 est inclus dans le vecteur M, $M = \{1,2,4\}$. Les nœuds qui sont reliés au nœud 2 peuvent être atteints, par contre, par le nœud 4 avec un coût plus bas.

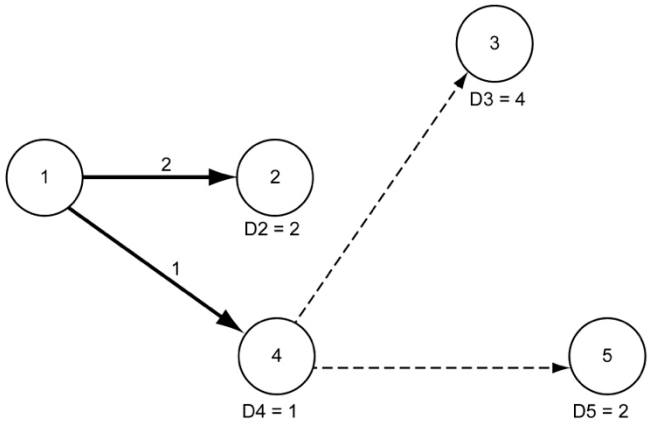


Figure 11.29 Troisième itération (Dijkstra)

Toutefois, il est plus court pour la source d'atteindre le nœud 2 directement (*coût de 2*) que de passer par le nœud 4 *(coût de 3*).

Durant la prochaine itération, le nœud 5 est ajouté au vecteur M, (figure 11.30) *M* = {*1,2,4,5*}. Le chemin le plus court pour atteindre le nœud 5 part du nœud 4. D'autre part, un nouveau chemin plus court au nœud 3 est trouvé par le biais du nœud 5. De la sorte, les nœuds qui peuvent être atteints à partir du nœud 5 sont identifiés.

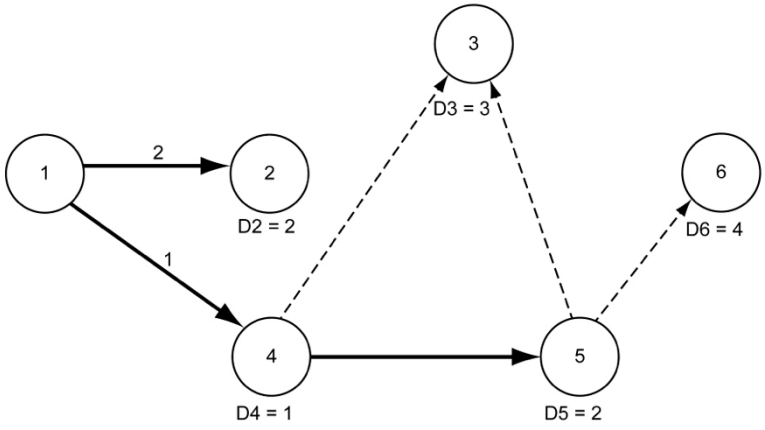


Figure 11.30 Quatrième itération (Dijkstra)

Les itérations qui suivent, incluent les nœuds 3 et 6 de même que les chemins les plus courts qui y mènent (figure 11.31). Le vecteur qui en résulte est *M* = {*1,2,3,4,5,6*}.

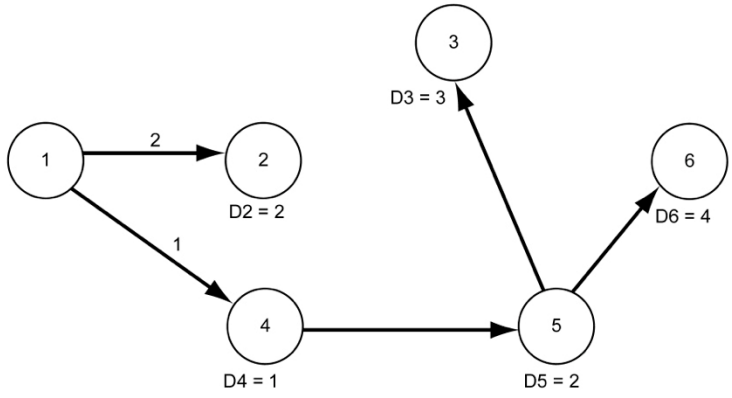


Figure 11.31 Sixième itération (Dijkstra)

La représentation des itérations de notre exemple est illustrée à la figure 11.32. Cette figure indique les vecteurs de même que les chemins et les coûts qu'ils représentent.

Itération	M	D2	chemin	D3	chemin	D4	chemin	D5	chemin	D6	chemin
1	{1}	2	1-2	5	1-3	1	1-4	inf	-	inf	-
2	{1,4}	2	1-2	4	1-4-3	1	1-4	2	1-4-5	inf	-
3	{1,2,4}	2	1-2	4	1-4-3	1	1-4	2	1-4-5	inf	-
4	{1,2,4,5}	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
5	{1,2,3,4,5}	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
6	{1,2,3,4,5,6}	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6

inf – infinité

Figure 11.32 Représentation des itérations

L'algorithme de Dijkstra est formellement défini par les étapes suivantes :

1 – Le vecteur M comprend en premier le nœud source pour la première itération. Nous tentons d'établir le coût du nœud n qui est directement rattaché à la source, c'est-à-dire au nouveau contenu du vecteur M.

$$M = \{S\}$$

$$D_n = d_{sn} \quad \text{pour } s \neq n$$

S - source.
D_n - coût source-destination n.
d_{in} - coût du lien du nœud i au nœud n.

2- Trouver le nœud voisin ζ ne se trouvant pas dans le vecteur M ($\zeta \notin M$), mais étant directement relié au dernier nœud inclus dans ce vecteur. Calculer le chemin le plus court du nœud ζ aux autres nœuds j qui lui sont rattachés, mais qui ne se trouvent pas dans le vecteur M ($j \notin M$).

Si $\zeta \notin M$,

pour $D_\zeta = \min_{j \notin M} D_j$

Le nœud ζ est par la suite inclus dans le vecteur M

$M = \{ S, \zeta \}$

3- Déterminer, lors de chacune des itérations, le chemin le plus court du nœud S au nœud n. Ce chemin peut être celui qui a déjà été calculé D_n ou le nouveau chemin comprenant le nouveau nœud $D_\zeta + d_{\zeta n}$ s'il existe.

$D_n = \min [D_n , D_\zeta + d_{\zeta n}]$ pour tous $n \notin M$

Si le deuxième terme $D_\zeta + d_{\zeta n}$ est le plus court, le chemin de S à n est alors le chemin de S à ζ concaténé au chemin de ζ à n. Il est bon de souligner que le chemin de S à ζ n'est pas nécessairement un chemin direct.

4- Les étapes 2 et 3 sont reproduites jusqu'à ce que tous les nœuds du réseau soient inclus dans le vecteur M.

11.5.2 Algorithme de Bellman-Ford

À l'instar de l'algorithme de Dijkstra, celui de Bellman-Ford se propose de calculer le chemin le plus court dans un réseau. Il est toutefois basé sur l'algorithme de routage de vecteur de distance (Distance-Vector Routing Algorithm). C'est aussi un algorithme asynchrone, distribué et itératif. Il est à la base du protocole de routage RIP.

L'exemple illustré à la figure 11.26 utilisé pour l'illustration du fonctionnement de l'algorithme de Dijkstra, sera repris dans la figure 11.33 pour démontrer le fonctionnement de l'algorithme de Bellman-Ford. Ainsi, chaque embranchement, et chaque direction du réseau comprennent un coût associé. Ce coût peut être fonction de la distance ou de la congestion.

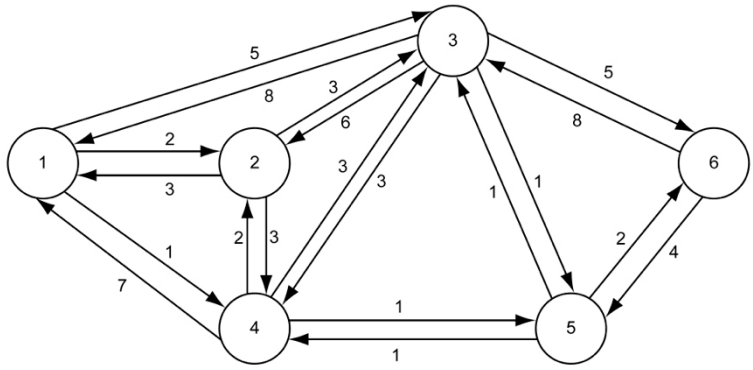


Figure 11.33 Réseau utilisé à titre d'exemple

L'objectif visé est de calculer le chemin le plus court à partir d'un nœud donné vers les autres nœuds du réseau en augmentant itérativement le nombre de liens pour atteindre chacun des nœuds.

Ainsi il s'agit de :

- Créer une arborescence à partir des connexions du réseau
- Définir les branches les moins coûteuses c'est-à-dire le trajet complet le moins coûteux.

Aux fins du présent exemple, nous allons identifier un nœud de départ, le nœud 1, par exemple. Nous procédons par la première itération qui identifie les nœuds qui sont accessibles à partir du nœud 1 par un saut. Pour la première itération (figure 11.34) $h = \{1\}$. Les branchements possibles sont identifiés par des lignes et le coût de ces lignes est identifié par $D_x^{(h)}$.

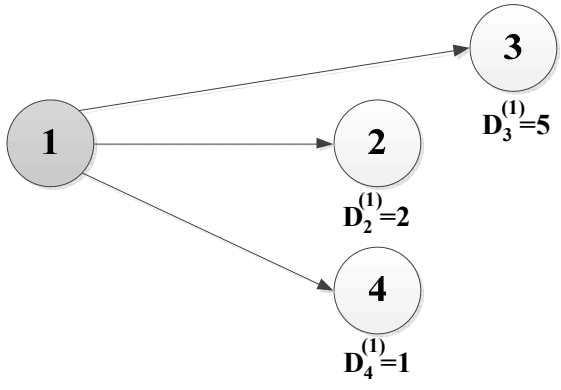


Figure 11.34 Première itération (Bellman-Ford)

Ainsi le nœud 1 peut accéder aux nœuds 2, 3 et 4 directement. Les coûts associés à ces chemins sont indiqués par $D_2^{(1)}=2$. $D_3^{(1)}=5$ et $D_4^{(1)}=1$.

À la deuxième itération *h*={*2*}. Il s'agit d'identifier les nœuds qui sont accessibles à partir du nœud source, notamment le nœud 1, en deux sauts. La figure 11.35 illustre le résultat. Le but n'est pas seulement d'identifier l'accès par deux sauts, mais aussi de trouver un chemin plus court que celui identifier lors de l'itération précédente. On remarque que l'accès au nœud 3 est moins coûteux en passant par le nœud 4 (coût total de 4) que l'accès direct qui avait été trouvé lors de la première itération (coût de 5).

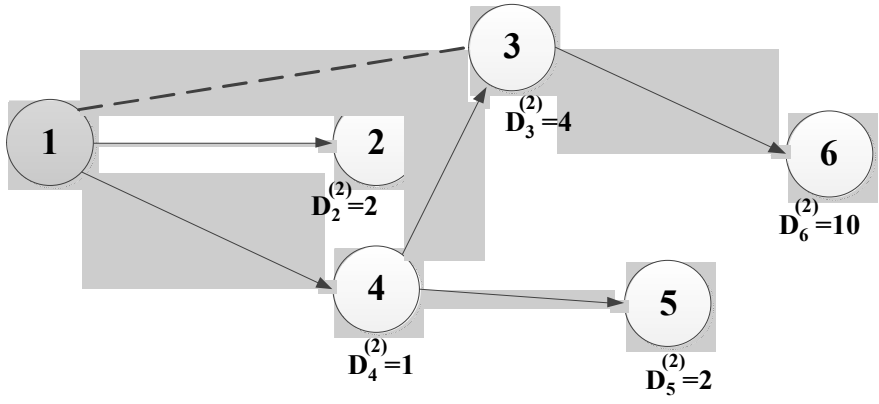


Figure 11.35 Deuxième itération (Bellman-Ford)

La troisième itération *h*={*3*}, comme on peut le prévoir, est celle de l'accès à trois sauts à partir de la source. Les coûts de chemins menant au même nœud sont comparés avec l'itération précédente et le plus petit est choisi pour représenter le chemin le plus court vers ce nœud. Dans ce cas de figure, la troisième itération est la dernière vu que tous les nœuds sont couverts.

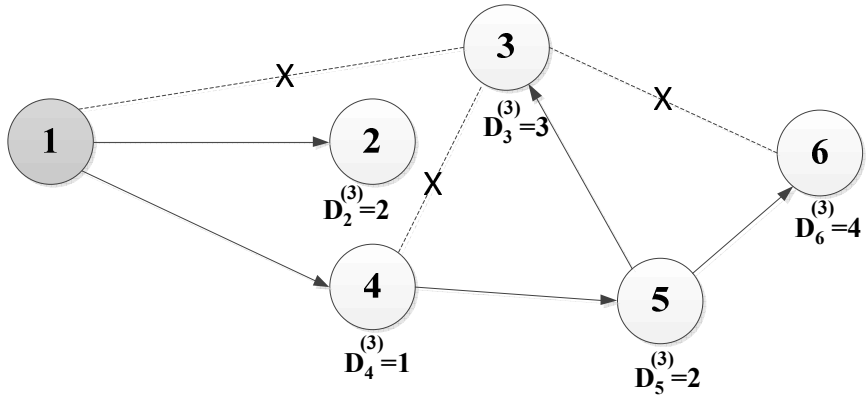


Figure 11.36 Troisième itération (Bellman-Ford)

La figure 11.36 illustre l'arborescence finale menant la source à tous les nœuds du réseau par le chemin le plus court en suivant l'algorithme de Bellman-Ford. Notons que le résultat est le même que celui obtenu en suivant l'approche Dijkstra.

La représentation des itérations de notre exemple est illustrée à la figure 11.37. Cette figure indique les itérations de même que les chemins et les coûts qu'ils représentent.

Itération h	D2	chemin	D3	chemin	D4	chemin	D5	chemin	D6	chemin
0	inf	-	inf	-	inf	-	inf	-	inf	-
1	2	1-2	5	1-3	1	1-4	inf	-	inf	-
2	2	1-2	4	1-4-3	1	1-4	2	1-4-5	10	1-3-6
3	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
4	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6

inf – infinité

Figure 11.37 Représentation des itérations

La quatrième itération est simplement exécutée pour ne s'assurer qu'aucun changement n'ait lieu par rapport à l'itération précédente. La méthode figurative (figure 11.36) n'a pas besoin d'une quatrième itération puisqu'il est clair que toutes les possibilités sont couvertes.

La représentation formelle de l'algorithme de Bellman-Ford est selon les étapes suivantes :

Soit

S représentant le nœud source,

h nombre de liens utilisés dans un chemin (aussi l'itération en cours).

d_{ij} le coût du lien du nœud i au nœud j. notons que lorsque $d_{ij} = 0$ ou l'infini, les nœuds i et j ne sont pas connectés.

$D_n^{(h)}$ est le coût du chemin le moins cher de la source au nœud n après *h* itérations.

$D_S^{(h)} = 0$ pour toute valeur de *h*

$D_n^{(0)} = \infty$, pour tout $n \neq S$

Pour chaque $h > 0$ successif

$D_n^{(h+1)} = \min [D_j^h + d_{jn}]$

(j étant le nœud lié directement au nœud n)

Ainsi le plus petit coût est choisi à chaque itération.

11.6 Multi distribution (*Multicast*)

Dans la foulée des développements technologiques des dernières années, les applications en télécommunication ont infiniment évolué. Les réseaux hauts débits, aussi bien ATM (*Asynchronous Transfer Mode*) que IP sont assez performants pour permettre le transfert rapide de fichiers volumineux, la téléphonie sur IP ainsi que le multimédia. Certaines applications ayant recours à ces réseaux requièrent une interaction entre une station source et plusieurs terminaux de destination. Il s'agit alors de « multicast » ou *multi distribution*, et implique la transmission de données d'une station source vers plusieurs terminaux hôtes de destination. Cette transmission ne survient qu'une seule fois par le biais de la station source. Par ailleurs, les données sont recopiées en chemin à des routeurs pour être acheminées vers différents terminaux de destination. Chaque routeur reçoit un paquet par un port d'entrée et ré acheminent une copie vers chaque port de sortie qui mène à un membre du groupe.

Le déploiement des services à valeur ajoutée, tel que la visioconférence, la télédistribution, ou encore la diffusion audiovisuelle constituent autant

d'incitatifs en faveur du développement massif de fonctions optimisant le traitement du trafic en *multi distribution* (multicast) au sein d'un réseau.

11.6.1 Besoin et contexte

La vidéoconférence, la conférence multimédia (qui comprend en plus de la vidéoconférence le transfert de données interactives), la duplication des données, le multicast des données en temps réel, les jeux sur réseaux et la simulation sont des exemples d'applications recourant au multicast. Ainsi, dans un cas de duplication des données tout comme la transmission des changements de prix fait par le bureau central d'une chaîne de magasins, la transmission ne se fait pas que pour un seul magasin à la fois, mais plutôt en mode multicast, soit une seule fois et simultanément pour tous les magasins. Le multicast de données en temps réel est aussi une application essentielle pour la diffusion des côtes à la bourse. Avec le multicast, les jeux sur réseaux et la simulation n'ont plus besoin d'établir N^2 connexions pour connecter N joueurs. Le débit s'en trouve ainsi considérablement réduit, tout en permettant une augmentation du nombre de joueurs sans pour autant recourir à une largeur de bande plus importante.

Le multicast pose toutefois des problèmes. Ainsi, le multicast sur réseau IP doit être amélioré puisque, tout comme nous le verrons, le service offert aujourd'hui n'est pas fiable et sa qualité laisse à désirer.

11.6.2 Le multicast sur réseau IP

Le multicat sur réseau IP fait l'objet de problèmes distincts. Le protocole TCP (*Transport Control Protocol*), fiable, ne peut pas être utilisé parce que ce dernier est propre à la communication bout en bout. Par conséquent, la livraison de « paquets » pour le multicast doit faire appel au protocole de transport UDP (*User Datagram Protocol*), lequel n'est pas doté d'un mécanisme de contrôle de la congestion et n'est pas pleinement fiable. Ainsi, l'utilisateur multicast IP doit s'attendre à éprouver des pertes occasionnelles de paquets.

Les paquets envoyés lors de la transmission multicast sont recopiés lorsqu'ils atteignent un point de rendez-vous. Il arrive qu'une même destination puisse être atteinte par plusieurs chemins, ce qui entraîne la réception de plusieurs copies du même paquet à ce point de rendez-vous. Comme le protocole UDP n'est pas doté du mécanisme de contrôle de congestion, lorsque cette dernière se manifeste sur une partie du réseau, le trafic multicast qui s'y trouve ne fera qu'amplifier le problème.

Le multicast dans le réseau IP considère des adresses ayant un préfixe distinct. Le champ d'adresses multicast étant limité, il n'est pas possible de commuter le trafic multicast selon l'adresse MAC (commutation couche 2). Il faut donc avoir recours à la commutation paquet par la couche 3 (c'est à dire la couche IP), ce qui a pour effet de ralentir la communication.

La mise sur pied d'un service multicast fiable et acceptable doit permettre la reconnaissance des groupes d'utilisateurs multicast. Ce service doit permettre et contrôler l'annexion et le départ d'utilisateurs. Afin d'optimiser la répartition du trafic, les protocoles du routage multicast doivent être sélectionnés selon les besoins dudit trafic. Il importe aussi de clarifier quels seront les protocoles d'application assignés à chacun des services multicast.

Plusieurs problèmes relatifs aux services multicast sont identifiés. Ainsi, lorsqu'un utilisateur veut se joindre ou quitter un groupe multicast, la source doit en être avisée. Cette information est des plus importantes tant pour les routeurs que les commutateurs qui desservent les ramifications des branchements du groupe. Ces éléments de réseaux pourront alors décider de savoir s'il est nécessaire de créer un nouveau raccordement pour permettre à un nouvel abonné de se joindre au groupe, ou s'il faut se débarrasser d'un autre parce qu'un membre du groupe s'est retiré et qu'il n'y a plus personne à desservir sur ce branchement.

D'autre part, lorsqu'il s'agit de connexion multimédia, les données sont de types distincts (voix et données) et les contraintes sont uniques. Ces contraintes de la voix par exemple sont celles du délai primordialement alors que dans le cas des data il est question du contrôle des erreurs. Si chacun de ces éléments est traité par le réseau selon l'approche la plus adéquate, ceux-ci risquent d'être acheminés par des voies divergentes vers la même station de destination. Ainsi, et pour chacun des terminaux de destination, il risque d'y avoir un branchement supplémentaire et une Qualité de Service (QdS) différente. Par contre, si le même raccordement sert pour tous les éléments du service multimédia, cela permettrait alors de contrôler d'une part la connexion dans son ensemble et de garantir, de l'autre, le séquencement, par contre les ressources du réseau seront mises à rude épreuve.

Le service multicast doit pouvoir garantir la livraison simultanée de l'information à tous les membres du groupe. Parmi les membres du même groupe, on doit éviter une différence de temps de réception de l'information. Aussi, il faut nous assurer d'une certaine synchronisation dans la diffusion de l'information. La synchronisation doit aussi se faire, dans le cas du multimédia, entre tous les éléments du service.

Étant donné que la répartition du trafic se fait vers plusieurs destinations et que la QoS est très importante, il faut pouvoir s'assurer que les données ont bien été reçues par tous et dès lors en aviser la station source. Si un terminal de destination a éprouvé des problèmes de réception, il faut savoir comment ces erreurs seront traitées. Va-t-il y avoir retransmission? Est-ce que l'aspect inéquitable de la qualité de service est acceptable? Nous nous trouvons face à une question de fiabilité du service multicast.

Parmi les autres problèmes importants, citons : la taille du groupe, le contrôle de flux du trafic et la rectification des erreurs. Le protocole de transport TCP ne peut être exploité tout comme dans le cas de la communication unicast. Ceci est principalement dû aux multiples échanges que ce protocole exige. Il ne peut pas y avoir de tels échanges entre une station source et plusieurs terminaux de destination simultanément sans créer de confusion. Par contre, sans ces échanges, il ne nous est pas donné de connaître la position des terminaux de destination. Il incombe aux routeurs de s'informer entre eux en regard de la position de chaque membre qu'ils desservent. Toutefois, cette gestion devient de plus en plus laborieuse au fur et à mesure que le groupe grandit.

La sécurité des interconnexions n'est pas moins importante en mode multicast. En effet, comment protéger le groupe de l'accès d'indésirables? Le multicast fait appel au protocole de transport UDP. Ce dernier est bloqué par le « Firewall » (coupe-feu). Il s'agit alors de permettre à ces coupe-feu de reconnaître le trafic multicast tout en assurant une sécurité d'accès. C'est un point qui reste à développer.

Les principaux défis techniques du routage multicast sont les suivants[4] :

- Minimiser la surcharge du réseau en évitant les boucles et la concentration du trafic sur un lien ou sur un sous-réseau.
- Offrir un support de base pour une transmission fiable. La perte de liens ne doit pas accroître le délai de transmission ou réduire les ressources disponibles.
- L'algorithme du routage doit pouvoir créer des routes optimales en prenant en considération différents paramètres, incluant les fonctions inhérentes aux coûts, les ressources disponibles, la largeur de bande, le nombre de liens, le délai bout à bout, etc.

Tous les services pour des terminaux aux destinations multiples ont besoin d'un arbre de distribution plutôt que de chemins à travers le réseau. L'algorithme idéal

[4] R. Braudes and S. Zabele, « *Requirements for multicast protocols* », *RFC 1458*, May 1993.

devrait pouvoir offrir un arbre qui se ramifierait aux membres du groupe exclusivement et en outre, il pourrait :

- Évoluer avec l'appartenance du groupe.
- Optimiser la route en prenant en considération la fonction coûts.
- Éviter la concentration du trafic sur les liens ou sur les nœuds.

Il existe quelques algorithmes de base pour établir l'arborescence multicast, dont : le routage basé sur la source (*Source based routing*), l'arborescence Steiner (*Steiner trees*) et l'arborescence basée sur un centre (*Center based trees*). Toutefois, tout comme C. Diot et al[5] l'ont identifié, un certain nombre de problèmes sérieux prévalent et doivent être résolus afin de rendre ce service plus viable.
Le protocole multicast IP est fondé sur le récepteur. Il fait partie de la catégorie de protocoles connus sous le nom de *Receiver-based Multicast Protocol*. Les terminaux hôtes qui font partie d'un groupe partagent la même adresse multicast. Ils reçoivent les paquets adressés à cette adresse multicast et transmise par un terminal qui n'est pas nécessairement membre du groupe. Ce terminal n'a pas connaissance des membres de ce groupe et il fonctionne tout comme s'il s'agissait d'une transmission Unicast. L'importance réside dans le déploiement des paquets vers les terminaux de destination.

11.6.3 Adresse de groupe

La session multicast a besoin d'opter pour une adresse de groupe. Comme nous l'avons vu, les adresses de groupe IPv4 font partie de la classe D et elles couvrent les adresses 224.0.0.0 à 239.255.255.255. Les adresses 224.0.0.0 à 224.0.0.255 sont réservées et ne sont pas acheminées en dehors du sous-réseau. Les adresses 239.0.0.0 à 239.255.255.255 sont réservées aux réseaux privés. Ces dernières adresses peuvent être exploitées dans un domaine multicast privé sans crainte de recevoir une communication de l'externe avec cette adresse.

L'adresse MAC Ethernet doit aussi prévoir des adresses multicast. Le bit 0 du premier octet de l'adresse MAC nous signale si l'adresse de la station de destination est en mode de diffusion/multicast ou unicast. Lorsque ce bit se trouve à 1, indiquant qu'il est en mode multicast, la couche 2 peut alors transmettre des datagrammes IP multicast à un groupe d'hôtes dans un segment.

Un seul OUI est réservé aux adresses multicast Ethernet et de ce OUI, seule la moitié est utilisée pour les adresses multicast. L'adresse de ce OUI est : 01:00:5e. Les adresses MAC multicast sont données conformément à leur format binaire, soit :

[5] C. Diot et al « *Deployment Issues for the IP Multicast Service and Architecture* », IEEE Network, Jan/Feb 2000

00000001:00000000:01011110:0xxxxxxx:xxxxxxxx:xxxxxxxx.

Ainsi, sous MAC, il n'y a que 23 bits disponibles pour préciser les groupes d'adresses multicast. Dans le cas des adresses IP, 28 bits sont disponibles pour désigner les adresses multicast étant donné que les bits les plus significatifs doivent être 1110. Il faut donc faire correspondre 28 bits IP d'adresses multicast à 23 bits MAC d'adresses multicast (figure 11.38).

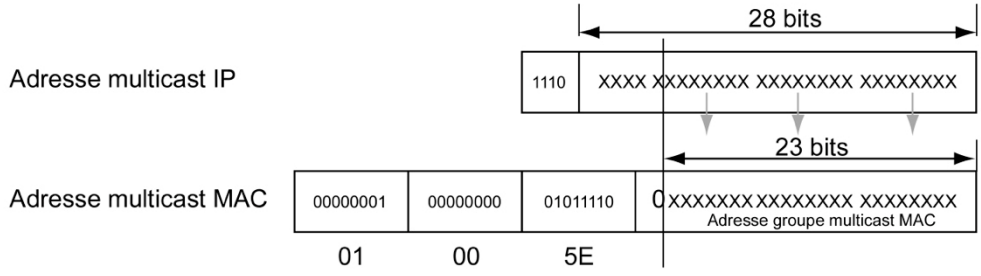


Figure 11.38 Adresses multicast

Nous remarquons que, pour chacune des adresses multicast MAC, il peut y avoir 32 adresses multicast IP correspondantes puisque seuls 23 bits d'adresse MAC correspondent à 28 bits d'adresse IP. En d'autres termes, 32 différentes adresses multicast IP correspondent à une seule adresse multicast MAC. Pour reconnaître une adresse multicast, un routeur se doit de vérifier l'adresse à la couche 3.

11.6.4 Distribution multicast

La distribution du trafic multicast peut se faire soit par le biais d'une arborescence émanant de la source (*source trees*), soit à partir d'une arborescence partagée (*shared trees*).

Dans le cas d'une arborescence émanant de la source (*source trees*) telle qu'illustrée à la figure 11.39, la racine de l'arbre représente la source même du trafic multicast. L'arborescence qui est établie en direction des terminaux de destination membres du groupe multicast repose sur le Spanning tree qui a été traité au chapitre 7. Cette arborescence est également connue sous le sigle SPT (*Shortest path tree*). La notation utilisée est (S,G), où S signifie l'adresse de la source et G l'adresse du groupe.

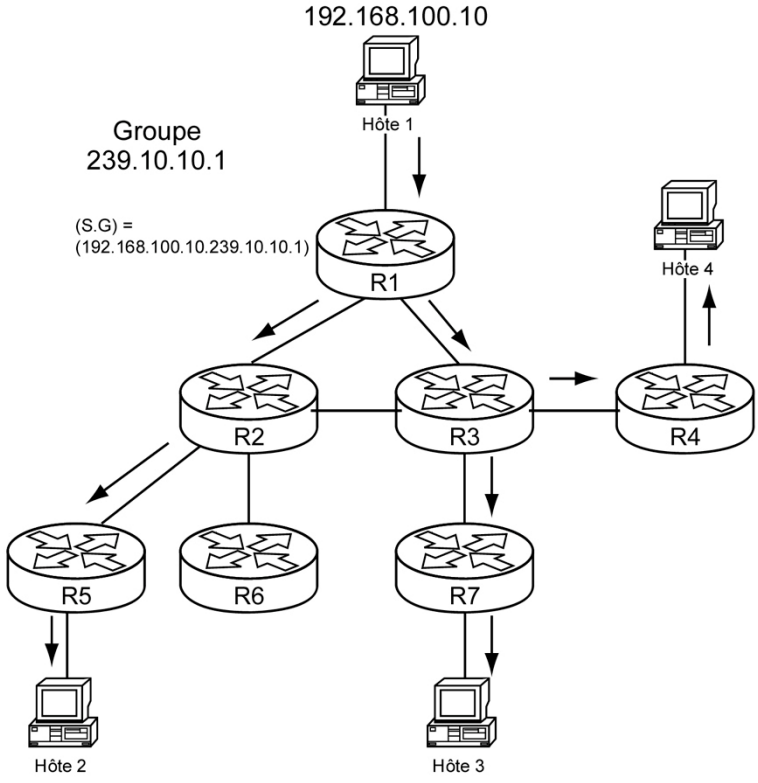


Figure 11.39 Arborescence émanant de la source (*source trees*)

L'arborescence partagée (*Shared Tree*) illustrée à la figure 11.40 comporte un routeur commun visant à atteindre toutes les destinations d'un groupe donné. Chacun des hôtes (sources) qui désirent transmettre envoie ses paquets à son routeur qui les relaie à ce routeur commun connu sous le nom de point de rendez-vous RP (*rendez-vous point*). Elle est aussi connue sous le nom de RPT (*RP tree*). Il peut exister conjointement plusieurs RP dans un réseau.

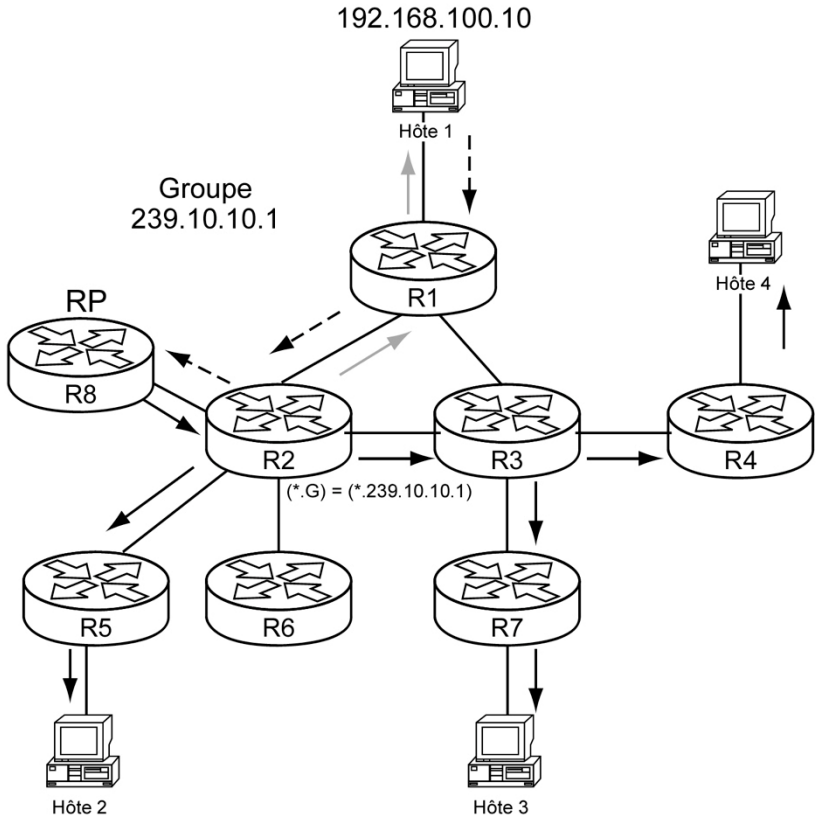


Figure 11.40 L'arborescence partagée (*Shared Tree*)

Le RP évalue le chemin le plus court vers tous les membres du groupe. Chacun des hôtes sources transmet l'information au RP en favorisant le chemin le plus court pour l'atteindre. Cette méthode est employée par le protocole du routage PIM (*Protocol Independent Multicast*). La notation retenue est (*,G). L'astérisque indique que toutes les sources empruntent le même chemin partagé.

Chaque routeur doit connaître la provenance des paquets ainsi que le chemin menant à la station source. Lorsqu'un routeur reçoit un paquet en mode multicast, il vérifie, pour l'acheminer, si le port d'où ce paquet est parvenu correspond bien au chemin à prendre pour atteindre la station source. Si tel est le cas, le paquet est accepté, autrement, il est tout simplement détruit. Ce mécanisme est connu sous l'acronyme RPF (*Reverse Path Forward*).

Pour limiter l'étendue du domaine multicast, le champ TTL (*time to live*) figurant dans l'en-tête du datagramme IP peut être fort utile. En effet, comme nous l'avons déjà vu, ce champ comporte une valeur initiale signalée par la source du paquet. Après chaque visite du paquet, le routeur décrémente le champ de 1.

Lorsque ce dernier atteint 0, le paquet est détruit. Toutefois, le routeur pourrait fixer le seuil de la durée de vie (TTL) qui serait acceptable à une valeur autre que 0. Si ce seuil est établi à 32, tout paquet reçu dont le TTL est égal à 32 ou moins, sera détruit. Ainsi, des régions pourront être spécifiées en vue de limiter la propagation du trafic multicast. À titre d'exemple, les routeurs de bordure d'une région du réseau pourraient fixer le seuil d'un TTL à 128. Les sources de trafic généreraient alors des paquets avec un TTL moindre que 128. Ces paquets seront donc astreints à cette région et ne pourront jamais aller au-delà des routeurs de bordure.

11.6.5 Protocoles du routage multicast

Les protocoles du routage sont basés conformément à l'une des trois catégories suivantes :

- Protocoles à mode dense DM (*Dense Mode protocols*)
- Protocoles à mode clairsemé SM (*Sparse Mode protocols*)
- Protocoles d'état de liens (*Link state protocols*)

Parmi les protocoles à mode dense il y a le protocole DVMRP (*Distance Vector Multicast Routing Protocol*) et le protocole PIM DM. Ces protocoles ont recours à l'approche en arborescence émanant de la source ou plus précisément du SPT. Le principe de ces protocoles consiste à inonder de paquets tous les membres du groupe. Les routeurs qui détectent l'absence de connexions de membres du groupe renvoient vers la source un message d'élagage (*prune*) afin de limiter le trafic multicast. Les messages d'élagage ont une certaine durée de vie ce après quoi le trafic multicast est rediffusé à tous les membres du groupe. Les routeurs qui désirent maintenir le même état doivent transmettre périodiquement des messages d'élagage.

Ces protocoles peuvent aussi greffer un embranchement multicast qui a été élagué. Ce processus est plus rapide que de devoir attendre l'expiration des temporisateurs d'élagage. Les messages de greffe sont transmis à cet effet.

Les protocoles à mode clairsemé ont recours quant à eux à l'approche de l'arborescence partagée. Le protocole PIM SM en est un bon exemple. Contrairement au mode dense, ce protocole ne transmet du trafic multicast qu'aux routeurs qui se joignent à la session. Les routeurs ayant des hôtes désireux de prendre part à la distribution multicast, doivent transmettre des messages faisant part de leur souhait de se joindre (*join*), en direction du point de rendez-vous. Pour éliminer la connexion d'un routeur qui s'est joint à la distribution

multicast, un message d'élagage est transmis au routeur suivant se trouvant en direction du RP.

Les protocoles à état de liens font appel à l'approche d'une arborescence émanant de la source (SPT) pour répartir le trafic multicast. Le protocole MOSPF en est un exemple. Par contre, et contrairement aux protocoles à mode dense, le MOSPF ne fait nullement appel au mécanisme d'inondation ou à celui d'élagage que l'on retrouve dans les protocoles DVMRP et PIM DM. Ce sont les liens, identifiés dans les informations échangées en regard de l'état des liens, qui sont inondés par le trafic en mode multicast.

11.6.6 Le protocole IGMP

Le protocole IGMP (*Internet Group Management Protocol*) permet aux applications des hôtes de s'associer à un groupe en mode multicast. Ce protocole est détaillé dans le RFC 1112 et mis à jour dans le RFC 2236. C'est lui qui permet aux hôtes d'aviser leur routeur multicast local lorsqu'ils désirent se joindre à un groupe ou se retirer d'un groupe.

Avec le concours de ce protocole, chaque hôte, doté d'une application en mode multicast, transmet des messages d'annonces au routeur l'avisant des sessions multicast auxquelles il veut se joindre ou desquelles il désire se retirer. De la sorte, le routeur peut mettre à jour les tables de groupe dans lesquelles tous les hôtes participants figurent.

Il existe deux versions du protocole IGMP. La version 2 illustrée dans la figure 11.34 est une amélioration de la version 1. Nous nous limiterons dans ce contexte à la description de la version 2.

Le paquet IGMP est transporté dans un datagramme IP (figure 11.41). Le champ « protocole » de l'en-tête IP mentionne que la valeur est 2. Cela signifie que les 8 octets qui suivent l'en-tête forment un paquet IGMP.

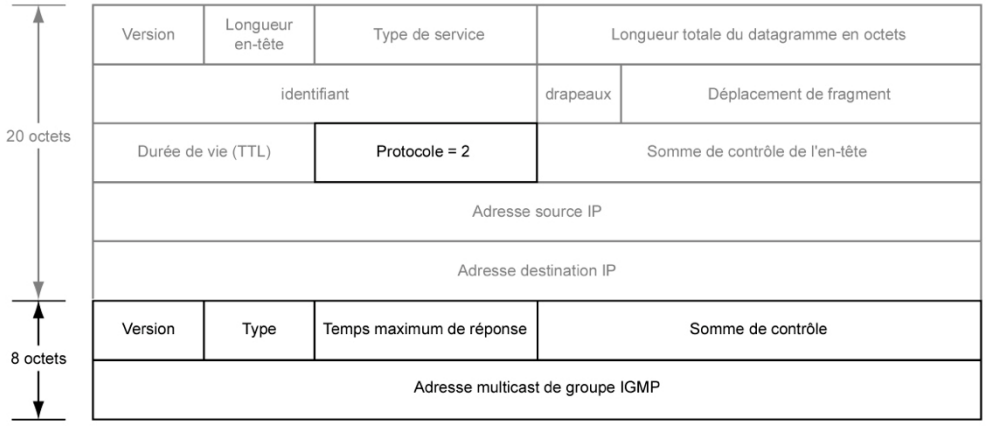


Figure 11.41 Format du paquet IGMPv2

Le champ version du paquet IGMP est codé à 2 pour signaler qu'il s'agit de la version 2. Le champ suivant, Type fait part des différents genres de message. En règle générale, il s'agit de messages de demande ou de report. Les catégories de message du paquet IGMPv2 sont :

- Demande d'adhésion (*Membership Query*) (0x11). Si la demande s'adresse à tous les groupes alors l'adresse multicast est : 0.0.0.0. Si la demande est faite pour un groupe en particulier, l'adresse du groupe est mentionnée. La durée de vie ou TTL est alors mise à 1.
- Rapport d'adhésion IGMPv1 (*Membership report*) (0x12)
- Rapport d'adhésion IGMPv2 (0x16)
- Retrait du groupe (*leave group*) (0x17) envoyé à l'adresse 224.0.0.2 (soit à tous les routeurs)

Le champ suivant « Temps maximum de réponse » indique le temps d'attente maximum du terminal hôte avant l'envoi d'un report en réponse à une demande. Ce champ contient des 1/10 de secondes et sa valeur par défaut peut atteindre 100, c'est-à-dire 10 secondes.
Le champ « somme de contrôle » vérifie les 8 octets du paquet IGMP en mettant son propre champ à 0.

11.6.6.1 Mécanisme de fonctionnement IGMP

Nous allons décrire le mécanisme qui permet aux hôtes de s'inscrire à une session multicast particulière. Considérons un sous-réseau composé de plusieurs routeurs. Un seul des routeurs est sélectionné pour traiter les demandes d'adhésion. Ce routeur est connu sous le nom de IGMP *Querier*. Périodiquement,

ce routeur transmet à tous les hôtes du sous-réseau le message « Demande d'adhésion » (*Membership Query*) avec l'adresse 224.0.0.1.

Si un hôte du sous-réseau désire adhérer, il adresse le message « Rapport d'adhésion IGMPv2 » en y indiquant l'adresse du groupe. Si un autre terminal hôte du même sous-réseau désire également adhérer au même groupe, il n'a pas besoin de transmettre de message puisque celui du terminal hôte précédent suffit et de ce fait permet au routeur de diffuser l'information de la session multicast demandée sur le sous-réseau. Par contre, si un terminal hôte désirait s'inscrire à un groupe différent, il se devrait de transmettre son message de « Rapport d'adhésion IGMPv2 » avec l'adresse de son groupe.

Grâce à la version 2 de IGMP, le mécanisme de retrait du groupe s'en trouve accéléré. Un hôte d'un sous-réseau désireux de se retirer de son groupe, transmet à l'adresse IP 224.0.0.2 le message de « Retrait du groupe » (*leave group*) à tous les routeurs en y précisant l'adresse du groupe.

Le routeur IGMP Querier ne sachant pas s'il reste d'autres hôtes appartenant à ce groupe et relevant de ce sous-réseau, lui adresse aussitôt le message de « Demande d'adhésion » (*Membership Query*) précis, c'est-à-dire en y mentionnant l'adresse du groupe, et non pas général comme il en a été question lors de l'adhésion des stations.

Si un autre hôte du sous-réseau participe toujours à la session multicast de ce groupe, il transmet le message de « Rapport d'adhésion IGMPv2 » avec l'adresse du groupe. Comme nous l'avons vu précédemment, un autre hôte faisant aussi parti du groupe n'a pas besoin de transmettre le message d'adhésion. Le routeur va donc continuer à transmettre au sous-réseau les échanges du groupe multicast.

Par contre, si aucun hôte ne reste dans le sous-réseau appartenant au groupe multicast après le départ de la dernière station, le routeur IGMP Querier ne recevra pas de réponse à son message précis de « Demande d'adhésion » (*Membership Query*). Le routeur n'attend qu'une seconde après ce message. Il retransmet ce message et s'il ne reçoit aucune réponse, il décide alors de retirer le sous-réseau de la liste de distribution de messages multicast du groupe en question.

11.7 Résumé

La robustesse d'une part et le choix du chemin le plus judicieux sont principalement les deux éléments que nous recherchons dans un protocole du routage. Nous nous sommes penchés sur les deux méthodes fondamentales

visant à acheminer des paquets dans Internet, notamment : Les protocoles basés sur le vecteur de distance (*Distant Vector*), et ceux basés sur l'état de lien (*Link State*). Ils ont été respectivement représentés par les protocoles de routage RIP et OSPF.

Le choix du coût des liens n'est pas chose facile d'autant plus qu'il a une influence capitale sur la décision relative au chemin le plus court, soit le moins onéreux. Une fois le chemin choisi, les liens qui le composent doivent pouvoir tolérer la charge du trafic. Nous voyons donc que le choix du coût a un impact direct sur la congestion du trafic.

Le coût dépend de plusieurs paramètres autres que la largeur de bande. On pourrait prendre en considération la sécurité du point d'accès, le coût monétaire du lien voire même le moment de la journée pour procéder à la transmission.

Nous savons que le protocole OSPF utilise le concept d'aire pour l'appliquer à des réseaux étendus.

Nous n'avons couvert que les éléments fondamentaux des protocoles de routage. Par contre, dans la littérature, il existe plusieurs contributions qui définissent un grand nombre d'algorithmes. Entre autres, citons en exemple celui proposé par Fratta, Gerla et Kleinrock[6] et dont l'objectif vise à minimiser le délai moyen dans l'algorithme de routage.

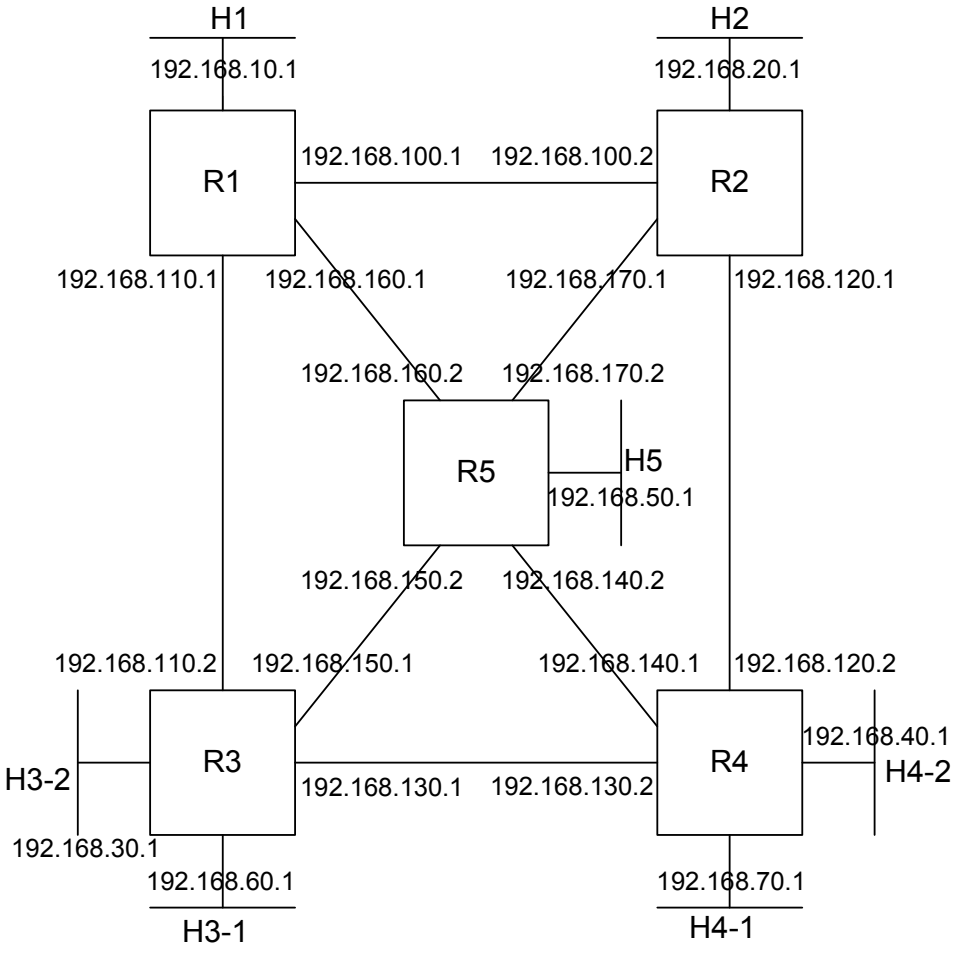
Nous avons fait un survol des concepts de la multi distribution ou multicast. Une meilleure compréhension des méthodes de distribution est essentielle pour être plus en mesure de saisir les assises des protocoles utilisés en mode multicast.

[6] Fratta, L., M. Gerla et L. Kleinrock, « *The Flow Deviation Method: An Approach to Store-and-Forward Communication Network Design* », *Networks*, vol. 3, pp. 97-133, 1973.

11.8 Exercices

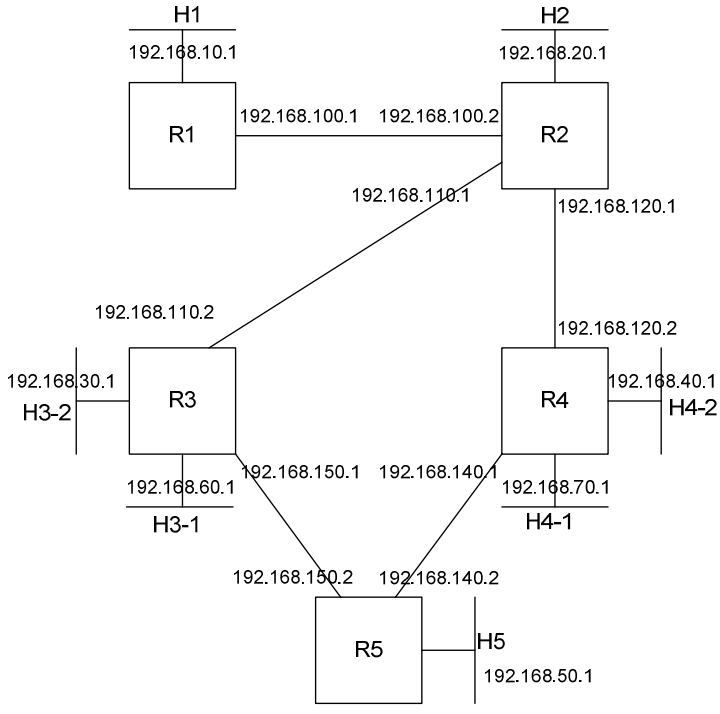
11.1

Le réseau suivant fait appel au protocole de routage RIP. Avec les données telles qu'illustrées, complétez les tables de routage de chacun des routeurs.



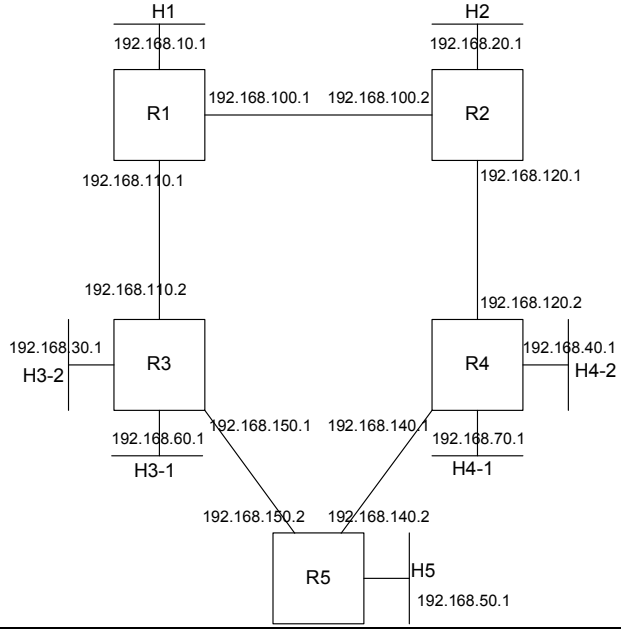
11.2

a) Établissez les tables de routage RIP pour le réseau suivant.
b) Si le réseau H1 venait à tomber en panne y aurait-il éventuellement un problème de convergence? Décrivez lequel.
c) Expliquez quelle solution permettra de résoudre ce problème.



11.3

Pour le réseau suivant, établissez les tables de routage RIPv2 dans chaque routeur. Notez que le masque réseau est basé sur la classe de l'adresse.



11.4

Expliquez pourquoi la contiguïté (*adjacency*) est aussi importante dans le cas du protocole OSPF. Expliquez à quoi elle mène.

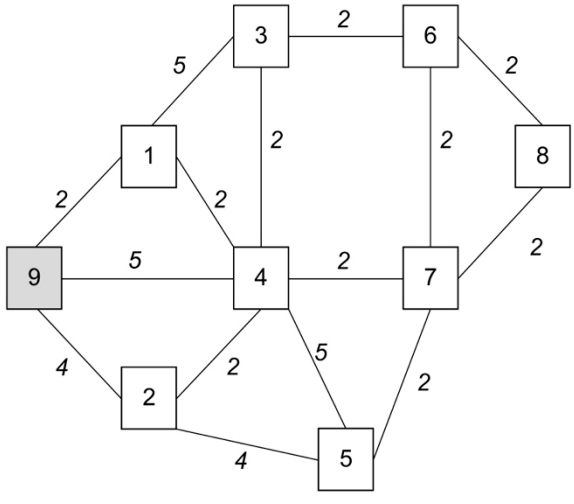
11.5

Expliquez la différence entre la base de données LSDB des liens OSPF et la table de routage élaborée par le protocole OSPF. Quel lien prévaut entre elles?

11.6

Soit le réseau suivant. Si l'on fait appel au protocole OSPF,

a) calculez la base de données LSDB de chacun des nœuds après établissement de la contiguïté entre tous les nœuds. Les coûts de chaque lien figurent en italique.
b) Décrivez le fonctionnement d'OSPF dans ce réseau et le processus de convergence de la table de routage notamment la mémoire *ROUTAGE*. Indiquer le contenu des mémoires *ROUTAGE* et *ITÉRATION* à chaque itération.



11.7

Pour le réseau précédent, trouvez les chemins les plus courts à partir du nœud 9 en recourant à l'algorithme de Dijkstra.

11.8

Pour le réseau précédent, trouvez les chemins les plus courts à partir du nœud 9 en recourant à l'algorithme de Bellman-Ford.

11.9

Indiquez l'adresse MAC multicast correspondant aux adresses IP multicast suivantes :

a) 224.156.34.58
b) 224.223.49.157
c) 239.245.44.32
d) 239.0.4.2

11.10

Indiquez toutes les adresses IP multicast qu'il est possible d'obtenir pour chacune des adresses MAC multicast suivantes :

a) 01:00:5e:5b:a2:35
b) 01:00:5e:73:67:41
c) 01:00:5e:7f:bd:89
d) 01:00:5e:00:11:16

11.11

Si un terminal hôte du sous-réseau transmet le message « *Rapport d'adhésion IGMPv2* » avec une adresse de groupe, un autre terminal hôte faisant aussi partie du même groupe n'a pas besoin de transmettre de message d'adhésion. Expliquez en quoi cette situation est souhaitable.

12 La couche transport

12.1 Introduction

Tout comme nous l'avons vu dans le modèle de référence OSI et TCP/IP, la couche Transport fait appel au protocole de communication bout en bout. Cela signifie que la communication ou la pertinence de l'échange d'informations ne se fait qu'entre source et destination. Pour sa part la couche Application fait également appel à des protocoles de communication bout en bout.

La couche Transport du modèle TCP/IP fait état de deux protocoles distincts : le TCP (*Transmission Control Protocol*) et le UDP (*User Datagram Protocol*).

La couche Transport joue un rôle prépondérant dans la série de protocoles Internet. En effet, c'est une fois parvenu à ce niveau que les terminaux établissent la communication entre la source et la destination. Par contre, en ce qui a trait aux réseaux de télécommunication, elle demeure transparente aux activités du réseau puisque ce dernier n'a pas connaissance de la couche transport.

Les programmes d'application comprennent des processus qui génèrent les données, véhiculées par la couche Transport et par le réseau. Ces données sont destinées à la couche application qui se trouve dans le terminal de destination. Le processus ne peut pas toujours être identifié par celui de la source. La raison serait causée par un changement advenu au processus de destination. Comment un processus destination pourrait être alors adressé par un processus source si ce dernier n'est pas avisé du changement? Il faut donc que la destination des données ne soit pas identifiée par le processus lui-même étant donné qu'il pourrait changer, mais bien plutôt par les fonctions qu'il offre.

La solution à ces problèmes d'accès aux processus appropriés repose sur le recours aux ports d'accès liés à chaque processus tant à la source qu'à la destination. Le port d'accès assume le rôle d'interface entre la couche Application et la couche Transport. Il est identifiable par un entier positif.

Ainsi, pour avoir accès au processus de destination il suffit d'en connaître son port d'accès et de l'adresser. Le processus peut changer à cause d'une mise à jour ou un nouveau développement et le nouveau processus peut obtenir ses données par le même port pour autant que ses fonctions soient compatibles à celles adressées par la source.

En ce qui a trait aux protocoles de transport TCP et UDP, les numéros de ports sont assignés séparément. Pour des raisons d'ordre pratique et lorsque cela s'avère possible, les mêmes numéros sont assignés aux ports UDP et TCP ayant accès aux mêmes types de fonctions. Toutefois, ces ports sont différents même s'ils conservent le même numéro.

Les numéros des ports d'accès sont définis par l'IANA (*Internet Assigned Number Authority*) et leur attribution est faite par elle. Ces numéros sont répartis en trois catégories :

- Ports bien connus (*Well Known Ports*)
- Ports enregistrés (*Registered Ports*)
- Ports dynamiques /privés (*Dynamic and/or Private Ports*)

Les ports bien connus sont numérotés de 0 à 1023. Ils sont généralement utilisés par des processus de système ou par des programmes exécutés par des « utilisateurs privilégiés » (*priviledged users*) tel qu'indiqué dans le RFC.

Nous donnons ci-dessous des exemples de ce type de ports :

20 FTP (*File Transfer Protocol*),
23 Telnet,
69 TFTP (*Trivial File Transfer Protocol*),
80 WWW (*World Wide Web HTTP*),
110 POP3 (*Post Office Protocol – version 3*),
161 SNMP (*Simple Network Management Protocol*),
520 RIP (*Routing Information Protocol*).

Les ports bien connus pour les protocoles TCP et UDP sont énoncés et consignés dans les RFC 793, et RFC 768 respectivement.

Les ports enregistrés sont numérotés de 1024 à 49151. Ces nombres sont utilisés par des processus ou programmes communs.

Les ports enregistrés du TCP, tels qu'ils figurent dans la RFC 793, permettent de procéder à une demande de connexion. Durant l'établissement de l'appel, la source peut préciser son propre port d'accès. Dans la mesure du possible, les

ports assignés à un protocole UDP (RFC 768) devraient être les mêmes que ceux assignés à un protocole TCP.

Les ports dynamiques ou privés vont de 49152 à 65535 et sont disponibles à tous processus privés.

Chaque fois qu'un hôte communique avec un autre, il a besoin de l'adresse IP et du port d'accès du processus de destination.

Si les ports sont connus, la communication se fait directement et aucun échange de numéros de port n'est, au préalable, nécessaire. Si le port destination est enregistré, le processus de destination n'a pas besoin de le faire initialement connaître. Le processus source peut faire connaître son port au moment de l'établissement de l'appel s'il s'agit d'une connexion TCP. Si le port destination est dynamique ou privé il faut que le processus source en prenne connaissance. Puisque les processus qui communiquent sont privés, ils doivent avoir en leur possession le numéro de port. Il peut en faire la demande à un serveur. Pour avoir accès à un serveur il utilise un port connu. Le processus « *Port Mapper* » dont le port est connu, peut aider à trouver le port du processus désiré.

12.2 UDP (*User Datagram Protocol*)

UDP (*User Datagram Protocol*) tel que consigné dans la RFC 768, est un protocole transport sans connexion. Le processus ou programme d'application qui génère des données, constitue des messages UDP, transmis l'un après l'autre. Chacun de ces messages est encapsulé dans un datagramme IP.

Le protocole UDP est très simple. Il ne comporte aucune technique de contrôle de flux pour ralentir le transmetteur. Par contre, dans son entête, il existe un champ de détection d'erreurs : le champ *checksum* dont l'utilisation est optionnelle.

Le format du message UDP, tel qu'illustré à la figure 12.1, comprend un en-tête de 64 bits soit une longueur de 8 octets et les données. L'en-tête comporte : les numéros de port UDP source et destination, la longueur du message et le champ checksum.

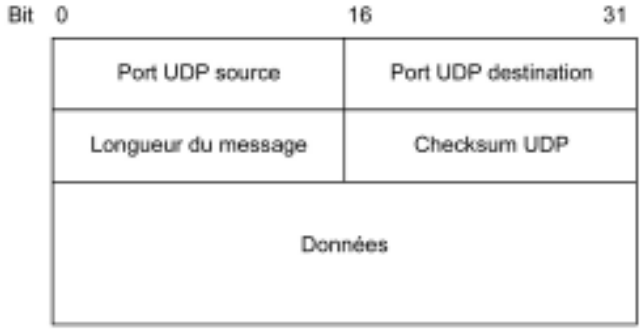


Figure 12.1 Message UDP

Le checksum de l'en-tête UDP est optionnel. S'il n'est pas utilisé sa valeur est mise à 0.

Le calcul du checksum, lorsqu'il est utilisé, se fait sur un pseudo en-tête en plus du message complet UDP. Le champ checksum du message UDP est à 0 lorsqu'il y a calcul. Le résultat du calcul est alors placé dans ce même champ Checksum UDP du message UDP avant sa transmission.

Le pseudo en-tête illustré à la figure 12.2 comporte des adresses IP tant source que destination, le type de protocole figurant dans l'en-tête IP dont la valeur est 17 (UDP), et la longueur du message UDP excluant le pseudo en-tête. Le pseudo en-tête est long de 12 octets.

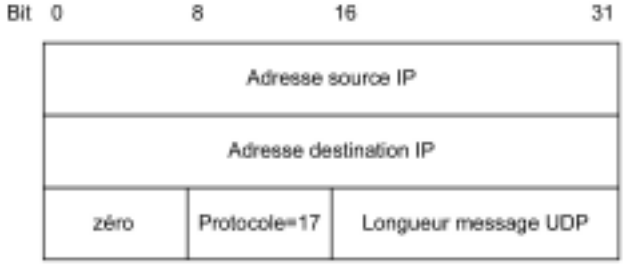


Figure 12.2 Le Pseudo en-tête

La raison pour laquelle on a recourt au pseudo en-tête consiste non seulement à procéder à la vérification d'erreurs mais également à valider la nature et la destination du message.

Le transmetteur du message calcule le checksum conformément à la méthode décrite. Le pseudo en-tête n'est pas transmis; il ne sert qu'au calcul du checksum.

Le récepteur du message UDP vérifie si le checksum est utilisé. Si tel est le cas (checksum différent de 0), le récepteur fait alors appel aux champs adresses IP, au protocole du datagramme IP de même qu'à la longueur du message pour reconstituer le pseudo en-tête. De la sorte, le récepteur du message peut calculer le checksum à partir du pseudo en-tête et du message UDP reçu. Il importe de souligner que, durant le calcul du checksum, le champ checksum du message reçu est remplacé par 0.

S'il n'y a pas d'erreurs, le récepteur sait alors que le contenu du message UDP est le bon.

Il importe de noter que, quoique le checksum du datagramme IP soit calculé de la même façon que le checksum UDP, il n'en demeure pas moins qu'une différence fondamentale prévaut entre eux. Dans le cas du datagramme IP, le checksum ne s'applique qu'à l'en-tête. Par contre, le checksum du message UDP procède à la vérification de l'en-tête, des données et s'assure que le message est le bon.

En théorie, la longueur d'un datagramme IP est de 65535 octets. Ceci est dû au 16 bits du champ « longueur totale du datagramme ». Si l'en-tête IP a une longueur de 20 octets et l'en-tête UDP 8 octets, il reste encore au datagramme une longueur de 65507 octets de données dans un message UDP.

La longueur des messages UDP est pratiquement de 8192 octets, 4160 octets voire même 512 octets tout dépendant des réseaux par lesquels ils sont appelés à passer.

En résumé la couche Transport UDP a les quatre caractéristiques suivantes :

- UDP est sans connexion.
- UDP ne traite que d'un datagramme IP à la fois.
- UDP est simple.
- UDP n'est pas fiable.

12.3 TCP (*Transmission Control Protocol*)

Le TCP (*Transmission Control Protocol*) est, conformément aux RFC 793 et RFC 1122, un protocole de la couche Transport offrant un transfert fiable des données de bout en bout. Ce protocole (TCP) est orienté connexion et les services qu'il offre sont les suivants :

- Le multiplexage
- La gestion de la connexion
- Le transport des données
- Le report d'erreurs

Les connexions TCP se font entre les points d'accès. Chaque point d'accès est identifié par une adresse IP et un port d'accès. Pour sa part, la connexion est identifiée par les ports d'accès tant source que destination. Ainsi, deux ports d'accès sont nécessaires pour établir une connexion.

Des connexions différentes peuvent avoir en commun le même point d'accès.

Ce dernier est également connu sous les noms d'interface de connexion (*Socket*), API (*Application Programming Interface*), de TLI (*Transport Layer Interface*) ou encore d'interface de connexion Berkeley pour indiquer l'endroit où il a été conçu et mis au point. L'identification d'une interface de connexion unique se fait en indiquant un numéro de port et une adresse IP.

Le service par TCP est donc établi par le biais d'une connexion logique entre une paire d'interfaces de connexion. Une connexion se compose des éléments suivants :

- Le type de protocole (TCP)
- L'adresse IP source
- Le numéro de port source
- L'adresse IP destination
- Le numéro de port destination.

Pour un terminal source tout comme pour un terminal destination, le système d'exploitation crée une interface de connexion locale avec : la valeur de son adresse IP, celles de son port d'accès, et celles du type de protocole.

L'établissement d'appel n'étant pas symétrique, l'approche client-serveur est privilégiée. La source est le client et la destination le serveur.

Prenons l'exemple de la figure 12.3 ci-dessous.
Un processus d'application X de l'hôte A désire communiquer avec un processus de l'hôte C.
Le processus d'application X se voit assigner le port 2. Il veut établir une connexion avec le port 1 de l'hôte C. Afin d'établir l'appel, il a recours aux interfaces de connexion.

La couche TCP de l'hôte A transmet alors une demande de connexion à la couche TCP de l'hôte C en lui mentionnant le port d'accès respectif (cette information est transférée à la couche réseau par la primitive IP SEND qui assigne A comme adresse source et C comme adresse destination).

Une fois la connexion établie, les données du processus d'application X rassemblées dans des segments du TCP sont transmises au port 1 de l'hôte C. Les données des segments reçues, adressées au port 2 de l'hôte A et venant du port 1 de l'hôte C sont seulement alors livrées au processus d'application X. Au même moment, le processus d'application Y de l'hôte A relié au port 1 peut échanger des données avec le port 3 de l'hôte B par une autre connexion TCP.

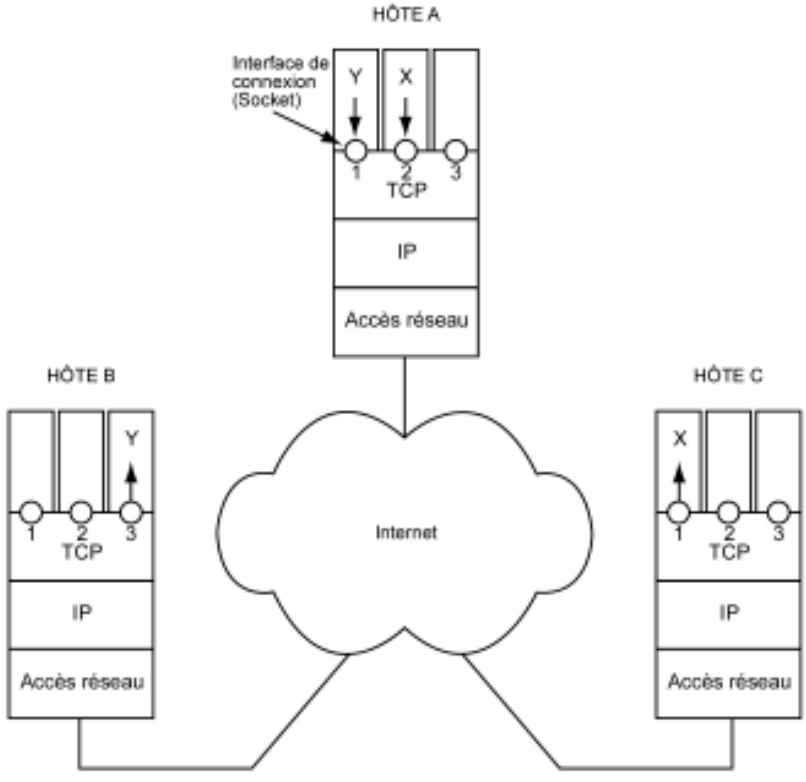


Figure 12.3 Communication utilisant les services du TCP

12.3.1 Services du protocole TCP

Le protocole TCP offre une variété de services. Étant donné que le TCP est orienté connexion, la bonne gestion de cette dernière est indispensable. Cette

gestion comporte les services suivants : établissement, maintien et libération d'un appel.

Le service d'établissement de l'appel permet à deux utilisateurs du TCP d'établir une connexion logique entre l'une de leurs interfaces de connexion respectives dans la mesure où aucune connexion n'existe entre ces deux interfaces de connexion et que les ressources internes du TCP sont suffisantes.

Le service de maintien de l'appel permet de procéder à l'échange de données entre les deux interfaces de connexion, assure le service de transport des données de même que celui de capacités spéciales.

Le service de cessation de l'appel : Ce service peut survenir brusquement ou se faire normalement. Dans le cas d'une cessation d'appel brusque, les données en transit peuvent être perdues. Par contre, dans celui d'une cessation normale d'appel on est sûr que, de part et d'autre, les données ont bien été reçues.

Les services de transport de données comprennent six sous-services :

- Le duplex intégral qui permet d'avoir une communication simultanée dans les deux sens;
- la transmission chronométrée qui ne dure que pendant le temps alloué;
- les segments TCP ordonnés en séquence, c'est-à-dire que l'ordre dans lequel les segments sont transmis est le même que l'ordre dans lequel ils sont reçus;
- l'étiquetage de sécurité et le degré de préséance sur toutes les données transférées; le contrôle de flux TCP permettant de régler le débit des données;
- le contrôle d'erreurs TCP qui fait appel à un checksum simple.

Data Stream Push – Utilisateur transmetteur de TCP
La fonction PUSH peut exiger de la couche TCP la transmission de toutes données en attente jusqu'à celles identifiées par le fanion PUSH. Coté réception, la couche TCP met normalement dans la mémoire tampon toutes données reçues, et les adresse à l'utilisateur de temps à autres. Lorsque des données marquées par le fanion PUSH sont reçues, la couche TCP envoie immédiatement au destinataire toutes les données se trouvant en mémoire tampon jusqu'au segment portant le fanion PUSH.

Signalisation de données urgentes
Ce genre de signalisation donne la possibilité d'informer le destinataire TCP que des données urgentes se trouvent dans le segment.

Rapport d'erreurs
TCP établit un rapport de panne de service lorsqu'elle ne peut être réglée.

12.3.2 Primitives de service TCP et IP

Les segments TCP échangés entre hôtes sont remis par la couche TCP du transmetteur à la couche IP par la primitive *Send*, et livrés par la couche IP à la couche TCP par le biais de la primitive *Deliver* lorsqu'ils sont reçus. La figure 12.4 en illustre le concept.

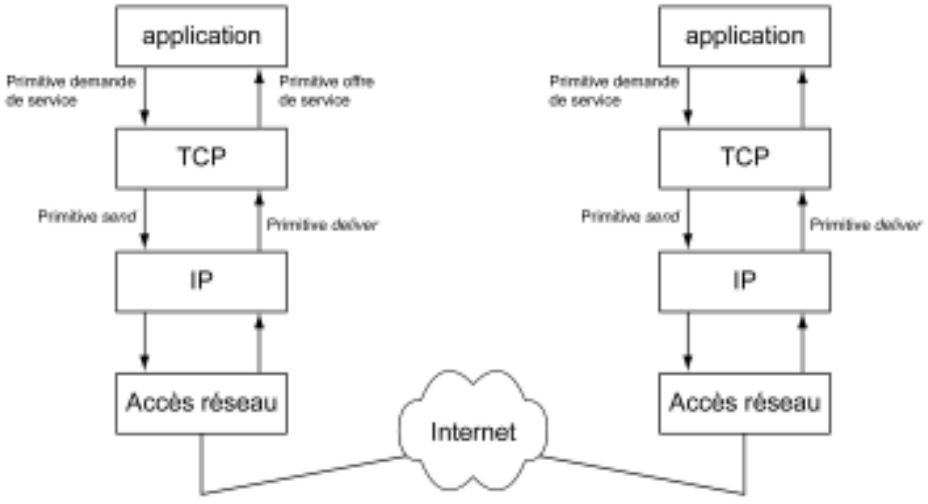


Figure 12.4 Primitives de service TCP et IP

12.3.3 Format d'en-tête TCP

Le format du segment TCP comprend un en-tête d'un minimum de 20 octets. Cet en-tête fournit les numéros des ports source et destination et le numéro de séquence des segments qui permet de conserver dans l'ordre les données reçues. Il y a aussi un champ pour l'acquittement des segments. Le numéro de séquence des segments acquittés est identifié dans ce champ avec une indication que ceci est un acquittement : le fanion ACK est alors mis à 1.

Le format du segment TCP se trouve à la figure 12.5.

Les champs du segment TCP sont les suivants :

- Numéro du port source. Long de 16 bits, il indique le numéro de port du processus d'application de l'hôte qui transmet le segment.

- Numéro du port destination. Long de 16 bits, il indique le numéro de port du processus d'application de l'hôte qui reçoit le segment.
- Numéro de séquence. Long de 32 bits, il donne le numéro de séquence du premier octet des données transmises dans ce segment. Lors de l'établissement d'un appel, le premier numéro de séquence est choisi de manière aléatoire.
- Numéro d'acquittement. Long de 32 bits, il indique le numéro de séquence du dernier octet accepté plus 1. C'est un acquittement positif pour tous les octets reçus jusqu'à celui indiqué moins 1.

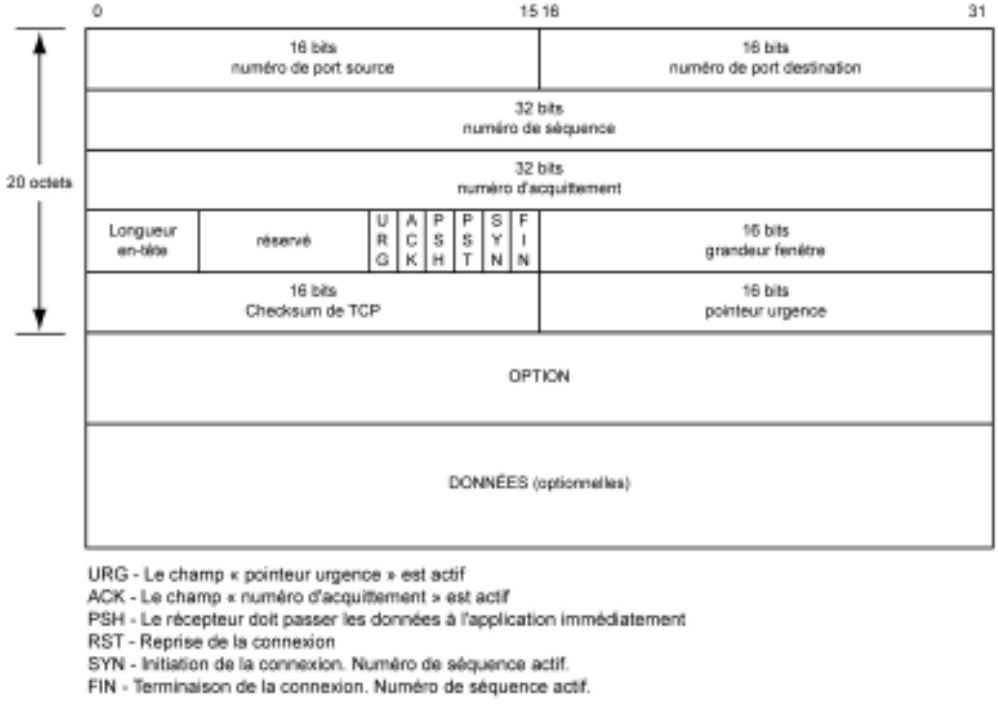


Figure 12.5 Segment TCP

- Longueur de l'en-tête : 4 bits. Il indique la grandeur de l'en-tête en multiples de 4 octets. Ainsi, si le contenu est égal à 5 cela signifie que l'en-tête a une longueur de 20 octets.
- Le champ « réservé » n'est pas utilisé.
- Les fanions de contrôle sont au nombre de six. Ils ont chacun une longueur de 1 bit et indiquent la nature du segment. La combinaison de certains d'entre eux est valide tout comme dans le cas des fanion SYN et ACK. Les six fanions sont les suivants :

- URG – signifie que le champ «pointeur d'urgence» est actif.
- ACK – Ce fanion a valeur d'acquittement. Aussi, le champ « numéro d'acquittement » doit alors être pris en considération.
- PSH – Le bit PUSH signale que la fonctionnalité décrite à la section 12.3.1 est activée. Il signifie au récepteur de remettre immédiatement les données à l'application sans perdre de temps à les réordonner. Ce cas peut être mis en évidence lorsque l'application comporte des critères de transmission rigides.
- RST est utile pour la reprise de la connexion.
- SYN – est utile pour amorcer un nouvel appel.
- FIN – est utilisé pour demander la libération d'un appel.

- Le champ « grandeur fenêtre » a une longueur de 16 bits. Il permet d'exercer un contrôle de flux. Ce cas sera également traité plus loin.
- Le champ Checksum de TCP sert à détecter les erreurs. Le calcul du checksum se fait de la même manière que dans le cas du checksum de UDP. On a cette fois-ci recours à un pseudo en-tête et au segment TCP (Cf. figure 12.6).

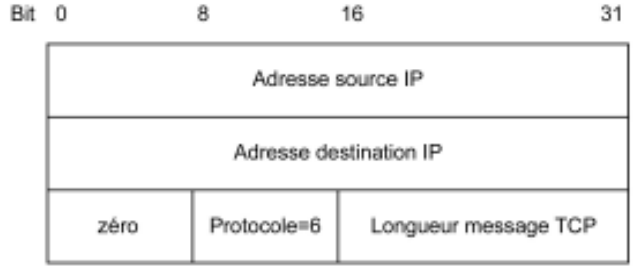


Figure 12.6 Pseudo en-tête TCP

- Le pointeur d'urgence est activé par le bit URG. Lorsque le transmetteur veut remettre des données urgentes au récepteur, il les place au début des données du segment, remet URG à 1 et insère le numéro de séquence de l'octet, où les données urgentes prennent fin, dans le champ « pointeur d'urgence ». Le reste des données du segment ne comportant que des données normales.
- Le champ « Option » (Cf. RFC 854 et RFC 1323) est un champ optionnel. Il peut renfermer jusqu'à 40 octets d'informations optionnelles.
    - Lors de l'établissement d'un appel l'option MSS (*Maximum Segment Size*) ou longueur maximale du segment est négociée. La longueur de cette option est de 4 octets.

1 octet	1 octet	2 octets
Kind=2	Len=4	Longueur maximale de segment (MSS)

- Une autre des options est la graduation de la fenêtre (*Windows scale*). Cette option nous permet d'avoir recours à des fenêtres plus grandes. Cette graduation peut aller de 2^1 à 2^{14}. Nous remarquons que le champ intitulé «grandeur fenêtre» nous permet d'avoir une fenêtre dont la taille maximale est de 65535 octets (i.e. 2^{16} - 1). Si par exemple l'option *Window Scale* est utilisée avec une valeur de 2^2 et que le contenu du champ « grandeur fenêtre » est de 65535 octets, la nouvelle fenêtre sera donc de 65535 x 2^2 = 262140 octets. La fenêtre maximale est de 65535 x 2^{14} = 1073725440 octets.

1 octet	1 octet	1 octet
Kind=3	Len=3	facteur

- La troisième option est l'horodateur (*Timestamp*). Cette option permet à chacun des segments d'indiquer le moment précis de la transmission. Cette option est importante lorsque le numéro de séquence a fait le tour du compteur. L'option permet de distinguer les segments qui pourraient porter le même numéro.

1 octet	1 octet	4 octets	4 octets
Kind=8	Len=10	Valeur de l'hodorateur	Réponse écho de l'hodorateur

- NOP (*No Operation*) est une option nulle, sans opération.

1 octet
Kind=1

12.3.4 Établissement d'appel

L'établissement d'un appel TCP se fait de chaque côté de la communication. Il faut que la demande de connexion soit transmise des deux côtés et l'acquittement doit être reçu pour chacune des demandes.

Lors de l'acquittement de la première demande d'établissement d'appel, plutôt que de transmettre deux trames, soit la trame ACK suivie de la trame SYN pour l'établissement de l'appel dans l'autre sens, une seule trame représentant la somme des deux (ACK + SYN) est transmise à la place.

De la sorte, la connexion TCP avec trois segments s'établit dans les deux sens. C'est la raison pour laquelle on s'y réfère souvent comme le «*three way handshake*».

Afin d'établir la connexion, le protocole TCP doit se trouver dans un état qualifié de « FERMÉ ». Cet état indique que la connexion est fermée et qu'il n'y a aucun lien d'établi. C'est à ce moment que la couche TCP du client transmet son PDU ou plus précisément son segment pour demander la connexion au serveur. Le premier segment transmis par le client comprend dans son en-tête l'indication ou le fanion SYN mis à 1. Il va de soi que le port destination doit correspondre à celui avec lequel il désire se connecter. Le numéro de la séquence doit être celui du premier segment transmis.

En réponse au segment SYN, le destinataire, dans le cas présent le serveur, qui accepte la connexion, envoie également un segment SYN (SYN=1) avec son numéro de séquence. Par contre, son segment doit aussi comporter un acquittement à l'endroit du premier segment du client. Ceci est indiqué par le bit ACK mis à 1 ainsi que par l'utilisation du champ «numéro d'acquittement» qui fait état du numéro de séquence du segment acquitté plus 1.

Le client doit alors envoyer un acquittement au SYN reçu du serveur. À son tour, il transmet alors un segment avec le bit ACK mis à 1 de même que le champ «numéro d'acquittement» qui renferme le numéro de séquence plus 1 du SYN reçu. Nous en donnons une illustration à la figure 12.7.

Bien que cela soit peu probable, il est possible de voir se présenter le cas où deux applications s'adressent des connexions mutuellement et simultanément. L'adressage mutuel implique que les deux connexions aient recours aux mêmes adresses et aux mêmes ports. Ces deux demandes simultanées de connexions donnent lieu à une ouverture simultanée (*Simultaneous Open*) ce qui signifie qu'il n'y a qu'une seule connexion d'établie dans les deux sens.

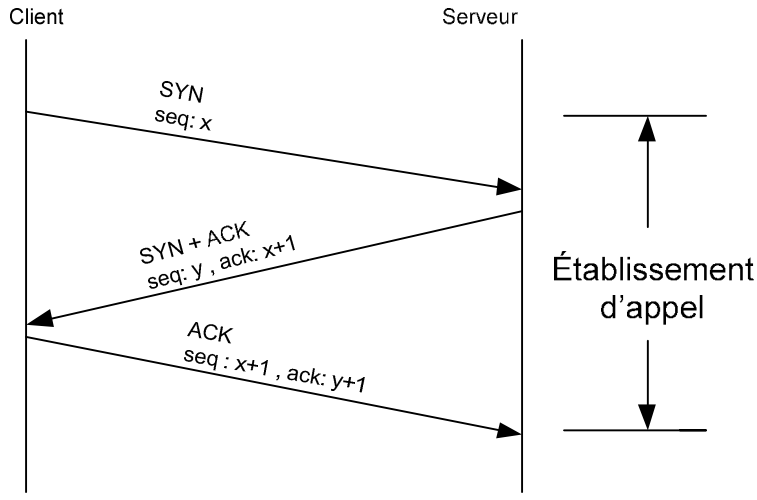


Figure 12.7 Établissement d'un appel

12.3.5 Échanges des données

Une fois l'établissement de l'appel fait, les données peuvent être échangées. Cet échange est réalisé par des segments les renfermant. Les données reçues sont acquittées par des segments ACK. Ces derniers peuvent aussi contenir des données.

Le numéro de séquence est celui du premier octet de données compris dans le segment. Les segments comprennent un nombre d'octets connus.

Le segment ACK acquitte ces données en indiquant le numéro de séquence du dernier octet plus 1 dans son numéro d'acquittement. La figure 12.8 illustre l'échange des données et leur acquittement.

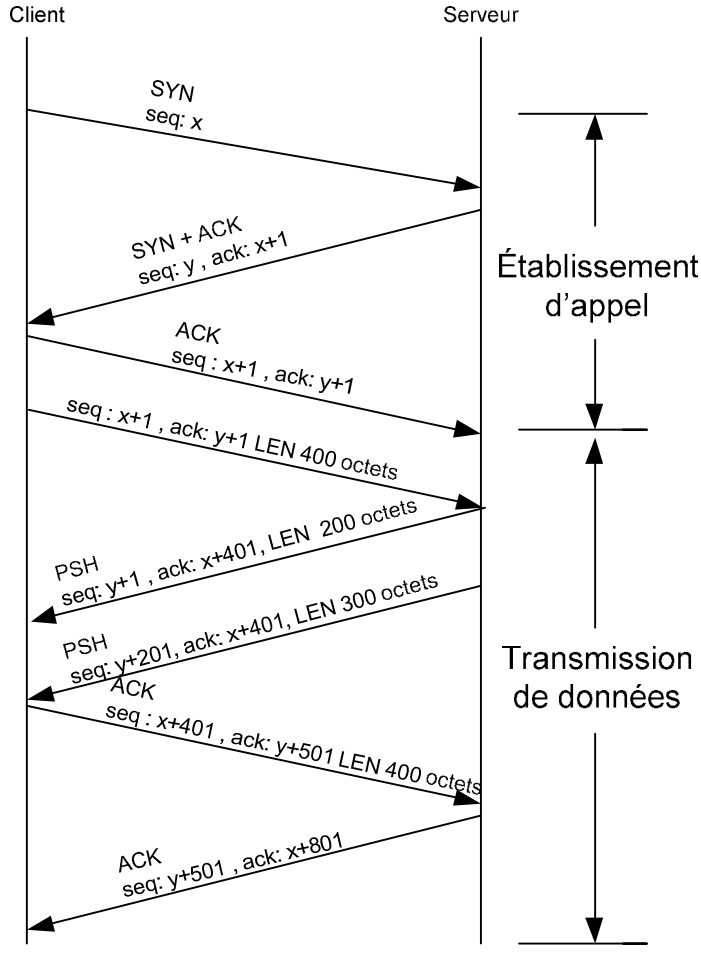


Figure 12.8 Échange des données

12.3.6 Libération de l'appel

La libération d'appel se fait en deux temps. Elle peut être initiée par le client ou le serveur. Le premier qui demande la déconnexion transmet un segment avec le bit FIN mis à 1. Il indique aussi le numéro de séquence de mises à jour et peut acquitter le dernier segment reçu en mettant le bit ACK à 1 et en incluant son numéro de séquence plus 1 dans le champ « numéro d'acquittement ». Son vis-à - vis transmet un segment avec le bit ACK mis à 1 en guise d'acquittement et avec le numéro de séquence du segment FIN plus 1. Par contre, ce même côté transmet aussi un segment FIN avec son numéro de séquence de même que le bit ACK mis à 1. Il reçoit par la suite un segment ACK en guise d'acquittement. La figure 12.9 en est l'illustration type. L'accusé de réception du premier FIN

reçu peut être combiné avec le FIN de l'autre coté. Puisqu'il s'agit de la fin de la transmission de données la combinaison du ACK avec FIN n'est pas importante.

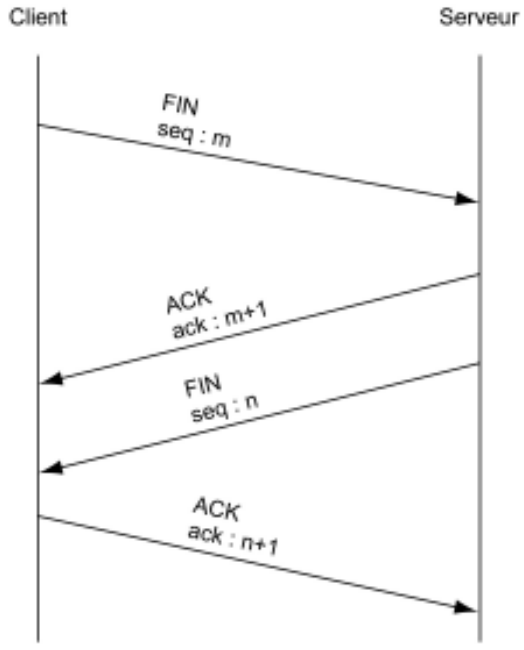


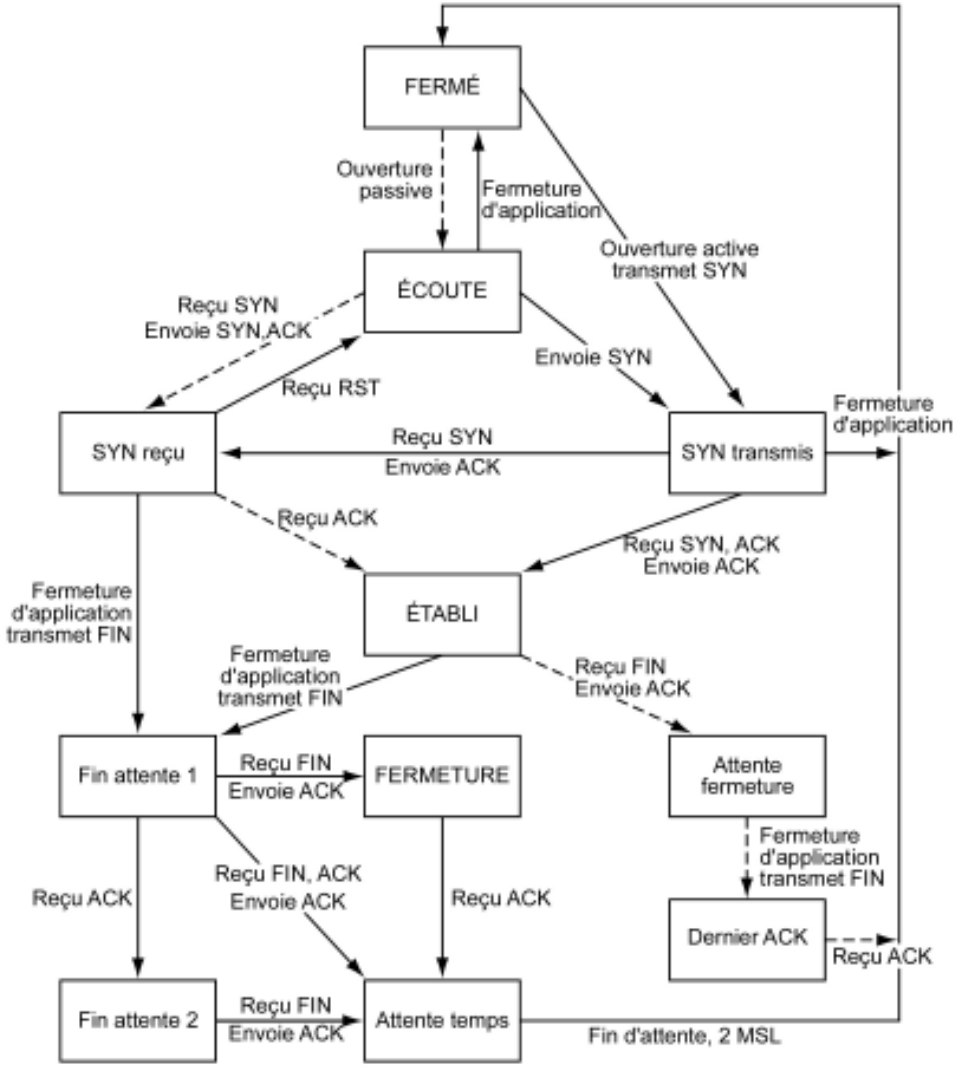
Figure 12.9 Libération d'appel

12.3.7 Machine d'état TCP

Le processus d'application fait savoir au système d'exploitation qu'il est prêt à recevoir un appel en activant la fonction «ouverture passive». C'est à ce moment seulement que le système d'exploitation lui assigne un numéro de port TCP.

Le processus d'application qui souhaite par contre établir une connexion, avise le système d'exploitation par le biais de la fonction « ouverture active », et établit une connexion par le biais de son interface de connexion en transmettant un segment SYN.

Le diagramme d'état de la figure 12.10 représente la dynamique du programme TCP. Les transitions entre les états montrent l'effet de la réception d'un certain segment et le segment transmis en réponse.



MSL - Maximum Segment Lifetime

Figure 12.10 Diagramme d'état pour l'établissement et la cessation de TCP

Le programme TCP débute toujours par l'état « **FERMÉ** ».

Lorsqu'un programme d'application fait appel à la fonction «ouverture passive», le programme TCP passe alors à l'état « **ÉCOUTE** ». S'il y a fermeture d'application on retourne à l'état « **FERMÉ** ».

Si le programme application veut établir une connexion, il a alors recours à la fonction « ouverture active », et il transmet un segment SYN puis se trouve à l'état **SYN transmis**.

- Si ce programme reçoit un segment SYN + ACK (demande de connexion et acquittement de sa propre demande), il envoie un segment ACK en guise de réponse et se trouve alors à l'état « **ÉTABLI »**. Le programme reste dans cet état jusqu'à ce que l'un ou l'autre des intervenants décide de mettre fin à l'appel.
- Advenant le cas où le programme reçoive un segment SYN, il adresse alors un segment ACK en guise de réponse et se retrouve dans l'état « **SYN reçu »** attendant de recevoir une confirmation à sa demande de connexion.

Lorsqu'il se trouve à l'état « **ÉCOUTE »**, le programme TCP est en attente. Si un segment SYN est reçu, il transmet un segment SYN + ACK aux fins d'acquittement de la première demande, et demandera à son tour une connexion. Il passe ensuite à l'état « **SYN reçu »**. Sitôt le segment ACK reçu, le programme passe à l'état « **ÉTABLI »**.

Le reste du diagramme d'état traite de la libération de l'appel.

À partir de l'état « **SYN reçu »** ou de l'état « **ÉTABLI »,** s'il y a fermeture d'application et que le programme TCP transmette le segment FIN, le diagramme d'état passe à l'état « **FIN attente 1 »**.

Si, de l'état **« ÉTABLI »**, un segment FIN est reçu, un segment ACK est transmis en réponse pour mettre fin à cet appel. Toutefois, on ne doit pas oublier que, dans l'autre sens l'appel est encore actif.

Le diagramme d'état doit assurer la libération des connexions dans les deux sens avant de revenir à l'état « **FERMÉ »**.

L'état « **Attente de temps »** doit patienter pendant 2 MSL avant de revenir à l'état « **FERMÉ »**.

Le MSL (*Maximum Segment Lifetime*) est la durée de vie maximum d'un segment dans un réseau avant qu'il ne soit mis de côté. Cette durée dépend pour beaucoup du datagramme IP. La RFC 793 qui consigne le TCP accorde au MSL une durée de deux minutes. Les versions appliquées ont recours à des valeurs plus courtes comme 30 secondes.

12.3.8 Contrôle de flux

Les données sont normalement placées dans la mémoire tampon et ne sont transférées à la couche supérieure qu'une fois cette mémoire pleine.

Toutefois, si le bit PSH est mis à 1, parce que le programme d'application ne peut traiter que peu de données à la fois, les données reçues par la couche TCP sont directement envoyées à la couche supérieure. PSH est utilisé lorsque les données sont interactives.

Tout comme dans le cas de l'application *Rlogin*, les données interactives sont transmises au rythme d'un caractère par segment et chacun de ces caractères attend de recevoir un écho (figure 12.11). Avant de transmettre un autre segment, un accusé de réception est reçu. En règle générale, l'écho et l'accusé de réception sont reçus par un seul et même segment.

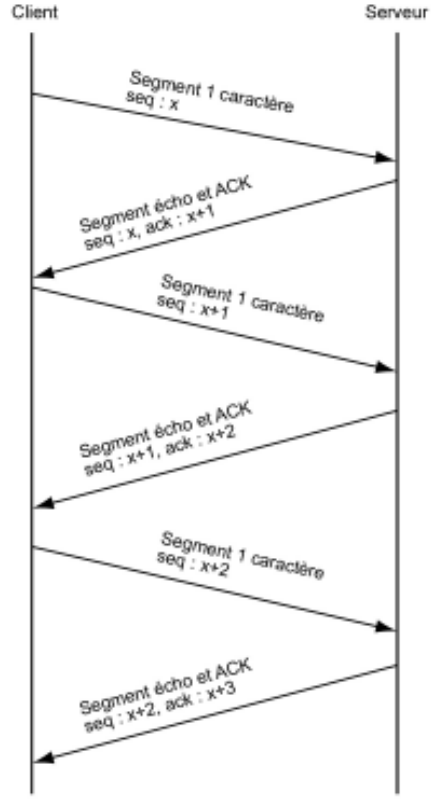


Figure 12.11 Transmission des données interactives

Cette interaction est connue sous le nom de syndrome de la fenêtre folle (*Silly Window syndrome*). Ce comportement se retrouve dans les situations ou le transmetteur envoie de petits segments ou le récepteur accepte lentement les données. Il est clair que dans ces situations la surcharge dans chaque segment est importante. Ainsi. Dans le cas précédent où la transmission est d'un octet de charge utile par segment alors que chaque segment comprend au moins 20 octets pour TCP en plus des 20 octets pour IP, l'efficacité est de 1/41 ou de 2,4 %.
L'algorithme de Nagle est une solution proposée permettant d'éviter que le transmetteur TCP continue à transmettre un octet à la fois. Cet algorithme améliore la réponse du protocole TCP pour les données interactives. Selon cet algorithme, un premier segment est transmis même s'il ne contient qu'un octet de données. En attente de l'acquittement, le transmetteur accumule les données dans la mémoire tampon sans les transmettre. Une fois l'acquittement reçu, le prochain segment transmis comprendra toutes les données accumulées entre-temps, tel qu'illustré à la figure 12.12. L'algorithme de Nagle prend donc en considération la vitesse du réseau et celle du programme d'application. Ceci diffère de la transmission interactive où, chacun des caractères est transmis séparément dans un segment.

Si la ligne est lente, au lieu d'attendre, les données successives sont acceptées et mises en mémoire jusqu'au moment où elles pourront être transmises par un seul segment. Ainsi nous voyons que, malgré la lenteur du lien, l'algorithme de Nagle optimise le temps d'attente et permet d'avoir une réception plus rapide.

Lorsque c'est le récepteur qui ralenti la réception des segments, il y a aussi un effet de *Silly Window syndrome*. Dans ce cas de figure, le récepteur vide lentement sa mémoire tampon. Si par exemple un octet est libéré, un acquittement est transmis avec une fenêtre d'un. Le transmetteur transmettra un segment d'un octet de données seulement. Ceci se répétera et créera l'effet *Silly Window syndrome*.
Une solution est proposée par Clark (RFC 813). Il s'agit de retourner un acquittement aussitôt les données reçues par le récepteur, par contre la fenêtre annoncée sera zéro. Ceci indique que les données sont bien reçues, mais que le transmetteur ne peut transmettre d'autres données. Une fois la mémoire tampon du récepteur se vide suffisamment ou de 50%, le récepteur renvoie un nouvel acquittement avec une fenêtre plus grande de la taille de l'espace disponible dans sa mémoire tampon.
Un temporisateur nommé *Persistent Timer* décrit plus loin, est requis lorsque l'acquittement avec une fenêtre zéro est utilisé. Ce temporisateur évitera une situation d'interblocage. L'interblocage est possible lorsque, après la réception d'un acquittement avec fenêtre zéro, un autre acquittement positif qui

permettrait de transmettre de nouveau segment n'est pas reçu. Le transmetteur et récepteur seraient tous deux en attente l'un de l'autre.
Le *Persistent Timer* est amorcé après l'envoi de l'acquittement comprenant la fenêtre zéro. Une fois le temps du temporisateur expiré, un segment *probe* est transmis par le transmetteur, indiquant au récepteur qu'il attend son acquittement. Le *probe* comprend un octet de données avec un numéro de série.

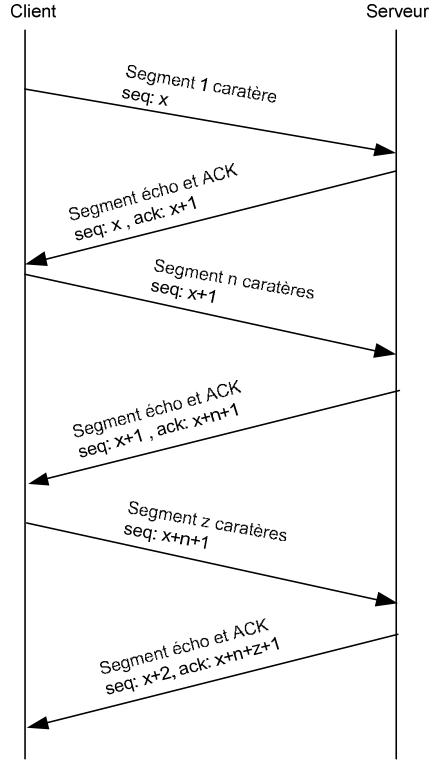


Figure 12.12 Application de l'algorithme de Nagle

12.3.8.1 Fenêtre coulissante

Lorsque les données sont transmises en bloc, une forme différente du contrôle de flux est utilisée. Il s'agit de la méthode qualifiée de « fenêtre coulissante » (*Sliding Window*). Ceci permettrait au transmetteur d'envoyer plusieurs segments avant d'attendre la réception d'un accusé de réception.
Le transfert de données est bien plus rapide parce que le transmetteur n'a pas à attendre d'acquittement après chaque transmission mais pourra mettre à profit ce temps pour transmettre d'autres segments. Selon la fenêtre coulissante du protocole TCP, le récepteur n'a pas besoin de transmettre des accusés de réception pour chaque segment reçu immédiatement. L'acquittement de trames

est cumulatif, ce qui signifie qu'un seul accusé de réception peut acquitter plusieurs octets à la fois. Les compteurs des segments représentent des octets.

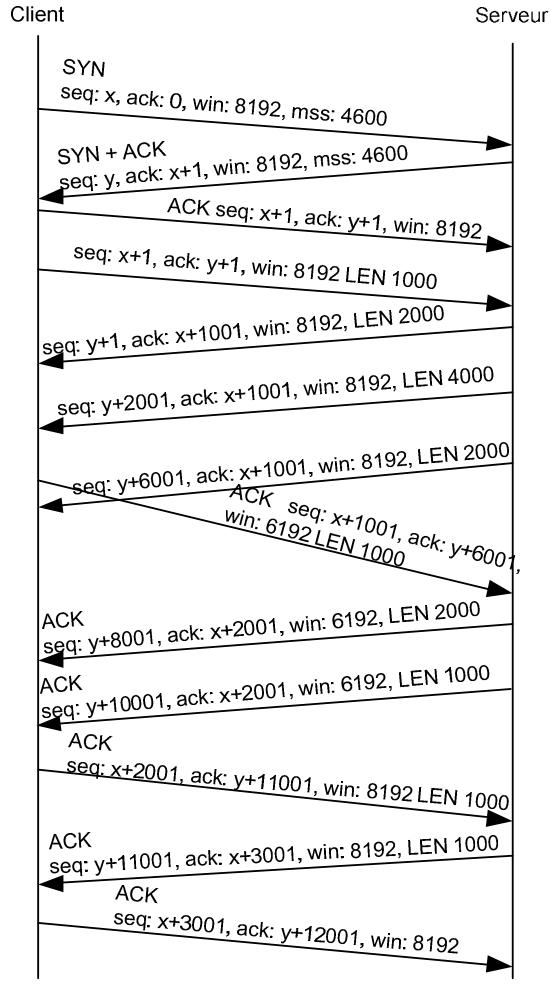


Figure 12.13 Application de la fenêtre coulissante

Soulignons à titre d'exemple la communication TCP représentée à la figure 12.13. La connexion sollicitée par le client établit la grandeur de la fenêtre win à 8192 octets et la grandeur maximale du segment mss à 4600 octets.
Le serveur transmet trois segments de données. Le premier comporte 2000 octets (len=2000), le second 4000 et le troisième 2000. Ainsi, un total de 8000 octets de données sera reçu par le client. La fenêtre étant de 8192 octets, le serveur ne peut pas transmettre plus de 192 octets. Toutefois, quoique le client ait reçu la totalité des octets (i.e. 8000) il n'en acquitte que 6000, ce qui ajuste la

fenêtre à 6192 octets. Notons que les 2000 octets non acquittés ne sont pas pour autant rejetés.

Les deux segments suivants du serveur au client comportent respectivement 2000 et 1000 octets. Sur réception du client, ce dernier transmet un accusé de réception acquittant de la sorte le dernier segment de la transmission précédente de même que les deux derniers segments. Ainsi, cet accusé de réception vient compléter l'acquittement des 11,000 octets reçus et la fenêtre reprend sa valeur initiale de 8192 octets.

On remarque aussi que les segments n'excèdent pas la grandeur maximale telle qu'établie lors de l'établissement de la connexion.

La rotation de la fenêtre dans le contrôle de flux du protocole TCP est semblable au mécanisme de rotation de la fenêtre telle que décrite dans le cadre du protocole LLC au chapitre 3. Toutefois, la différence, repose sur le fait que dans le cas présent la fenêtre se fonde sur le nombre d'octets transmis et non sur le nombre d'unités de données du protocole.

La figure 12.14 nous permet d'illustrer l'exemple ci-dessus en montrant la rotation de la fenêtre.

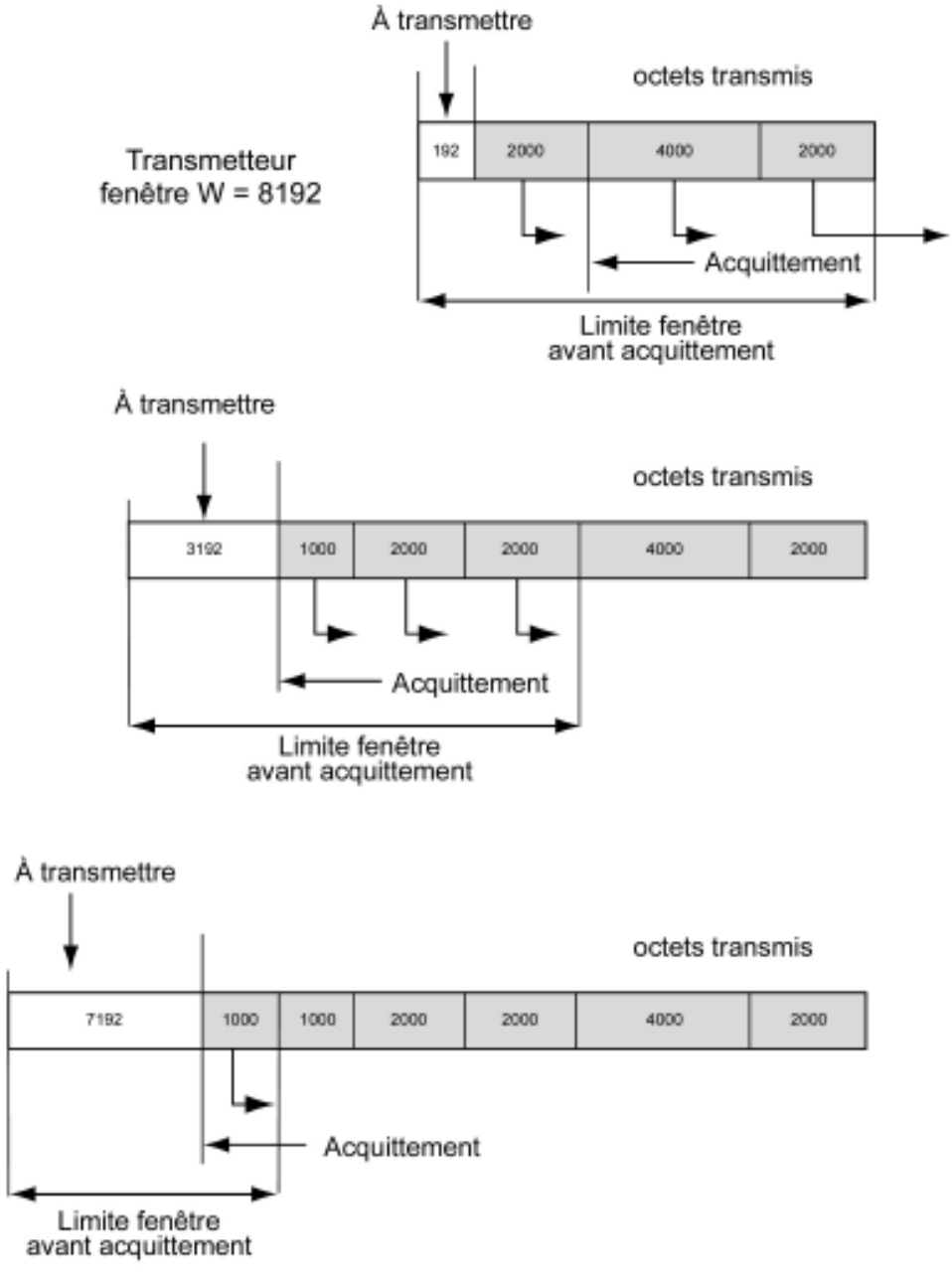


Figure 12.14 Rotation de la fenêtre pour le protocole TCP coté serveur

Ainsi et pour le récepteur, il devient dès lors possible de ralentir voire même de moduler la quantité des données émises par le transmetteur en acquittant une partie des données. Lorsque le transmetteur envoie le nombre d'octets que sa

fenêtre lui permet de transmettre, il attend que le récepteur en acquitte au moins une partie pour pouvoir continuer.

De la sorte et pour chacun des segments transmis, il y a un temporisateur qui s'enclenche. Les temporisateurs de segments ne sont arrêtés qu'une fois l'accusé de réception reçu et acquittant les octets du segment. Ces temporisateurs ont une durée de vie donnée, de sorte que lorsque cette durée expire le segment est considéré comme étant perdu et doit être retransmis.

La méthode de la fenêtre coulissante devient pratique pour le contrôle de flux car le transmetteur est plus rapide que le récepteur. Ainsi, si le transmetteur émet tous les octets permis par sa fenêtre et que le récepteur les reçoit sans erreurs, mais n'a pas le temps de les traiter, ce dernier peut renvoyer un accusé de réception acquittant tous les octets reçu, mais avec une fenêtre mise à 0.

Cet accusé de réception permet au transmetteur de remettre ses horloges de transmission à 0. Par contre, il devra, attendre la permission du récepteur pour continuer à transmettre. Cette permission prend la forme d'un nouvel accusé de réception envoyé par le récepteur lorsque celui-ci est prêt à recevoir, ayant le même acquittement mais indiquant une fenêtre plus grande que 0.

Si le récepteur détecte une erreur dans le segment, aucun accusé de réception n'est transmis. Rappelons que le protocole TCP n'a que des acquittements positifs. Après un certain temps, le temporisateur atteint la fin de sa durée de vie et le segment doit être retransmis.

12.3.9 Contrôle de congestion

Le contrôle de congestion a pour but d'éviter les pertes inutiles de segments sans pour autant nuire à leur débit de transmission. Le protocole TCP offre un service de transport fiable. Le contrôle de congestion de TCP est effectué de bout en bout de la source à la destination et n'est pas fonction de la congestion du réseau. Ceci indique que lorsque le réseau n'est pas congestionné il pourrait y avoir une congestion au niveau TCP.

Lorsqu'une perte de segments est décelée, cela signifie qu'il y a congestion et on se doit d'appliquer une réduction du débit.

La RFC 2581 apporte une solution au contrôle de congestion pour le protocole TCP. Quatre algorithmes de contrôle de congestion interdépendants y sont consignés. Il s'agit des algorithmes suivants :

- Démarrage lent (*Slow Start*)
- Évitement de la congestion (*Congestion avoidance*)
- Retransmission rapide (*Fast retransmission*)
- Récupération rapide (*Fast recovery*)

Ces quatre algorithmes sont mis en application avec certaines variantes dont nous ferons état. Il faut rappeler que les travaux de Jacobson[1] sont à l'origine même de ces algorithmes.

Soulignons que chacun des segments reçoit son propre accusé de réception. Le temps d'arrivée de ce dernier doit être plus bref que celui du temporisateur de retransmission qui n'est amorcé qu'une fois le segment transmis.

12.3.9.1 Démarrage lent (Slow Start)

Slow start ou démarrage lent, permet de sonder le réseau dans lequel les conditions ne sont pas très bien connues. Le protocole TCP doit de ce fait questionner lentement le réseau pour en établir sa capacité.

Slow start utilise la fenêtre de congestion **cwnd** (*congestion window*) qui est un contrôle de flux imposé par le transmetteur. Cette fenêtre s'ouvre au début de la connexion avec la valeur de 1 MSS (*Maximum Segment Size*). Lors de chaque accusé de réception, la fenêtre de congestion est augmentée de 1 MSS par segment acquitté. Soulignons que, quoique les acquittements ne se font que pour les octets et que chaque segment peut avoir une grandeur maximale, les accusés de réception ne sont appliqués qu'aux segments tout en acquittant les octets qui se trouvent dans le segment.

Après le premier accusé de réception, la fenêtre est mise à deux. Deux segments peuvent alors être transmis sans devoir attendre un nouvel accusé de réception. Cela revient à dire aussi qu'il ne peut y avoir plus de deux segments non acquittés. Après avoir obtenu un accusé de réception, la fenêtre est mise à quatre. Puisque deux segments sont acquittés, on ajoute 1 MSS au cwnd pour chacun d'entre eux, ce qui signifie tout simplement que l'on double le contenu de cwnd. Ainsi nous constatons que la croissance de la fenêtre est exponentielle.

Une autre variable d'état est utilisée pour faire état du seuil à partir duquel cette croissance exponentielle de la fenêtre de congestion **cwnd** doit être atténuée. Il

[1] V. Jacobson, « *Congestion Avoidance and Control* », Computer Communication Review, vol.18, no.4, pp. 314-329, Aug. 1988.

s'agit du seuil du démarrage lent ou *slow start threshold* **ssthresh**. La valeur de cette variable est initialement élevée. Toutefois, elle est arbitraire.

Lorsque la fenêtre de congestion atteint le **ssthresh**, l'algorithme du contrôle de congestion change d'algorithme pour passer de *Slow Start* à *Congestion avoidance*. En d'autres termes, l'algorithme *Slow Start* est utilisé lorsque **cwnd** < **ssthresh**. L'algorithme *Congestion Avoidance* est appliqué lorsque **cwnd** > **ssthresh.** Lorsque **cwnd**=**ssthresh** le transmetteur peut avoir recours à l'un ou l'autre des algorithmes.

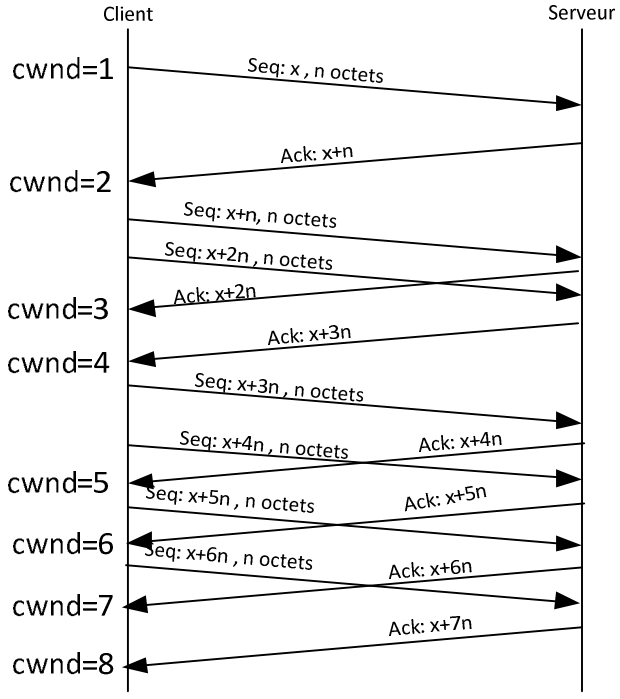


Figure 12.15 Croissance de la fenêtre de congestion (démarrage lent)

Dans le cas de la figure 12.15 où la rotation de la fenêtre et sa croissance sont illustrées pour le *Slow Start*, les accusés de réception sont transmis pour chaque segment de données.

La figure 12.16 illustre un autre cas similaire où les accusés de réception ne sont pas transmis sitôt les segments reçus.

Dans les 2 cas, la fenêtre de congestion **cwnd** est augmenté de un pour chaque acquittement (ACK) reçu.

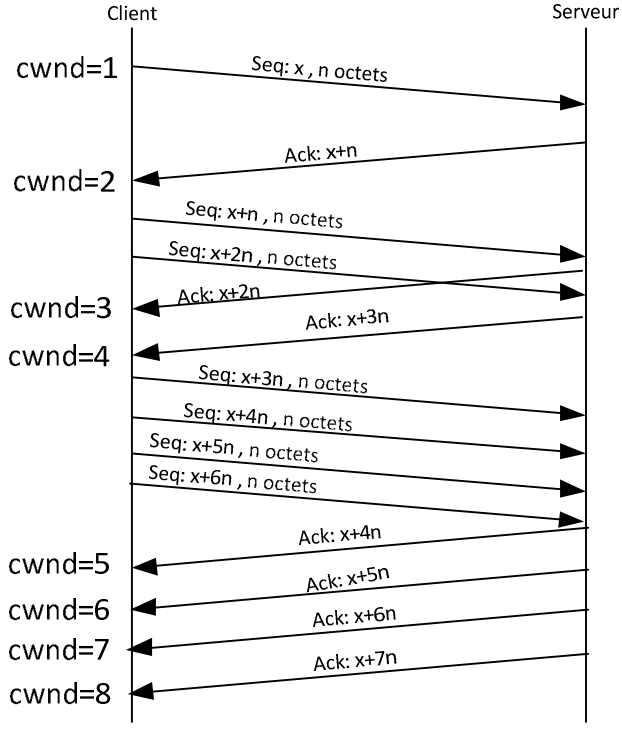


Figure 12.16 Croissance de la fenêtre de congestion : autre exemple

12.3.9.2 Évitement de la congestion (Congestion avoidance)

Congestion Avoidance est appliqué en faisant croître de un MSS la fenêtre **cwnd** lorsque chaque accusé de réception est reçu. En d'autres termes, **cwnd** est accru de 1 MSS à chaque aller-retour ou RTT (*Round Time Trip*). Cette croissance est additive. Elle n'est appliquée qu'aux acquittements de segments non retransmis.

Lorsque le routeur chargé d'acheminer les segments commence à en détruire, cela signifie que la fenêtre de congestion est trop large. Cette valeur est donc la grandeur maximale que la fenêtre peut atteindre.

Lorsque le temporisateur de retransmission d'un segment du transmetteur expire avant de recevoir un accusé de réception, la valeur du seuil **ssthresh** est réduite de la moitié du **cwnd**. Par la suite le **cwnd** est remis à un.

12.3.9.3 TCP Tahoe

TCP Tahoe se comporte tel que décrit précédemment. De plus, lorsque le transmetteur reçoit trois accusés de réception dédoublés, la valeur du seuil **ssthresh** est réduite de moitié de celle du **cwnd** courant. Par la suite le **cwnd** est remis à un. La conséquence de cela en est le retour à l'algorithme *Slow Start* avec une nouvelle valeur de seuil **ssthresh**.

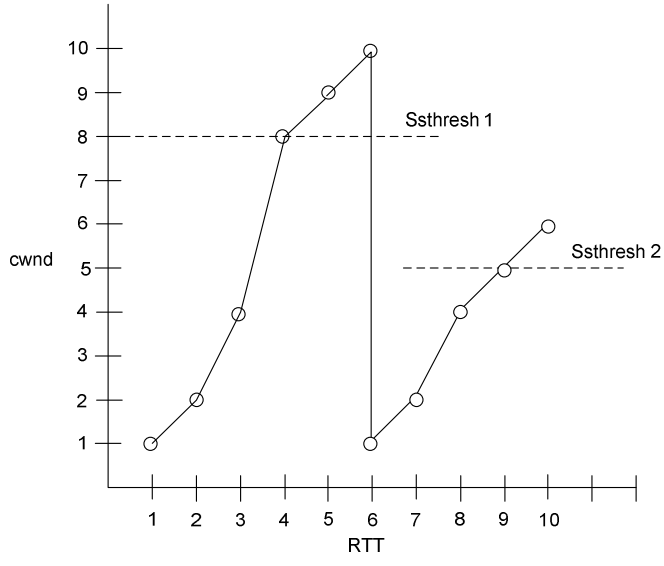


Figure 12.17 Fenêtre de congestion sous TCP Tahoe

La figure 12.17 nous donne un exemple du comportement de la fenêtre de congestion **cwnd** sous TCP Tahoe. La particularité de TCP Tahoe, par rapport à la méthode dont nous traiterons plus loin, réside dans le cas de la réception de trois accusés de réception dédoublés. Cela nous signale qu'il y a au moins un segment qui a été transmis après ces accusés de réception et qui est perdu ou bien, qu'un segment nous parvient en désordre. La retransmission de l'accusé de réception se fait normalement après réception de chaque segment, qui n'est pas celui que l'on attend. Les trois retransmissions de l'accusé de réception indiquent qu'il y aurait eu possibilité que les trois segments aient été reçus mais avec des numéros de séquence différents de ceux attendus.

12.3.9.4 TCP Reno

Tout comme dans le cas précédent, TCP Reno[2] se conforme à la méthode de récupération lors de la fin de la durée de vie du temporisateur de retransmission. Toutefois, lorsque le transmetteur reçoit trois accusés de réception dédoublés, la valeur du seuil **ssthresh 1** est réduite de la moitié de celle du **cwnd** courant. Par la suite, contrairement à TCP Tahoe, le **cwnd** est remis à la moitié de sa valeur, c'est-à-dire à la même valeur que **ssthresh 2**. La croissance du **cwnd** reste linéaire à partir de ce point.

Il est bon de mentionner que TCP Tahoe n'est pratiquement plus utilisé remplacé par TCP Reno qui est bien plus répandu.

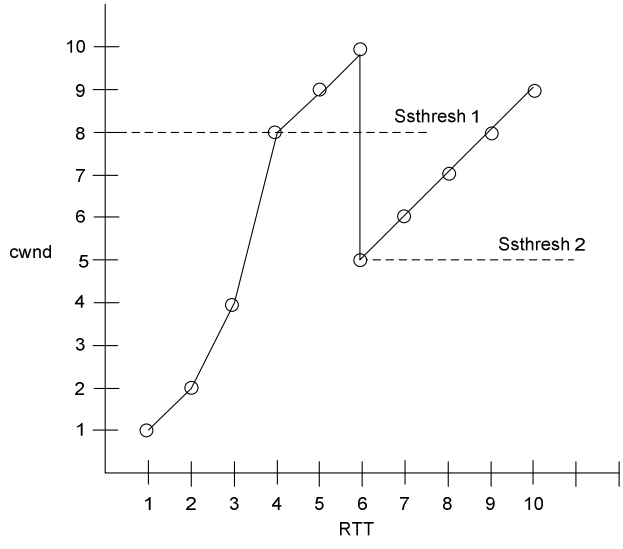


Figure 12.18 Fenêtre de congestion sous TCP Reno

La figure 12.18 illustre le comportement de la fenêtre de congestion **cnwd** sous TCP Reno.

12.3.9.5 Retransmission rapide et récupération rapide - new Reno

Cette approche est aussi connue sous le nom de **TCP NewReno** et fait l'objet de RFC 2582. Tout comme le TCP Reno, cette méthode est distincte advenant le

[2] K. Fall, S. Floyd, « *Simulation-based Comparison of Tahoe, Reno and SACK TCP* », Computer Communication Review, July 1996.

cas de la réception de trois accusés de réception dédoublés. Dans ce cas de figure, illustré à la figure 12.19, les étapes suivantes sont appliquées :

- le transmetteur réduit la valeur du seuil **ssthresh** de la moitié de celle du **cwnd**.
- Le segment perdu est retransmis.
- Le **cwnd** est mis à ssthresh + 3 MSS (pour permettre de transmettre les trois segments qui se trouvent encore dans la mémoire tampon et qui ne sont pas acquittés.)
- Chaque fois qu'un accusé de réception dédoublé est reçu, le **cwnd** est accru d'un MSS.
- Lorsqu'un nouvel accusé de réception est reçu, la valeur du **cwnd** est remise à celle du **ssthresh**.
- À partir de ce point, la croissance du **cwnd** devient additive, c'est-à-dire que la valeur du **cwnd** augmente de 1 MSS pour chaque RTT.

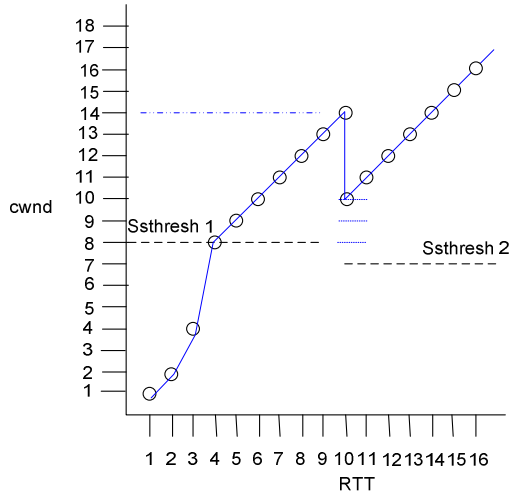


Figure 12.19 Fenêtre de congestion sous TCP New Reno

12.3.10 Temporisateurs, RTT (round Time Trip) et RTO (retransmission timeout)

Lorsqu'un segment est transmis son temporisateur est amorcé. Le temps d'expiration de ce temporisateur ainsi que le nombre de retransmissions possibles restent à déterminer.

Pour chaque connexion, il existe quatre différents temporisateurs.

- Le temporisateur de retransmission (*Retransmission Timer*) – lorsque l'on attend un accusé de réception.
- Le temporisateur de persistance (*Persist Timer*) – pour continuer d'évoquer la grandeur de la fenêtre.
- Le temporisateur de survie (*Keepalive timer*) – pour savoir quel est l'état à l'autre bout.
- Le temporisateur 2 MSL qui désigne le temps passé dans l'état **Attente temps**. Le MSL (*Maximum Segment Lifetime*) a été expliqué à la section 12.3.7.

Le temps de retransmission est fondé sur la mesure du RTT des segments TCP. Le RTT est continuellement mis à jour. Sa nouvelle estimation est calculée sur sa valeur précédente et sur la dernière mesure aller-retour.

La RFC 2988 consigne l'algorithme qui devra être utilisé par le transmetteur de segment TCP dans le calcul du temps d'attente du temporisateur de retransmission. Cet algorithme fait appel à deux variables, notamment le temps aller-retour arrondi SRTT (*Smoothed Round Trip Time*) et la variation du temps aller-retour RTTVAR (*Round Trip Time Variation*). Une valeur « g » qui désigne la granularité de l'horloge (*clock granularity*) fait partie aussi du calcul. La RFC 2988 suggère la valeur de 100 msec pour un rendement plus rapide de « g ».

L'algorithme de base permettant d'établir la durée de vie du temporisateur de retransmission RTO (*Retransmission Timeout*) est défini comme suit :

- Initialement, le calcul de démarrage de l'algorithme est fondé sur les estimations suivantes :
    - La valeur initiale de RTT est mise à 3 secondes.
    - Le calcul de l'estimation de RTT commence par la mesure initiale de RTT représentée par R. De la sorte, il est possible d'évaluer le RTO comme suit :
      SRTT = R
      RTTVAR = R/2
      RTO = SRTT + max (g, K·RTTVAR)

      K est une constante, K=4.

- Les mesures subséquentes du RTT sont annotées par R'. Par voie de conséquence, le calcul du RTO par chacune des itérations à venir est donc selon ces opérations :
    - $\mathbf{RTTVAR = (1 - \beta) \cdot RTTVAR + \beta \cdot |SRTT - R'|}$

SRTT = (1 – α) · SRTT + α R'

Dans ce cas, α=⅛ et β=¼

Notons que la valeur de SRTT dans le calcul du RTTVAR est celle d'avant la valeur recalculée de SRTT. En d'autres termes, pour chaque itération RTTVAR est calculé avant SRTT.

- RTO est alors mis à jour de la façon suivante :

  RTO = SRTT + max (g, K·RTTVAR)

  La valeur de RTO est d'au moins 1 seconde et ne doit jamais dépasser 60 secondes.

Signalons que si le RTT est trop bref, la retransmission peut être plus fréquente et éventuellement non nécessaire. Par contre, si le RTT est trop long la communication s'en trouve ralentie.

La proposition de Karn[3] de ne pas calculer l'évaluation du RTT pour un segment retransmis est appliquée lors de toute mise en vigueur de TCP. Cela permet d'éviter qu'il y ait confusion du côté du transmetteur à savoir si l'accusé de réception reçu est pour le tout premier segment ou pour le segment retransmis. Il est clair que dans une telle situation le RTT ne peut pas être évalué avec précision.
La capacité du conduit dépend de la largeur de bande et du délai d'aller-retour (*RTT – pour Round Trip Time*).

Cette capacité est représentée de la façon suivante :

C=W x R

Où,

C est la capacité, bits
W est la largeur de bande, bit/sec
R est le RTT, seconde

Nous voyons que, si le RTT est doublé, la capacité du conduit l'est aussi. La même situation prévaut avec la largeur de bande.

[3] Karn P., Partridge C., « *Improving Round-Trip Time Estimates in Reliable Transport Protocols* », Computer Communication Review, vol. 17, no.5, pp.2-7, (Aug 1987)

La congestion est l'effet de la transmission de données à haut débit acheminée dans un conduit à petit débit.

12.3.10.1 Temporisateur de persistance (Persist timer)

Lorsque les segments de données sont transmis, ils sont acquittés par des segments d'accusé de réception. Par contre, ces derniers ne sont pas acquittés à leur tour.

Imaginons un terminal A qui a transmis tous les segments que lui permet sa fenêtre. Si l'accusé de réception du terminal B permettant la rotation de cette fenêtre venait à être perdu, le terminal A n'aurait aucun moyen de le savoir et les deux terminaux s'attendraient l'un l'autre. Afin d'empêcher l'inter blocage, le terminal A ou tout transmetteur de segments de données a alors recours au temporisateur de persistance. Grâce à celui-ci, le transmetteur questionne périodiquement le récepteur pour savoir si la fenêtre peut être accrue. Il utilise des segments de 1 octet connus sous le nom de *window probes*. Ces segments sont transmis à des intervalles de temps croissants jusqu'à recevoir un accusé de réception permettant la rotation de fenêtre. Le premier intervalle de temps se trouve à 1,5 seconde et chaque intervalle de temps suivant est retardé de 1,5 seconde de plus que l'intervalle de temps précédent. Lorsque les 60 secondes sont atteintes, les probes continuent d'être transmis au même intervalle de temps.

Lorsque, pour diverses raisons, le récepteur ne peut pas allouer une grande fenêtre et, au lieu d'attendre de pouvoir le faire permet l'échange de petites quantités de données, nous sommes face à un problème où la connexion est occupée par un échange continu de petits segments de données et d'accusés de réception.

Ce problème se nomme le syndrome de la fenêtre folle (*Silly window syndrome*) décrit plus haut. Pour l'éviter, le récepteur doit s'astreindre à transmettre de petites fenêtres et à ne permettre la transmission de données que lorsque la fenêtre peut être augmentée de la plus petite valeur des choix suivants : soit la grandeur maximale d'un segment ou de la moitié de sa mémoire tampon.

De son côté, le transmetteur, peut éviter ce problème en ne transmettant qu'un plein segment ou au moins la moitié du maximum de la fenêtre permise.

12.3.10.2 Temporisateur de survie (Keepalive Timer)

Lorsqu'une connexion TCP est établie, elle le reste indéfiniment, et ce, jusqu'à ce qu'il y ait une demande de libération de la connexion. Les modules TCP

pourraient ne pas s'échanger de segment d'aucune sorte sans affecter la connexion. Même si des routeurs, utilisés pour leurs interconnections, venaient à tomber en panne, la connexion TCP n'en serait pas affectée.

Pourtant, il existe des situations dans lesquelles un des terminaux tombe en panne et, l'autre terminal n'en étant pas avisé, conserve la connexion active.

Il existe un temporisateur optionnel qui pourrait être utilisé pour savoir si un terminal est en panne. Il s'agit du temporisateur de survie (*keepalive timer*). Une fois la durée de vie de ce temporisateur échue, le terminal transmet une sonde pour interroger son interlocuteur. Le temps de survie du temporisateur pourrait être de deux (2) heures. Une fois la sonde transmise, le terminal attend 75 secondes pour obtenir une réponse à sa demande. S'il n'a toujours pas de réponse, le terminal retransmet la sonde avec le même temps d'attente pendant un maximum de 10 fois pour obtenir une réponse. S'il ne reçoit toujours pas de réponse, le terminal considère alors que l'autre terminal est en panne et met fin à la connexion.

Le problème de ce temporisateur est qu'il pourrait parvenir à échéance et déconnecter une communication qui n'a pas de problème. Ceci est possible par exemple, lorsque la sonde est transmise au moment où un routeur se remet de sa panne. Si le routeur n'est pas encore rétabli d'ici la fin de la vérification, la connexion prend fin.

12.4 Exercices

12.1

Deux terminaux établissent une connexion TCP avec un serveur pour des applications FTP.

Le terminal 1 a l'adresse IP 192.168.100.62
Le terminal 2 a l'adresse IP 192.168.200.5
Le serveur a l'adresse IP 192.168.300.1

a) Indiquez les numéros de port utilisés dans ces connexions et indiquez de quels types ils sont.
b) Indiquez les interfaces de connexion pour chacune des connexions.

12.2

Expliquez en quoi l'algorithme de Nagle peut être plus efficace que la transmission interactive.

Dans quel cas l'algorithme de Nagle ne peut pas être utilisé?

12.3

Un serveur a 10300 octets de données à transmettre. Il utilise le protocole TCP pour transférer ses informations. Pour le contrôle de flux il applique la fenêtre coulissante qui est initialement de 8192 octets. Le MSS est à 4600 octets.

Si les RSDUs sont transmis dans un réseau dont les datagrammes sont limités à 1024 octets,

a) Donnez le nombre de segments qui seront transmis par la couche TCP. Indiquez également la grandeur de chaque segment et expliquez votre calcul.
b) Puisqu'une fragmentation aura lieu, indiquez le nombre et la longueur des datagrammes qui seront formés pour être transmis par la couche IP et expliquez votre calcul.

12.4

La connexion sollicitée par le client établit la grandeur de la fenêtre win à 8000 octets et la grandeur maximale du segment mss à 4600 octets.

Le serveur transmet dix segments de données. Chaque segment comporte 4000 octets (len=4000). Le récepteur transmet un accusé de réception chaque fois que la fenêtre est pleine, mais pas avant.

Établissez les échanges de segments selon le format de la figure 12.13 en y indiquant tous les détails.

12.5

a) Expliquez en quoi l'évitement de la congestion appliquant la méthode Reno est une amélioration de la méthode Tahoe.
b) Expliquez en quoi New Reno améliore l'évitement de la congestion par rapport à la méthode Reno.

12.6

Le transmetteur de segment TCP établit le calcul du temps d'attente du temporisateur de retransmission. L'algorithme du RTO (*retransmission timeout*) fait appel à deux variables, notamment le temps aller-retour arrondi SRTT (*Smoothed round trip time*) et la variation du temps aller-retour RTTVAR (*Round trip time variation*). Une valeur « g » qui désigne la granularité de l'horloge (*clock granularity*) qui fait partie aussi du calcul prend la valeur de 100 msec. De plus la constant K utilisé dans l'algorithme est 4 et comme il a été mentionné dans le texte les autres constantes sont $\alpha=1/8$ et $\beta=1/4$. Le temps aller-retour initial est 3 sec.

Si la valeur subséquente de temps aller-retour RTT (*Round Time Trip*) commence à 10 secondes et que cette valeur est incrémentée chaque fois d'une seconde, déterminez à quelle valeur de RTT le RTO dépassera 50 secondes.

a) Indiquez vos calculs de SRTT, RTTVAR et RTO pour chaque valeur de RTT.
b) Faites un graphe de RTT vs RTO.

13 Les applications sur Internet

13.1 Introduction

La couche application du modèle TCP/IP renferme de nombreux protocoles. L'objectif de certains d'entre eux est d'apporter un soutien au réseau de communication. Quelques-uns de ceux qui furent traités au chapitre 10 en font partie. Par contre, pour certains autres il s'agit de programmes dont le but est propre à l'application. Ces derniers desservent un besoin en particulier, non pas en regard du réseau, mais plutôt en regard de l'utilisateur de la station.

Parmi ces programmes, nous retrouvons ceux qui permettent de naviguer dans la toile et d'obtenir des informations. Il en va ainsi, du protocole HTTP exploité à ces fins. Un autre programme est employé pour procéder au transfert des fichiers. Tel est le cas du protocole FTP. L'échange de courriels se fait par le biais d'autres programmes formant des protocoles de la couche Application. Il s'agit entre autres, du protocole SMTP.

Bien que d'autres protocoles d'application existent, ce chapitre traitera exclusivement des protocoles d'application cités ci-haut.

13.2 Le protocole HTTP

L'accès à la toile se fait par l'entremise du protocole d'application HTTP (*Hypertext Transfer Protocol*). Ce protocole fait appel au concept « client-serveur ». À lui seul, il ne s'avère pas être d'une grande utilité sans le recours à un programme spécialisé qui lui tient lieu d'application du client. Il s'agit du fureteur ou *Browser* en anglais. D'autre part un ou plusieurs serveurs web sont indispensables pour répondre aux requêtes du client et lui envoyer les pages demandées. Ainsi, le client qui se trouve dans le fureteur et le serveur web communiquent en s'échangeant des messages « HTTP ». C'est le client qui en tout temps amorce la communication en transmettant une requête.

Le protocole HTTP en est un de la couche application. Il utilise les services de la couche transport et plus précisément le protocole TCP. HTTP est un protocole sans machine d'état. Cela signifie qu'il ne mémorise pas les échanges qui se tiennent entre client et serveur. Le protocole HTTP a évolué depuis son introduction. Il en est à la version HTTP 1.1 qui est détaillée dans le RFC 2616.

Le fureteur est un programme installé dans les stations et, outre le fait de transmettre des requêtes client aux serveurs web, interprète les données que ces derniers lui renvoient et les affichent de façon appropriée.

Les pages que le serveur transmet peuvent contenir plusieurs types d'objets. Ces derniers peuvent aller des simples fichiers à du texte ou encore à une image, du son, ou bien à des vidéos. Les pages sont adressées par des URL (*Uniform Resource Locator*) qui indiquent le serveur adressé ainsi que les pages requises.

Un URL est la forme la plus utilisée d'un URI (*Uniform Resource Identifier*). Un URL se compose principalement de l'identifiant du protocole utilisé dans la requête du fureteur, soit « HTTP », du nom du serveur auquel le fureteur s'adresse, par exemple « www.etsmtl.ca », et du nom de la ressource ou page demandée, tout comme : « soutien.html ». Ainsi, notre URL aurait la forme suivante : « http://www.etsmtl.ca/soutien.html ». Il est bon de signaler que la page peut être définie non pas seulement par son nom, mais également par la partie du nom qui identifie le chemin qui permet de l'atteindre. Dans le cas d'une page autre que la précédente nous pouvons en préciser l'objet tout comme : « /zone2/programmes/index.html ».

Le fureteur affiche les pages reçues en se fondant sur la norme de présentation HTML (*Hypertext Markup Language*). Cette dernière est utilisée pour formater du texte, créer des liens *hypertext* vers d'autres objets, faire référence à des images, à du son, ou à des Applets Java. La syntaxe du langage HTML est des plus simples. Un document, pris dans sa forme la plus rudimentaire, est énoncé en caractères ASCII. Le présent chapitre ne traitera pas de la façon de préparer une page HTML. Nous nous pencherons plutôt sur les protocoles employés dans cette application.

13.2.1 Session HTTP

Une session typique HTTP est amorcée par le fureteur. En tout premier lieu, ce dernier doit obtenir l'adresse du serveur. Il demande au DNS de la lui envoyer. Une fois l'adresse obtenue, le client ou si l'on veut, le fureteur, établit une connexion TCP avec le serveur. Les processus application du client et du serveur accèdent le protocole TCP en adressant le port 80. Le serveur lui renvoie l'objet

demandé. Chaque objet peut être adressé grâce à un URL et constituer dans l'ensemble une page web.

Un client a souvent besoin de plusieurs pages et objets du même serveur. Il n'a pas besoin de faire de nouvelle requête au DNS puisque l'adresse est mémorisée dans son antémémoire.

Par contre, la question qui se pose est : le client devrait-il établir une connexion TCP pour chaque objet demandé au serveur? Si tel est le cas, ceci préconise que le client devra se connecter au serveur au niveau TCP, faire sa demande d'objet au serveur au plan application avec le protocole HTTP, puis déconnecter l'appel TCP. Ceci devrait être fait pour chacun des objets. C'est la conception qui est privilégiée par la version 1.0 du protocole HTTP. On la désigne par l'expression suivante : « connexion non persistante ».

13.2.2 Connexion persistante

Pour gagner du temps, l'accusé de réception transmis par le client à la demande d'établissement d'appel du serveur peut inclure la requête de l'objet. La figure 13.1 nous donne une illustration de la situation lorsque nous avons recours à la méthode de « connexion non persistante ».

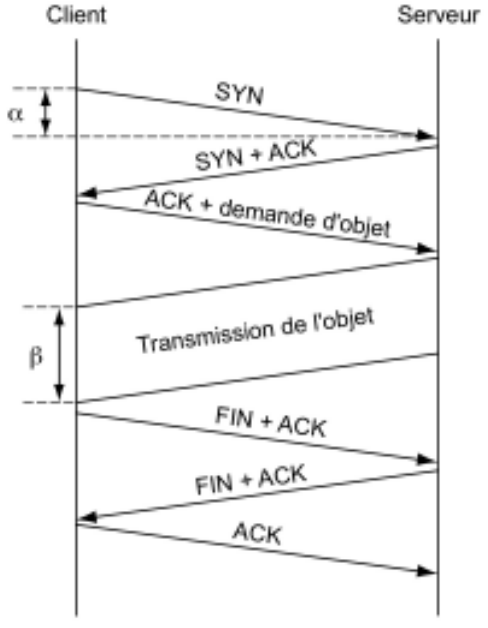


Figure 13.1 Exemple de représentation d'une connexion non persistante

Par défaut, l'approche privilégiée par HTTP version 1.1 vise à conserver la connexion TCP entre le client et le serveur et d'y faire toutes les demandes d'objets au niveau application avec le protocole HTTP. De la sorte, l'appel TCP demeure toujours actif tant et aussi longtemps que le client a des objets à demander au serveur. Cette méthode est désignée par l'expression de «connexion persistante». La figure 13.2 nous donne un exemple de connexion persistante lorsque deux objets sont demandés par le client. Signalons que la version 1.1 du protocole HTTP peut aussi favoriser la méthode de connexion non persistante.

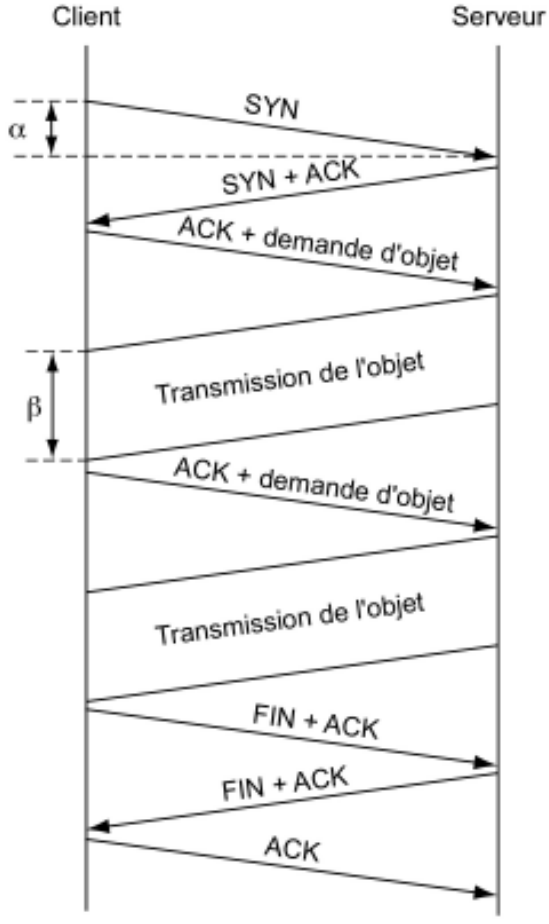


Figure 13.2 Exemple de représentation de connexion persistante avec demande de 2 objets

Nous remarquons que, même dans le cas de la connexion persistante, le serveur est inactif à partir du moment où il a transmit les dernières données sauf s'il reçoit une nouvelle requête du client. Ceci est dû au fait que le client attend

d'avoir reçu toute la transmission de l'objet avant d'adresser une nouvelle demande.

13.2.3 Pipelining

Durant la connexion persistante, la version 1.1 du protocole HTTP permet au client de transmettre, des demandes d'objets sitôt qu'il en a besoin. Ceci génère un grand gain de temps au serveur. Cette particularité de la nouvelle version, qui est d'ailleurs amorcée par défaut, se nomme le « *pipelining* ». La figure 13.3 illustre une connexion persistante avec deux demandes en « *pipelining* ».

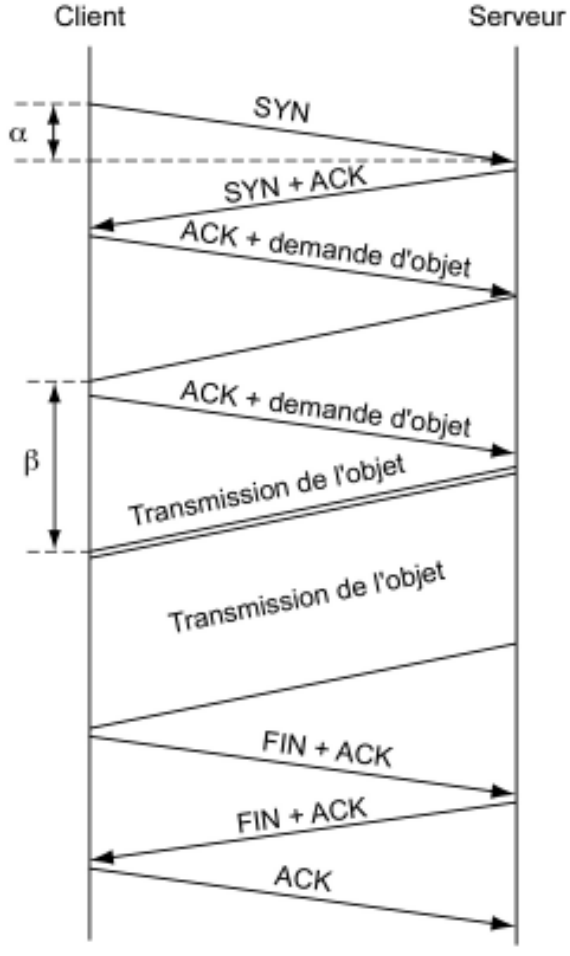


Figure 13.3 Représentation d'une connexion persistante avec 2 demandes en « *pipelining* »

13.2.4 Les messages HTTP

Les messages HTTP échangés entre le client et le serveur sont de deux types :

- *La demande* qui émane du client et est destinée au serveur
- *La réponse* qui correspond à la réaction du serveur à la demande reçue puis transmise du serveur au client.

13.2.4.1 Le message de demande

Le format des messages de demande se trouve dans la section 5 du RFC 2616. Le message de demande comprend une ou plusieurs lignes. La première ligne est la ligne de demande (*Request-Line*) suivie éventuellement de lignes d'en-tête et, selon la demande faite du corps du message (*message body*). Le RFC 2616 précise le message de la demande de façon très concise de la façon suivante :

```
Request  = Request-Line
                    *(( general-header
                      |  request-header
                      |  entity-header  CRLF)
                   CRLF
                   [message-body])
```

CRLF – Carriage return line feed. Retour de chariot

La ligne de demande (*Request-Line*) contient le nom d'une méthode, l'adresse URL de l'objet demandé et la version du protocole HTTP utilisé. Soit et toujours selon le RFC 2616 :

Request-Line = Method Request-URI HTTP-version

Exemple : GET www.etsmtl.ca/index.html HTTP/1.1

Dans le cas de la demande, plusieurs méthodes peuvent prévaloir. La plus courante est le « GET ». Les méthodes sont toutes en majuscules. Celles de la version 1.1 du protocole HTTP sont :

- « OPTIONS » : désigne une demande d'informations relativement aux options disponibles en matière de communications.
- « GET » : désigne tout le contenu de l'objet dont l'adresse est mentionnée.
- « HEAD » : Cette méthode est semblable à celle du « GET » à l'exception près que dans le cadre de cette demande la réponse ne renferme pas le contenu du message (*message-body*).

- « POST » : permet au serveur d'origine d'accepter l'entité incluse dans la demande comme étant la subordonnée de la ressource identifiée par l'URL.
- « PUT » : Cette méthode permet d'emmagasiner l'entité comprise dans la demande et de l'inclure dans l'objet défini par l'URL.
- « DELETE » : Méthode assez dangereuse puisqu'elle donne ordre au serveur d'effacer le contenu de l'objet défini par l'URL.
- « TRACE » : Méthode utile dans le cas de débogage. Le message invoque une application distante et demande de former une boucle et de renvoyer le message reçu.
- « CONNECT » : Cette méthode est utilisée avec un serveur proxy.

L'adresse URL peut adopter différents formats. Elle peut être absolue et comprendre tous les éléments de l'adresse comme dans l'exemple précédent, c'est-à-dire : GET www.etsmtl.ca/index.html HTTP/1.1

D'autre part, l'adresse URL de la ligne de demande pourrait ne contenir que le chemin de l'objet, le reste de l'adresse étant alors mentionné dans le paramètre « Host » de l'en-tête. Notre exemple pourrait aussi être représenté comme suit :

GET /index.html HTTP/1.1
Host: www.etsmtl.ca

L'en-tête général qui pourrait venir après la ligne de demande, permet de compléter la demande en question. Ainsi , dans le cas de la méthode « GET » le paramètre *connection* nous permet de savoir si la connexion TCP doit être persistante ou non. Si le paramètre indique *connection : close*, cela signifie que le client désire mettre fin à la connexion TCP après réception de l'objet. Aussi, nous nous trouvons en présence d'une connexion non persistante. Puisque la version HTTP est 1.1, ce paramètre n'est pas mentionné si une connexion persistante est demandée puisque c'est une option qui vient par défaut. Il existe plusieurs autres paramètres utilisés dans l'en-tête général dont :

Cache-Control
Date
Pragma
Trailer
Transfer-Encoding
Upgrade

Via
Warning

Chacun de ces paramètres est énoncé dans le RFC 2616. Par exemple, le paramètre général *Date* indique le jour et l'heure de l'envoi de la réponse et non celui de la création de l'objet.

L'en-tête de demande suit l'en-tête général. Il existe également une liste de ces paramètres, mais nous n'en indiquerons que quelques-uns.

Un paramètre de l'en-tête de demande utile est le « *User-agent* ». Ce dernier indique le genre de fureteur utilisé permettant ainsi au serveur de transmettre la version appropriée de l'objet si des différences prévalent dans l'interprétation du code HTML des contenus de chacun des fureteurs.

Celle qui présente un intérêt certain en matière d'économie de largeur de bande est le « *If-Modified-Since* ». Un client pourrait optimiser la demande d'objets en ayant recours à ceux qui se trouvent dans l'antémémoire. Ainsi, il pourrait ne demander que ceux qu'il n'a pas déjà stockés ou qu'il demande pour la première fois depuis un certain temps. Toutefois, l'objet pourrait être modifié entre-temps et le client devrait avoir un moyen de le savoir. Aussi, il doit transmettre le message suivant pour tout message qu'il a déjà dans son antémémoire.

```
GET /index.html http/1.1
Host: www.etsmtl.ca
If-modified-since : Fri, 26 Mar 2004 15:27:22
```

Il demande au serveur de lui transmettre l'objet seulement dans le cas où ce dernier a été modifié après la date précisée par le paramètre « *If-modified-since* ».

13.2.4.2 Le message de réponse

Le message de réponse se compose d'une ligne d'état suivie, selon le cas, par des en-têtes et par le contenu du message.

La ligne d'état de la version HTTP ainsi qu'un code d'état sont suivis d'un terme descriptif. Par exemple : HTTP/1.1 200 OK. Cette ligne d'état est transmise par le serveur lorsque la demande est acceptée.

Les lignes d'en-tête sont similaires à celles du message de demande.
Le code d'état englobe plusieurs possibilités. À titre d'exemple, citons les suivantes :

200 OK	Signale que la demande a réussi et ceci est sa réponse.

400 Bad Request	Fait mention de l'échec de la demande
401 Unauthorised	L'accès à la demande est refusé
404 Not Found	Le serveur ne trouve pas l'objet demandé
500 Internal Server Error	Erreur interne du serveur
503 Service unavailable	Service non disponible dans le serveur
505 HTTP Version not supported	Cette version de HTTP n'est pas assurée par le serveur.

Le protocole HTTP permet de mettre en action une application centrale à Internet. Ce protocole est des plus simples. C'est un exemple typique de client-serveur. Il importe de bien dissocier le protocole en question du contenu de l'objet qui compte également son propre protocole et fera état de la présentation de l'information.

L'exemple du message GET et de la réponse telle que détectée à l'aide du programme Ethereal nous donne une idée du comportement du protocole. La trace de la demande pour accéder au site de Microsoft est le suivant :

```
GET /isapi/redir.dll?prd=ie&pver=6&ar=msnhome HTTP/1.1\r\n
   Request Method: GET
Accept: */*\r\n
Accept-Language: fr-ca\r\n
Accept-Encoding: gzip, deflate\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4322)\r\n
Host: www.microsoft.com\r\n
Connection: Keep-Alive\r\n
Cookie:
MC1=GUID=e45da7cb07f2f34e9cb3e31693a503cd&HASH=cba7&LV=20036&V=3;
SITESERVER=ID=e45da7cb07f2f34e9cb3e31693a503cd&HASH=cba7&LV=20036&V=3\r\
n
\r\n
```

La réponse partielle en est :

```
HTTP/1.1 302 Object Moved\r\n
   Content-Length: 141\r\n
   Content-Type: text/html\r\n
   Location: http://www.msn.ca/\r\n

   Server: Microsoft-IIS/6.0\r\n
   P3P: CP="ALL IND DSP COR ADM CONo CUR CUSo IVAo IVDo PSA PSD TAI TELo
OUR SAMo CNT COM INT NAV ONL PHY PRE PUR UNI"\r\n
   X-Powered-By: ASP.NET\r\n
   Date: Thu, 01 Apr 2004 21:31:11 GMT\r\n
```

\r\n
Data (141 bytes)

Parmi les protocoles propres à l'objet, nous pouvons mentionner les fichiers HTML (*HyperText Markup Language*), XML (*Extensible Markup Language*), GIF, JPEG entre autres. Ces protocoles ne font pas l'objet du présent volume.

13.3 Le protocole FTP

Le protocole FTP (*File Transfer Protocol*) est un protocole d'application qui permet de transférer des fichiers ou toute sorte de données. Il est aussi axé sur le concept client-serveur. Le client peut soit transférer des données vers le serveur soit recevoir des fichiers du serveur. Le protocole FTP est traité dans son ensemble dans le RFC 959. C'est l'un des premiers protocoles Internet qui s'avère être d'une grande utilité de nos jours.

Une session FTP est amorcée par deux connexions en parallèle. La première est une connexion de contrôle et est utilisée pour l'échange de commandes et de réponses entre clients et serveur. L'autre permet le transfert des données qui ne peuvent se tenir que si la connexion de contrôle a été établie. Durant le transfert des données, la connexion de contrôle doit être maintenue. Les deux connexions sont établies par le biais du protocole TCP. Ce modèle est illustré à la figure 13.4. La connexion TCP utilisée pour l'établissement de la connexion de contrôle FTP adresse le port 21 du serveur. Le transfert des données utilise le port de destination 20.

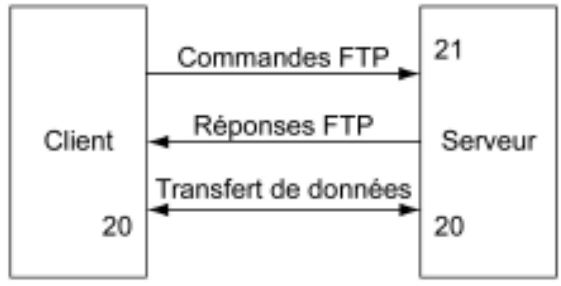


Figure 13.4 Modèle d'utilisation de FTP

Telles que mentionnées dans le RFC 959, les commandes FTP définissent les paramètres qui permettront d'établir la connexion des données ainsi que la nature de l'opération sur le fichier. Les paramètres de connexion sont : le port utilisé dans le transfert des données, le mode de transfert, le type de représentation des données et la structure des fichiers. Les opérations sur fichiers sont définies par les commandes FTP lesquelles permettent, entre autres, de

mémoriser, de transférer, ou de détruire des fichiers. Cet aspect sera traité un peu plus en détail.

Sitôt que le client transmet sa commande FTP, il se met à l'écoute du port TCP qui lui a été assigné attendant que la connexion pour le transfert des données soit amorcée par le serveur.

En l'occurrence, un client pourrait établir une connexion de contrôle avec deux serveurs et s'arranger à ce que ces deux derniers soient connectés. La commande de transfert se fait du client aux serveurs, mais le transfert de fichiers survient entre les serveurs. La figure 13.5 nous donne une illustration de ce cas.

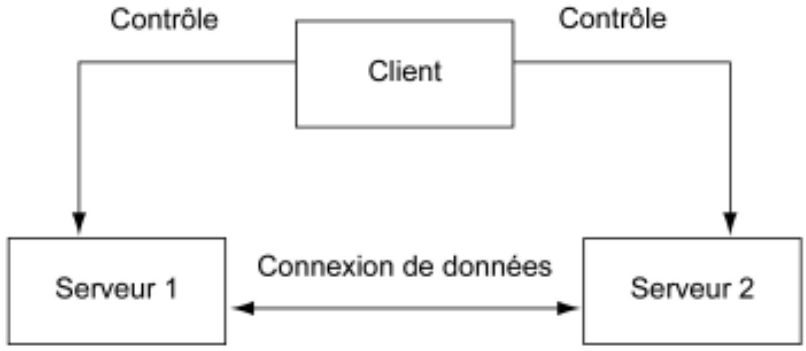


Figure 13.5 Représentation du transfert des données entre deux serveurs

Il importe de noter que la connexion des données s'interrompt dès que le fichier est transmis. Par contre, la connexion de contrôle est maintenue tant et aussi longtemps qu'une commande de déconnexion n'est pas transmise. Cela permet à tout utilisateur de transférer d'autres fichiers durant la même session. Toutefois, et pour chaque nouveau fichier, il faut établir une nouvelle connexion TCP.

Les données sont représentées par deux types précisant le code utilisé pour les représenter. Il s'agit de la représentation en code ASCII, et de celle en mode binaire. Le type ASCII est la représentation employée par défaut.
La structure des fichiers peut aussi être signifiée avec FTP. Il en existe notamment trois types :

- Structure du fichier – la structure par défaut précise qu'il n'y a pas de structure interne. Le fichier est donc considéré comme une suite d'octets.
- Structure de registres (*record*) – Mention à l'effet que le fichier est composé d'un nombre séquentiel de registres.
- Structure de pages – C'est le mode utilisé pour signaler que le fichier est composé de pages à accès aléatoire.

Le mode de transmission peut se faire conformément à trois modes distincts :

- Données formatées
- Données comprimées
- Données inchangées.

Les commandes de contrôles d'accès permettent de savoir qui est l'utilisateur ou le client et de valider son droit d'accès aux fichiers. Nous allons voir l'essentiel de certaines de ces commandes. La première est la commande « USER ». Elle est normalement transmise par le client au serveur sous le format suivant :

USER *nom d'utilisateur*

Elle identifie spécifiquement le client. Le serveur pourrait alors demander au client de lui transmettre son mot de passe. Ceci se fait par le biais de la transmission du client au serveur de la commande suivante :

« PASS » *mot de passe*

Les commandes de service permettent de procéder au transfert de données. Ces commandes sont accompagnées de paramètres précis. Le transfert de données se fait lors de la connexion du transfert des fichiers. Quelques commandes de service sont :

- « RETR » – Cette commande demande au serveur de transférer au client une copie du fichier mentionné dans les paramètres. Cette commande peut être remplacée par la commande GET.
- « STOR » – Le client utilise cette commande pour transférer un fichier vers le serveur. Elle peut être remplacée par la commande « PUT ».
- « APPE » – Cette commande est similaire à « STOR ». Toutefois, la différence est que si le fichier précisé existe déjà, le serveur ajoutera en fin du fichier les données transmises.
- « DELE » – Cette commande permet au client de demander au serveur de détruire un fichier.
- « RMD » – Cette commande ordonne au serveur de détruire le répertoire figurant dans les paramètres.
- « MKD » – Cette commande permet de créer un nouveau répertoire dans le serveur.
- « LIST » – Cette commande permet au client de demander au serveur de lui transmettre la liste des fichiers contenus dans le répertoire courant.
- « HELP » – Cette commande permet d'afficher la liste des commandes FTP.

Étant donné que nous avons à faire à un processus client-serveur, chaque commande FTP du client donne lieu à une réponse du serveur. Comme nous l'avons déjà vu dans le protocole HTTP, la réponse se compose de 3 chiffres suivis d'un message. À titre d'exemple, nous signalons les réponses suivantes :

200 Command OK

331 Username OK, need password

426 Connection closed, transfer aborted

530 Not logged in

13.4 Le protocole TFTP

Le protocole TFTP (*Trivial File Transfer Program*) est une version allégée du FTP. Il est détaillé dans le RFC 1350 et est surtout utilisé lorsque de petits fichiers (d'au maximum 32 mégaoctets) doivent être échangés. Des fichiers de reconfiguration des équipements de réseau sont téléchargés par exemple par le biais du TFTP.

TFTP est une application client-serveur. Chaque paquet transmis conduit à une réponse du serveur. Le TFTP a recours au protocole UDP et est donc limité en matière de fiabilité. Le port utilisé est le 69.

Il existe cinq types de paquets TFTP. Chacun de ces paquets contient son code et son nom. Il s'agit de :

« RRQ »	code=1	demande de lecture de fichier
« WRQ »	code=2	demande d'écriture de fichier
« DAT »	code=3	données du fichier
« ACK »	code=4	accusé de réception
« ERR »	code=5	erreur

Le format des deux premiers paquets est illustré à la figure 13.6. Il comporte un champ de 2 octets réservé au code du paquet. Dans le cas présent, il s'agit de 1 pour « RRQ » ou 2 pour « WRQ ».

octets	2	variable	1	variable	1
	Code	Nom de fichier	0	Mode	0

Figure 13.6 Format du paquet « RRQ » et « WRQ ».

Le nom du fichier est une séquence d'octets en code ASCII suivi d'un octet 0 pour en signaler la fin. Ce mode signale la façon retenue pour procéder au transfert des données. Cela peut être le code ASCII « NETASCII » ou en octets « octet » (pour représenter le mode binaire). Le 0 qui suit sert à signaler la fin du champ mode.

Le paquet de données « DAT » fait appel au format illustré à la figure 13.7. Il comporte le champ Code qui est 3 ainsi que le numéro du bloc transmis suivi des données. Le numéro de bloc commence par 1 pour le premier bloc. La grandeur maximale des données est de 512 octets. Si le paquet data contient moins que 512 octets cela signifie qu'il s'agit du dernier paquet de données du fichier.

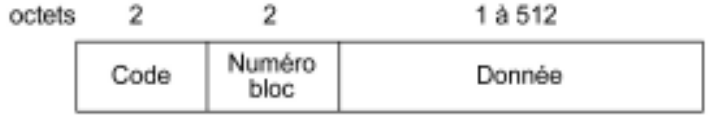


Figure 13.7 Format du paquet « DAT »

Le paquet « DAT » est transmis par le serveur en réponse à un paquet « RRQ ». La moindre erreur de transmission engendre une interruption du transfert des données.

Le paquet « ACK » est illustré à la figure 13.8 et comporte deux champs. Le premier est celui du code qui est mis à 4, et le second indique le numéro du bloc acquitté du dernier paquet « DAT ». Si le paquet « ACK » est transmis en réponse à un paquet « WRQ », le numéro de bloc est mis à 0.

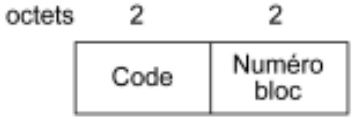


Figure 13.8 Format du paquet « ACK »

Si un accusé de réception n'est pas reçu après un temps déterminé de la transmission d'un paquet « DAT », le client pourrait décider de procéder à la retransmission. Après un certain nombre de retransmissions, il peut interrompre la transmission.

Lorsqu'un client demande la lecture d'un fichier (« RRQ ») il spécifie le port de destination 69 lors de l'accès UDP. Le serveur répond en adressant un paquet « DAT » mais en indiquant un port choisi aléatoirement dans l'accès UDP. Cela

signale au client que ce numéro de port doit être dorénavant utilisé durant la session en cours.

La même situation prévaut lorsque le client demande l'écriture du fichier. Celui-ci utilise le port 69. Le serveur lui indique un nouveau numéro de port lorsqu'il lui transmet le premier accusé de réception.

Le paquet « ERR » fait appel au format illustré à la figure 13.9. Le code du paquet est mis à 5. Le second champ signale le code d'erreur. Le champ suivant est un message en code ASCII qui sert à le décrire. Le 0 signale la fin du message.

octets	2	2	variable	1
	Code	Code d'erreur	Message d'erreur	0

Figure 13.9 Format du paquet d'erreur

Certains types de code d'erreur sont les suivant :

0 Undefined
1 File not found
2 Access violation
3 Disk full
4 Illegal operation

À titre d'exemple, considérons un client qui transmet une demande de lecture de fichiers à un serveur tel qu'illustré à la figure 13.10. Il lui transmet un paquet « RRQ » en indiquant dans le port de destination du segment UDP le numéro 69. Le serveur doit lui transmettre un fichier de 1149 octets. Il lui transmet un premier paquet « DAT » de 512 octets et signale au client que le nouveau port de destination n'est plus le 69 mais le 1234. Le client accuse réception de ce premier bloc de données en utilisant le nouveau port destination. Les autres blocs sont transmis jusqu'au dernier.

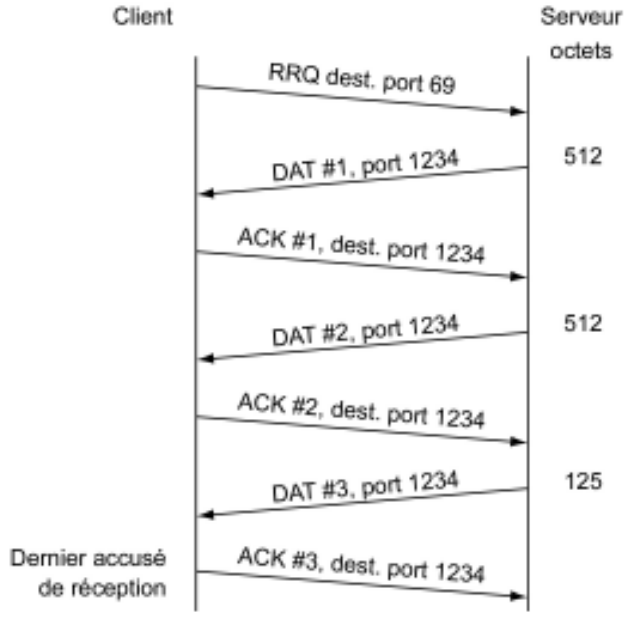


Figure 13.10 Lecture de fichiers avec TFTP

13.5 Le courrier électronique

Le courrier électronique tel qu'appliqué à Internet date de la création du réseau. L'UIT a entrepris également le développement de la messagerie électronique connue sous la série de recommandations X.400.

Le X.400 est un système assez complexe. Conformément au modèle OSI c'est un processus de la couche application. Le développement du X.400 a été le fruit d'une belle entreprise. L'objectif est de couvrir toutes sortes de moyens pour véhiculer les messages et consolider et intégrer ces systèmes de messagerie différents en un seul système.

Toutefois, en raison de la simplicité du courrier électronique connu sous le nom de courriel, ce dernier a été adopté universellement et est devenu le moyen pratique de transmission des messages.

Le courrier électronique est une application typique du concept *Store and Forward*. Les messages sont tout d'abord créés par un agent utilisateur (*User Agent*) qui les transmet à son serveur de messagerie (*Message Server*). Ce dernier les stocke puis les achemine à travers Internet vers le serveur de messagerie du destinataire. Le protocole qui permet aux serveurs de messagerie de communiquer et

d'acheminer les messages à travers le réseau se nomme « SMTP » (*Simple Mail Transfer Protocol*) et figure dans le RFC 2821. La figure 13.11 illustre un système de courrier électronique.

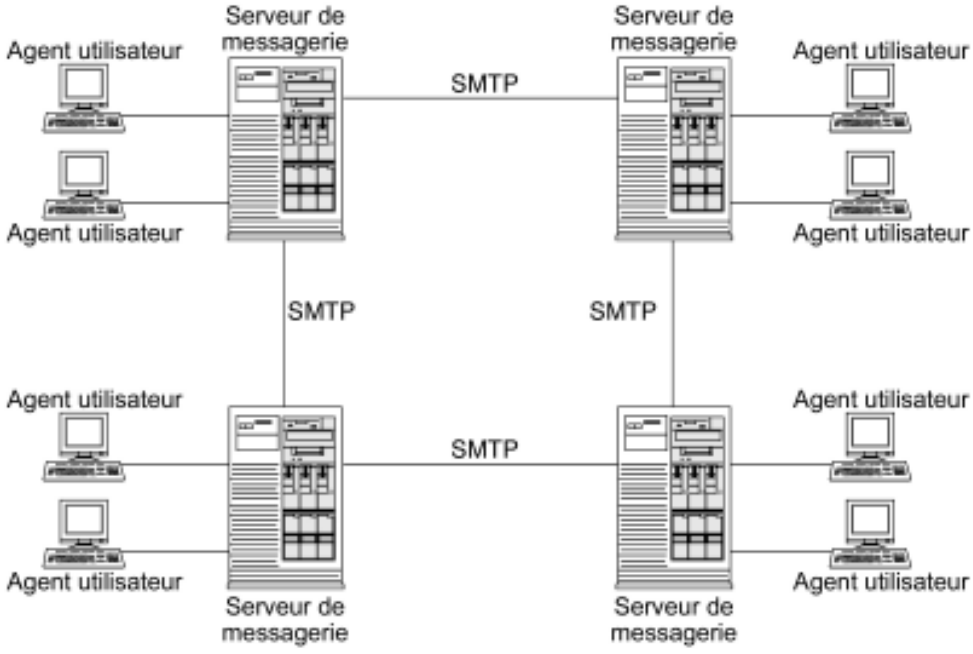


Figure 13.11 Système de courrier électronique « SMTP »

Il importe de mentionner que le protocole « SMTP » n'est utilisé qu'entre les serveurs de messagerie. Il n'est pas souhaitable d'avoir des agents utilisateurs qui soient continuellement connectés aux serveurs d'autant plus qu'il s'agit d'ordinateurs personnels qui ne sont pas toujours allumés ou libres. Les serveurs de messagerie de destination stockent les messages destinés aux agents utilisateurs qu'ils desservent. Chacun de ces agents utilisateurs a une boîte aux lettres se trouvant dans le serveur.

Lorsque l'agent utilisateur désire consulter son courriel, il s'adresse à son serveur de messagerie et se connecte à sa boîte aux lettres. Le protocole utilisé par l'agent utilisateur pour accéder au serveur ne peut pas être le « SMTP ». Il s'agit d'un protocole client-serveur qui est amorcé par l'agent utilisateur selon son désir. Il existe deux protocoles principaux qui permettent ce type de connexion. Il s'agit du protocole « POP3 » (*Post Office Protocol 3*) figurant dans le RFC 1939 et de « IMAP » (*Internet Mail Access Protocol*) se trouvant dans le RFC 2060.

13.5.1 Le protocole SMTP

Le protocole « SMTP » est également un protocole client-serveur. Le serveur de messagerie à la source reçoit un message de l'agent utilisateur et doit le transmettre au serveur destination. Le protocole « SMTP » permet aux deux serveurs se trouvant chacun à une extrémité de converser. Le serveur de messagerie à la source est le client et celui de la destination est le serveur. Le message pourrait transiter par plusieurs serveurs en allant de la source à la destination. « SMTP » ne se préoccupe toutefois que des échanges entre ces deux serveurs d'extrémité. Nous en donnons une illustration simple à la figure 13.12.

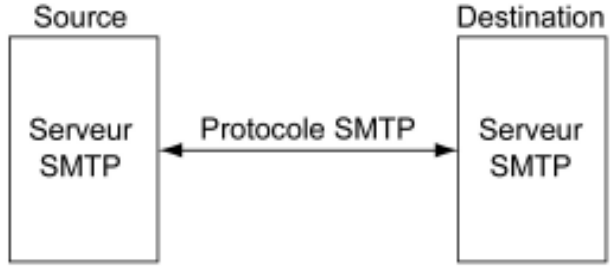


Figure 13.12 Le model SMTP

Le serveur « SMTP » est toujours à l'écoute de toutes les demandes de clients sur le port 25 TCP. Lorsque le serveur reçoit une demande de connexion TCP avec le port de destination 25 il répond au demandeur en adressant le message suivant :

220 Service Ready.

Le client (qui est le serveur de messagerie à la source) transmet alors le message HELO avec le nom de domaine de la source. Par exemple :

HELO etsmtl.ca

Le serveur répond que le message est bien reçu. Plusieurs messages peuvent être transmis à la suite l'un de l'autre. Le code 250 indique qu'un message est positif. Si un trait d'union suit le code, cela signifie qu'un autre message de réponse suit. Le dernier message ne comporte pas de trait d'union. Les messages suivants sont des exemples :

250- bell.ca greets

250 DNS

Le client (ne pas oublier qu'il s'agit du serveur de messagerie de la source) est prêt à transmettre le message en question. En premier lieu, il doit identifier la source, puis il transmet le message suivant :

MAIL FROM :<mkadoch@etsmtl.ca>

Le serveur de destination répond par :

250 mkadoch@etsmtl.ca Sender OK

Le client transmet l'adresse destination :

RCPT TO :<mk@bell.ca>

Le serveur accepte cette adresse et renvoie :

250 mk@bell.ca Recipient OK

Le client pourrait avoir une (ou plusieurs) autres destinations. Par la suite le client avertit le serveur qu'il veut transmettre le message. Il envoie :

DATA

Le serveur répond par le message suivant :

354 Enter mail; end with "." On a line

Le client transmet alors le contenu du message.

Ceci est un message transmis par le protocole « SMTP ».

.

La nouvelle ligne avec seulement un point indique la fin du message. Le serveur accepte le message en renvoyant :

250 Message accepted for delivery

Finalement, le client finit son message en transmettant :

QUIT

Le serveur répond par :

221 bell.ca closing connection

13.5.2 Contenu du message

Outre l'objet du message, son contenu comporte des éléments qui décrivent l'origine et la destination du texte. Ces éléments forment l'en-tête du texte et adoptent le format suivant selon le RFC 822 :

From : mkadoch@etsmtl.ca
To : mk@bell.ca
Subject : test

Ceci est un message transmis par le protocole « SMTP ».
.

L'en-tête ainsi que le message est en ASCII. Lorsque du contenu multimédia fait partie du texte il faut inclure dans l'en-tête une extension. Il s'agit de l'extension MIME (*Multipurpose Internet Mail Extensions*) figurant dans les RFC 2045 et 2046. Ainsi, si notre message comporte une image JPEG codée selon le code base64 tel qu'il figure dans le RFC 2045, le message sera :

From : mkadoch@etsmtl.ca
To : mk@bell.ca
Subject : test
MIME-version :1.0
Content-Transfer-encoding :base64
Content-Type :image/jpeg
(base064 encoded data … *code de l'image* … base064 encoded data)

Il existe plusieurs genres de contenu qui peuvent être présentés incluant le texte et l'image. Les agents « GUI » (*Graphical User Interface*) sont utilisés pour traiter ces interfaces multimédia. Microsoft Outlook en est un exemple.

13.5.3 Le protocole POP3

Le poste de travail de l'utilisateur n'est pas toujours disponible. Le serveur de messagerie doit donc stocker les messages dans la boîte aux lettres de l'abonné. Lorsque ce dernier décide de la consulter, il amorce un agent « GUI » (par exemple Microsoft Outlook) lequel enclenche une session « POP3 » avec le serveur de messagerie. Il peut ainsi en extraire chacun de ses messages.

Le protocole « POP3 » est une application client-serveur. Tout comme dans le cas de « SMTP », le serveur de messagerie est toujours à l'écoute sur le port 110 de toutes les demandes du client. Lorsque le serveur reçoit une demande de connexion TCP avec le port destination 110, le protocole « POP3 » est amorcé et il envoie au client un message d'activation.

Le protocole « POP3 » passe par plusieurs étapes dans son fonctionnement. La première est l'activation telle que nous l'avons vu lors de l'établissement de la connexion TCP. La figure 13.13 nous donne une illustration de cette étape.

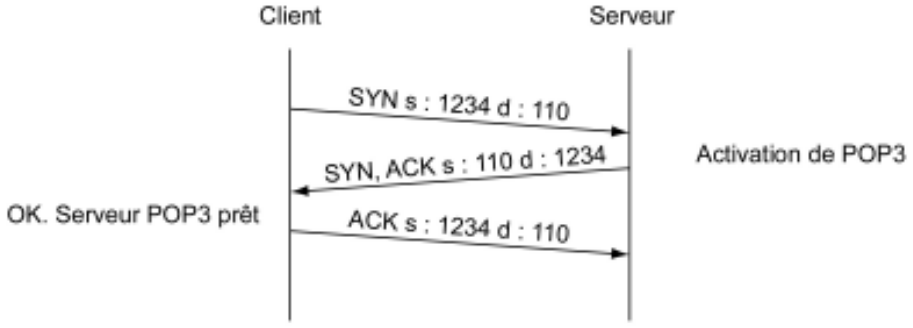


Figure 13.13 Activation de POP3

La seconde étape est celle de l'autorisation où le client doit s'identifier pour que le serveur lui permette d'accéder à sa boîte aux lettres. Les messages échangés sont codés en ASCII. Le client transmet au serveur son nom d'utilisateur en ayant recours à la commande « USER » Le serveur transmet un accusé de réception. Le client transmet alors son mot de passe à travers le message « PASS ». Encore une fois le serveur répond en adressant un accusé de réception. Cet échange est illustré à la figure 13.14.

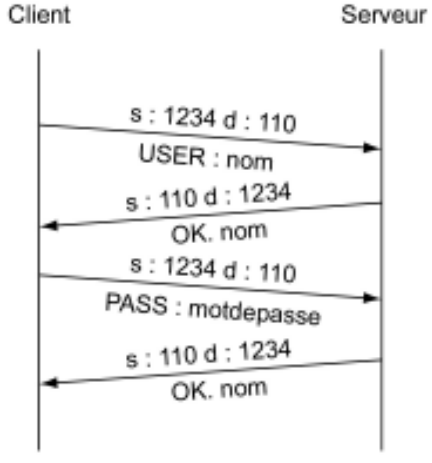


Figure 13.14 Étape d'autorisation

Comme nous pouvons le constater, la sécurité n'est pas très fiable puisque toute personne qui surveille la transaction peut facilement capter le nom d'utilisateur et le mot de passe.

La troisième étape est celle de la transaction. C'est à ce moment que le client peut transmettre des commandes au serveur pour l'interroger sur l'état de sa boîte aux lettres, extraire ses messages ou encore les détruire.

Le client peut par exemple s'informer sur le nombre de messages présents dans sa boîte aux lettres. Il transmet alors le message « LIST ». Le serveur lui renvoie la liste des messages et la longueur de chacun. Il peut décider de récupérer le premier message. Il transmet la commande « RETR 1 ». Le serveur lui renvoie un accusé de réception suivi du contenu du message. Si le client le désire, il peut détruire ce message en transmettant la commande « DELE 1 ». Le serveur lui transmet un accusé de réception pour l'aviser que l'opération a été accomplie. La figure 13.15 illustre l'exemple.

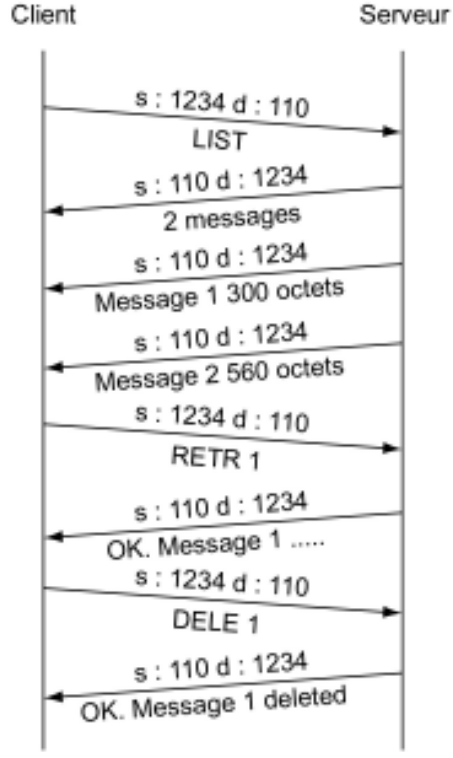


Figure 13.15 Étape de transaction

La quatrième et dernière étape est celle de la fin de l'appel. La connexion TCP est terminée conformément au protocole.

Il importe de noter que, lorsque le protocole « POP3 » passe à l'étape de la transaction, le serveur bloque la boîte aux lettres du client. Tout nouveau message arrivant à ce moment est mis en mémoire dans l'antémémoire et le client n'y a pas accès. Une fois la session terminée, les nouveaux messages sont placés dans la boîte aux lettres et le client doit établir une nouvelle connexion pour y accéder.

Il est essentiel de bien comprendre la différence qui prévaut entre les protocoles « SMTP » et « POP3 ». Le protocole « SMTP » est utilisé pour relayer les messages du serveur source vers celui de destination. C'est un protocole qui doit toujours être activé sans quoi les messages risquent d'être perdus. Par contre, le protocole « POP3 » permet la livraison de messages vers la station de destination. Il n'y a pas de relais de messages; ces derniers partent directement du serveur « POP3 » vers le client qui est la station. Dans le cas de « SMTP », les messages, quoique allant du serveur « SMTP » source au serveur « SMTP » destination, peuvent transiter par le biais d'autres serveurs « SMTP ».

13.6 Exercices

13.1

Quel est le port d'accès par défaut du HTTP, du FTP et du SMTP? Comment peut-on utiliser un autre port que celui par défaut? Donnez-en un exemple.

13.2

Calculez le temps relatif écoulé à partir de l'établissement de l'appel TCP lorsqu'un client demande à partir du protocole HTTP, 10 objets à un serveur dans le cas d'une :

a) « connexion non persistante »
b) « connexion persistante » sans pipelining
c) « connexion persistante » avec pipelining

Comparez les résultats. Donnez la valeur α pour établir le temps qu'un message met pour aller du client au serveur et β le temps de transmission d'un objet. (Note : voir le chapitre 12 sur TCP)

13.3

Un client HTTP demande quatre (4) objets à un serveur. Le client a les deux (2) derniers objets en antémémoire. Ces objets datent de *Mon, 26 Apr 2004 15 :20 :01*. Donnez les messages de demande HTTP que le client transmet ainsi que les messages de réponse qu'il reçoit du serveur en excluant le corps du message; c'est-à-dire pour la demande, signalez la ligne de demande et pour la réponse la ligne d'état ainsi que les en-têtes dans les deux cas.

13.4

Si un serveur doit transmettre un fichier à un client après une demande de lecture de fichier TFTP et que le fichier comprend 1024 octets, quels sont les paquets qui seront échangés?

13.5

Un client a des données longues de 1250 octets à transmettre à un serveur. Il a recours au protocole TFTP. Mentionnez tous les paquets échangés en y détaillant la longueur des blocs, et les ports d'accès.

13.6

La messagerie électronique SMTP procède au transfert de 2 courriels de la source abc@etsmtl.ca à la destination xyz@bell.ca.

a) Expliquez la transition complète des courriels.
b) Décrivez les messages échangés entre le client SMTP et le serveur SMTP.
c) Si la station de destination désire récupérer ses courriels, quels seront les échanges POP3 qui auront lieu entre le client et le serveur POP3?

14 La gestion de réseau SNMP

14.1 Introduction

La gestion de réseau vise à surveiller et contrôler les éléments de réseaux. Ceci est possible à l'aide des informations que ces derniers transmettent. Ces informations décrivent leur comportement. Dans le cas d'une anomalie elle peut être décelée engendrant ainsi une alarme déclenchée par le gestionnaire du réseau. Il incombe à ce dernier d'établir, par son application, les critères qui indiquent s'il y a alarme ainsi que son degré de gravité.

Le gestionnaire de réseau pourrait ne se limiter qu'à dresser un rapport de l'état du réseau, soit en affichant les détails de l'information obtenue ou tout simplement en indiquant par un simple message ou une lumière rouge ou jaune la gravité de l'alarme sans pour autant intenter une action quelconque. D'autre part, le gestionnaire pourrait être astreint à des tâches plus élaborées qui lui permettraient non seulement de déceler des erreurs, mais aussi de mettre en route des tests et d'agir conformément aux résultats obtenus. Cette action pourrait prendre la forme d'une réinitialisation de l'élément de réseau ou encore celle de sa désactivation provoquée par une panne.

Les pannes peuvent provenir de deux types de causes :

- Causes internes : Occasionnées par un composant en panne. Exemple - bris d'une ligne de transmission, dégradation de rendement d'un commutateur. Lorsque le composant est utilisé, le problème de la panne est ressenti.
- Causes externes : Occasionnées par l'environnement du système. Exemple - Congestion de réseaux due à un goulet d'étranglement et causée par un trafic trop intense pour être traitée efficacement autour du goulet. Ce cas est plus difficile à traiter que le premier parce que les facteurs qui interviennent ne sont pas toujours contrôlables.

La panne ne surgit pas soudainement. Pendant une durée donnée, le composant reflète une dégradation de sa performance avant de tomber en panne. Cette durée de dégradation dépend du composant, de son âge, de la durée d'opération et de son historique.

Quatre étapes sont identifiées dans le traitement de la panne :

- La signalisation du fonctionnement anormal (alarmes, performances)
- La localisation de la panne (tests de dépistage)
- La réparation et les tests de bon fonctionnement
- La mise en service et les tests

Il existe plusieurs moyens visant à déceler les problèmes de réseaux entre autres :

- Les alarmes ou messages d'erreur identifiant le composant afin de déterminer le domaine de gestion, le type, la date,…Un compteur reflète les occurrences d'une erreur au cours d'une durée donnée. Citons également le degré de sévérité de l'erreur.
- Les tests qui font état d'une dégradation du composant de réseau. À cet effet, des tests de sécurité et des tests de diagnostic sont prévus.
- Les seuils qui, selon leur niveau, préviennent l'opérateur lorsque ceux-ci ont été franchis.

Afin de réaliser une bonne gestion de réseau, il importe de concevoir un environnement adéquat qui permettrait tant la collecte que le stockage de l'information de gestion. Dans l'attente de développement de normes par l'ISO, le protocole de gestion SNMP (*Simple Network Management Protocol*) a été utilisé. Il fut, en raison de sa simplicité, l'intérim d'un système de gestion plus élaboré. L'UIT conjointement avec l'ISO a conçu et élaboré la série de recommandations X.700 décrivant les fonctionnalités de gestion de réseau par le menu détail. Le protocole SNMP, quoique incomplet par rapport aux normes X.700, répond à l'essentiel des besoins courants de gestion de réseau. C'est la raison pour laquelle il a été adopté *de facto* comme protocole de gestion du réseau Internet ou TCP/IP.

14.2 La gestion du réseau TCP/IP

Le protocole SNMP est une application client-serveur. Dans son contexte, le client est connu sous le nom de *Manager* et le serveur sous le nom de *Agent*. Le *Manager* est le gestionnaire qui demande normalement les informations sur des éléments de réseaux représentés par les *Agents*. Le *Manager* s'adresse donc à

plusieurs *Agents* et centralise les données de gestion reçues de la part de ces derniers. Le *Manager* peut demander une valeur spécifique à l'*Agent*. Ce dernier peut aussi signaler au *Manager* un événement important.

Un autre programme d'application pourrait réaliser l'analyse de ces données et préconiser l'action à prendre.

Le paradigme *Manager-Agent* est une application qui utilise les services de la couche UDP en faisant appel au port de destination 161 pour l'adressage. La figure 14.1 illustre bien ce concept. Par contre, il existe un PDU émanant de l'*Agent* et adressé au *Manager* qui le reçoit par le port 162. Il s'agit du message TRAP dont nous traiterons ultérieurement.

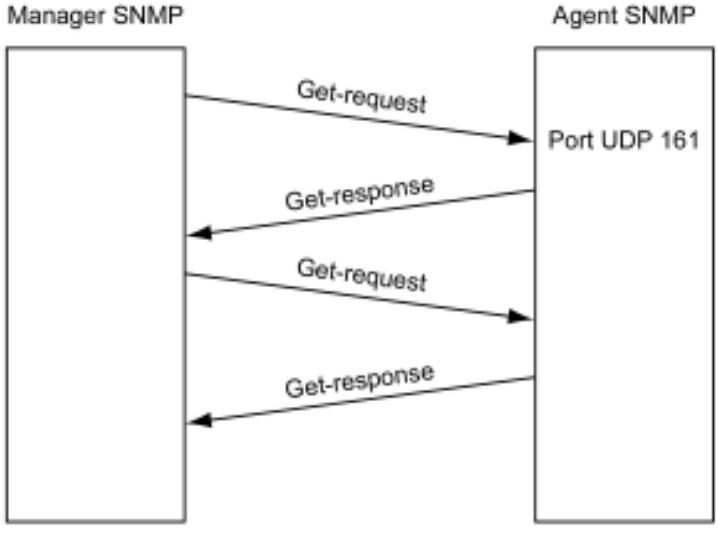


Figure 14.1 Paradigme *Manager-Agent*

La gestion de réseau TCP/IP repose sur trois points, notamment :

- Structure de Gestion d'Information SMI (*Structure Management Information*) qui est un schéma, structure et nomenclature utilisée pour faire référence aux variables dans la MIB (*Management Information Base*) (RFC 1155).
- Une base d'information de gestion MIB (*Management Information Base*) donne les variables soutenues par les éléments de réseau (RFC 1213).
- Le protocole SNMP (*Simple Network Management Protocol*) détaille le format des PDUs ainsi que la procédure utilisée dans la couche application (RFC 1157).

14.2.1 Structure de gestion d'information (SMI)

Le SMI permet d'identifier le type de données utilisées dans les MIBs ainsi que leur structure. Notons que la MIB est l'accumulation structurée d'informations sur les éléments gérés (*managed objects*). Dans le cas de gestion SNMP, les MIBs ne peuvent mémoriser que des données simples scalaires.

Les objectifs de la MIB sont :

- Les objets représentant une ressource particulière doivent être les mêmes pour tous les systèmes.
- Un schéma commun de représentation doit être utilisé pour en assurer l'interopérabilité (fait par le SMI).

Le SMI, tel qu'il figure dans le RFC1155, détaille la façon dont les MIBs doivent être établies et construites. Le SMI aussi normalisé par ISO 10165 nous indique la syntaxe et la sémantique de l'information de gestion emmagasinée dans la MIB. La MIB est utilisée par les *Agents* et les processus de gestion pour sauvegarder ou échanger des informations.

Les types d'informations utilisés par les MIB sont les suivants :

- Éléments de réseau (hubs, ponts, routeurs,…)
- Processus logiciels (programmes, algorithmes, fonctions des protocoles, bases de données) et
- Informations administratives (personne contact, numéro de compte).

Les objets gérés sont établis conformément à une structure arborescente spécifiée telle qu'énoncée dans le modèle OSI. Le MIT (*Management Information Tree*) est formé d'une structure composée de différents niveaux tel que l'illustre la figure 14.2. Au tout début, la structure se compose de la racine (*root*). Le premier palier immédiatement au-dessous de la racine se compose de trois nœuds : iso, itu-t, et iso-itu-t.

La branche *Administration* identifie les objets qui sont détaillés dans les RFCs. Une des sous branches qui en découle est la définition des MIB version 2.

Le numéro figurant dans chacun des nœuds permet d'identifier l'objet de chaque palier. Ainsi et par exemple, l'objet Internet est identifié par le numéro 1.3.6.1. Tous les objets gérés de Internet ont leur identification qui commence par ces quatre chiffres.

Les objets gérés sont précisés par leur type (identificateur et descripteur), leur définition, leur syntaxe de même que par leurs accès et statut. Les composantes ASN.1 sont appliquées afin de préciser cette structure, quoique tous les éléments de ASN.1 ne soient pas utilisés. Notez que ASN.1 est un langage qui permet de définir la structure des données et qui est défini dans les recommandations X.208 et X.680 de l'UIT-T. Toujours selon le RFC 1155, seules les primitives suivantes de ASN.1 sont autorisées d'après la définition qu'en donne le SMI :

INTEGER	entier
OCTET STRING	chaîne d'octet
OBJECT IDENTIFIER	identificateur d'objet
NULL	nul
SEQUENCE	séquence de valeurs du même type
SEQUENCE OF	séquence de valeurs de différents types
CHOICE	choix de valeurs de types

Outre ces éléments, le RFC 1155 définit de nouveaux types utilisés dans des applications. Quelques-uns sont énoncés comme suit :

NetworkAddress – Choix d'adresses parmi plusieurs familles de protocoles.
IpAddress – Adresse IP définie par la primitive OCTET STRING d'une longueur de 4 (32 bits).
Counter – Ceci représente un compteur avec des valeurs non négatives d'entiers (INTEGER) d'une longueur de 32 bits et incrémentés modulo 2^{32} –1.
Gauge – Manomètre doté de valeurs non négatives d'entiers oscillants de 0 à 2^{32}–1.
TimeTicks – Chronomètre ayant des valeurs non négatives d'entiers et dont le temps s'écoule en centièmes de seconde.

Le RFC1155 indique également le format utilisé pour déterminer les objets gérés. Un type d'objet comprend cinq paramètres :

OBJECT : Un nom pour le type d'objet (OBJECT DESCRIPTOR) avec l'OBJECT IDENTIFIER correspondant.
Syntax – Correspond à la syntaxe du type d'objet en ASN.1. Cela pourrait être un INTEGER ou un OCTET STRING.
Definition – Une brève description du type d'objet.
Access – Établit le type d'accès pour l'objet, soit lecture seulement (*read only*), lecture et écriture (*read-write*), écriture seulement (*write only*), ou non accessible (*not-accessible*).
Status – Le statut de l'objet peut être obligatoire (*mandatory*), optionnel (*optional*) ou désuet (*obsolete*).

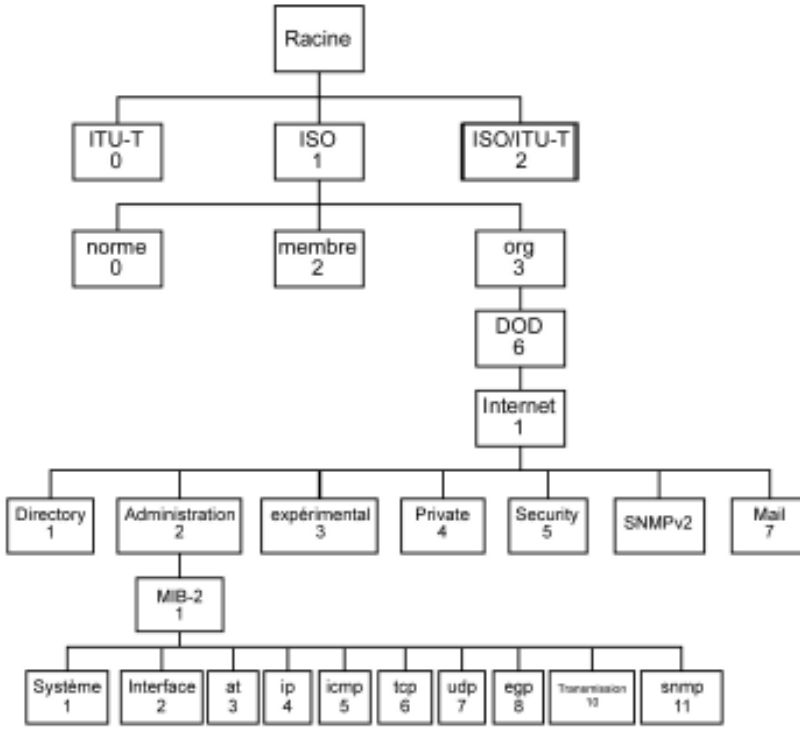


Figure 14.2 L'arbre d'information de gestion (MIT)

La figure 14.3 illustre l'arborescence de UDP. L'identificateur d'objet (OBJECT IDENTIFIER) d'une variable simple se termine toujours par un .0. Par exemple, UpInDatagrams est identifiée par 1.3.6.1.2.1.7.1.0 et non par 1.3.6.1.2.1.7.1. Le libellé de cette référence est : *iso.org.dod.internet.mgmt.mib.udp.udpInDatagrams.0*

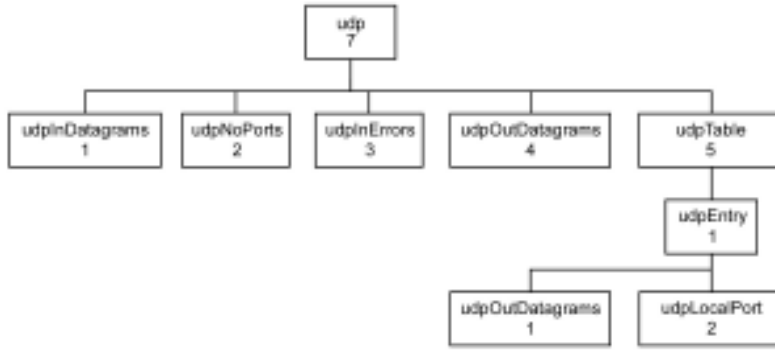


Figure 14.3 Identificateur d'objet

Le type d'objet udpInDatagrams tel que stipulé dans le RFC 1156 est représenté conformément à la syntaxe établie précédemment.

OBJECT :
udpInDatagrams (udp 1)

Syntax :
Counter
Definition :
The total number of UDP datagrams delivered to UDP users.
Access:
read-only.
Status:
mandatory.

Mentionnons que l'identificateur d'objet (OBJECT IDENTIFER) pour udpInDatagram est udp 1 ce qui représente 1.3.6.1.2.1.7 pour udp auquel 1 est ajouté. Puisque c'est une variable simple complète, on ajoute un zéro pour donner ce que nous avons décrit précédemment, soit : 1.3.6.1.2.1.7.1.0.

Cet objet est adapté au groupe UDP défini dans MIBv2 et est obligatoire pour tous les systèmes qui fonctionnent avec le protocole UDP.

Afin de représenter les objets définis dans les MIB de façon plus formelle, on utilise les macros pour les objets gérés. Notre définition générique des objets revêt alors la forme suivante :

```
OBJECT-TYPE MACRO :: =
BEGIN
        TYPE NOTATION :: = "SYNTAX" type (TYPE objectSyntax)
                           "ACCESS" Access
                           "STATUS" Status
        VALUE NOTATION :: = value (VALUE ObjectName)

        Access :: = "read-only"
                  | "read-write"
                  | "write-only"
                  | "not-accessible"
        Status :: = "mandatory"
                  | "optional"
                  | "obsolete"
END
```

Notre exemple sur la définition de l'objet udpInDatagrams sera représenté de la façon suivante :

```
udpInDatagrams OBJECT-TYPE
            SYNTAX    Counter
            ACCESS    read-only
            STATUS    mandatory
            :: = { udp 1}
```

14.2.2 MIB-II

Les MIBs figurant dans le RFC 1156 ont été accrues par l'apport de nouveaux objets et groupes indiqués dans les MIB-II (MIBv2) du RFC 1213. Tel qu'illustré à la figure 14.2 les MIB-II comprennent 10 groupes :

system – Inclut les objets en rapport avec le système. Notamment :

sysDescr	DisplayString	Description de l'entité
sysObject	ObjectID	Id. du vendeur (dans 1.3.6.1.4.1)
sysUpTime	TimeTicks	Temps depuis le redémarrage- 100ème de sec
sysContact	DisplayString	Nom du contact
sysName	DisplayString	Noeud du nom de domaine
sysLocation	DisplayString	Location physique du noeud
sysServices	[0..127]	Services fournis par le nœud : 1(physique), 2(liaison).

interface – Permet d'obtenir les informations relatives aux interfaces du système et de ses sous-systèmes. Le groupe interface n'a qu'une variable simple *ifNumber* indiquant le nombre d'interfaces réseau se trouvant sur le système et une table *ifTable* donnant 22 variables qui établissent les caractéristiques propres à chaque interface.

at – Pour *address translation*. Information sur la table de traduction d'adresses. Le groupe *at* définit une table qui sur 3 colonnes nous donne : le numéro d'interface (*atIfIndex*), l'adresse physique (*atPhysAddress*) et l'adresse IP (*atNetAddress*).

ip – Information en rapport avec la mise en vigueur du protocole IP. Le groupe ip comporte de nombreuses variables groupées dans trois tables. Le comportement du protocole, la table d'adresses IP ainsi que la table de routage et la table de traduction d'adresse IP peuvent être alors obtenus.

icmp – Information icmp sur les messages transmis ou reçus. Le groupe icmp se compose de quatre compteurs généraux ainsi que de 22 compteurs pour les différents types de messages ICMP (11 compteurs d'entrée et 11 compteurs de sortie).

tcp – Information sur les échanges TCP si ce protocole est mis en vigueur dans le système. Le groupe tcp ne compte qu'une seule table; la table de connexion TCP, qui contient une ligne par connexion pour indiquer l'état de la connexion, ainsi que les adresses IP et les ports locaux et distants.

udp – Information sur les échanges UDP si ce protocole est mis en vigueur dans le système. Ce groupe se compose également d'une table donnant des informations sur l'entité d'applications qui reçoit les datagrammes.

egp – Information sur les échanges EGP si ce protocole est mis en vigueur dans le système. La table comprise dans ce groupe offre les informations sur les passerelles voisines connues de cette entité.

transmission – Information sur la méthode de transmission et sur le protocole d'accès de chaque interface du système, par exemple Ethernet.

snmp – Information sur le protocole SNMP.

Chacun de ces groupes est doté de plusieurs objets simples et parfois de tables. Grâce au protocole SNMP le *Manager* peut obtenir les instances de ces objets. Celles-ci sont les valeurs stockées dans les objets contrairement au type qui lui, ne nous en donne que la structure. Quelques exemples pris dans la description du protocole SNMP nous permettront de mieux le comprendre.

14.2.3 Le protocole SNMP

Le protocole SNMP est utilisé pour obtenir les valeurs d'objets de MIBs, notamment les instances de ces objets. Il peut aussi être utilisé pour modifier ces valeurs lorsque l'ACCESS des objets le permet. Ceci se produit grâce aux commandes transmises par le *Manager* sous forme de demandes, et de réponses de l'*Agent* contenant les informations demandées ou la confirmation du changement effectué. Si l'*Agent* se trouve dans l'impossibilité d'acquiescer à la demande du *Manager* il adresse alors un message d'erreur.

Il existe une autre commande qui ne requiert pas de réponse et qui est transmise par l'*Agent* au *Manager*. Il s'agit de la commande TRAP. Cette commande permet à l'*Agent* de signaler un événement important en transmettant l'instance de l'objet

représentant cet événement. Cette commande évite la transmission continue de demande de la part du *Manager* et évite ainsi de congestionner la ligne à des moments où le réseau pourrait connaître un comportement critique.

Il existe trois versions du protocole SNMP. Le fonctionnement de la première est particulièrement simple, mais elle est aussi très limitée. Il importe de bien en saisir son fonctionnement avant de se pencher sur les deux autres. Les messages SNMP de la version 1 sont les suivants :

GetRequest, GetNextRequest, SetRequest, GetResponse, et Trap.

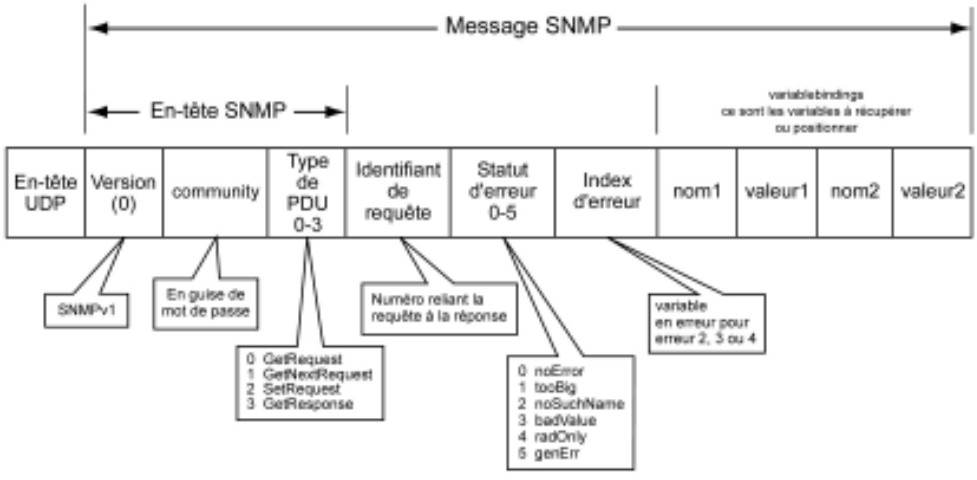


Figure 14.4 Format de messages SNMP

Les quatre premiers types de message font appel au format tel que représenté à la figure 14.4 alors que le message Trap a recours au format illustré à la figure 14.5.

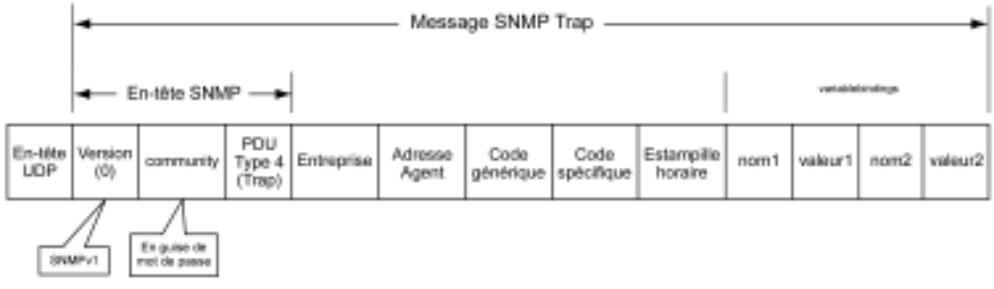


Figure 14.5 Format du message TRAP du protocole SNMP

Les messages GetRequest, GetNextRequest et SetRequest ont toujours les champs *Statut d'erreur* et *Index d'erreur* mis à 0. Alors que le message GetResponse a pour sa part recours à ces champs pour signaler une erreur si besoin est.

Le message est vérifié par l'*Agent* pour la version SNMP (0 pour version 1) et pour le droit d'accès à l'aide des adresses source et destination ainsi que du champ *community*. L'*Agent* authentifie l'accès du service en s'assurant que le *Manager* a les autorisations requises et que la valeur du champ *community* est conforme à l'attente de l'*Agent*. Ce champ sert, à toutes fins pratiques, de mot de passe.

Le champ *variablebindings* se compose d'une série de références faites à des instances d'objets et dans le cas du message GetResponse, également à leur valeur.

14.2.3.1 Le message GetRequest

Le message GetRequest est toujours transmis par le *Manager* et adressé à un *Agent*. Le champ type de PDU est à 0 pour indiquer que c'est un message GetRequest. Le champ *variablebindings* inclut la liste des instances d'objets dont les valeurs sont demandées par le *Manager* à l'*Agent*.

L'*Agent* qui reçoit ce message répond par le message GetResponse. Il importe que le champ *Identifiant de requête* ait le même numéro tant dans les messages GetRequest que GetResponse correspondant. L'*Agent* renvoie dans le champ *variablebindings* les valeurs des instances d'objets demandées par le GetRequest. Si toutefois une seule valeur parmi toutes celles demandées venait à être erronée, l'*Agent* ne renvoie aucune valeur et signale une erreur dans le message GetResponse. Cette indication d'erreur ne donne aucune indication sur la nature du problème.

Pour illustrer les messages échangés, prenons, à titre d'exemple, le groupe UDP. Les ramifications de ce groupe sont indiquées à la figure 14.6. La teneur est identique au contenu de la figure 14.3.

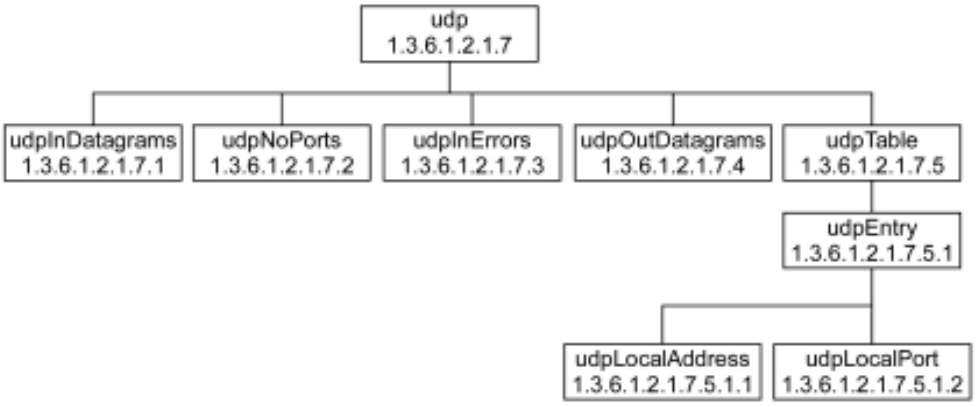


Figure 14.6 Groupe UDP

Nous remarquons que le groupe se compose de quatre objets et d'une table qui, elle, se compose de deux objets indexés. Les objets indexés peuvent avoir plusieurs entrées.

Tel qu'illustré à la figure 14.7, un message GetRequest peut inclure dans son champ *variablebindings* les instances des objets suivants :

udpInDatagrams.0, udpOutDatagrams.0

Le message GetResponse peut alors retourner les valeurs suivantes s'il n'y a pas d'erreurs :

udpInDatagrams.0=25, udpOutDatagrams.0=27

Si l'une des deux valeurs est erronée, le message GetResponse ne retourne aucune valeur et signale une erreur.

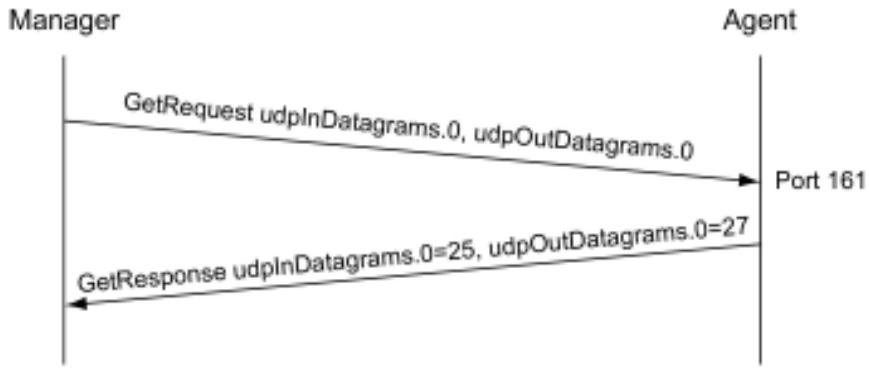


Figure 14.7 Demande d'information avec GetRequest

Notons que les instances d'objets étant simples, elles finissent par un 0. Seules les extrémités de l'arbre de groupe représentant les requêtes d'objets peuvent être utilisées dans le champ *variablebindings* du message GetRequest.

14.2.3.2 Le message GetNextRequest

Le message GetNextRequest ressemble de beaucoup au message GetRequest. Toutefois, la différence réside dans la spécification des instances d'objets demandées dans le champ *variablebindings*.

Le message GetRequest indique exactement les instances des objets désirées dans le champ *variablebindings*. Dans le cas du message GetNextRequest, l'instance d'objet demandée est l'entrée lexicographique suivante du contenu du champ *variablebindings*.

Le groupe UDP nous servira d'exemple pour mieux comprendre l'ordonnancement lexicographique. La lecture des instances d'objets dans la MIB se fait de manière et dans un ordre qualifié de lexicographique.

Supposons que les valeurs d'instances d'objets du groupe UDP pour un élément de réseau sont :

udpInDatagrams=25,
udpNoPorts=80,
udpInErrors=4,
udpOutDatagrams=27

Supposons également que la table du groupe UDP comprenne trois entrées dont les valeurs sont les suivantes :

udpLocalAddress	udpLocalPort
198.168.100.10	10
198.168.100.11	42
198.168.100.12	25

Lors de la lecture de l'arborescence du groupe UDP cela se traduit de la façon suivante. L'ordre lexicographique de tous les éléments de cet arbre est :

udpInDatagrams.0,
udpNoPorts.0,
udpInErrors.0,
udpOutDatagrams.0,
udpLocalAddress.198.168.100.10
udpLocalAddress.198.168.100.11
udpLocalAddress.198.168.100.12
udpLocalPort.198.168.100.10
udpLocalPort.198.168.100.11
udpLocalPort.198.168.100.12

Il faut toutefois être prudent et savoir que les valeurs simples se terminent par un 0. Cela nous indique que la valeur suivante de udpInDatagrams est udpInDatagrams.0=25. Par contre, la valeur qui suit udpInDatagrams.0 est udpNoPorts.0=80.

Lorsque la table udpTable est abordée par le message GetNextRequest, on obtient la première entrée en se référant aux objets suivants figurant dans le champ *variablebindings* : udpLocalAdress, et udpLocalPort.

Le message GetResponse retourne les valeurs suivantes :

udpLocalAddress.198.168.100.10=198.168.100.10,
udpLocalPort.198.168.100.10=10

Pour obtenir les entrées suivantes, le message GetNextRequest doit placer dans le champ *variablebindings* les paramètres suivants :

udpLocalAddress.198.168.100.10, udpLocalPort.198.168.100.10

Le message GetResponse renvoie les valeurs suivantes :

udpLocalAddress.198.168.100.11=198.168.100.11,
udpLocalPort.198.168.100.11=42

Ce dernier échange de messages entre le *Manager* et l'*Agent* est illustré à la figure 14.8.

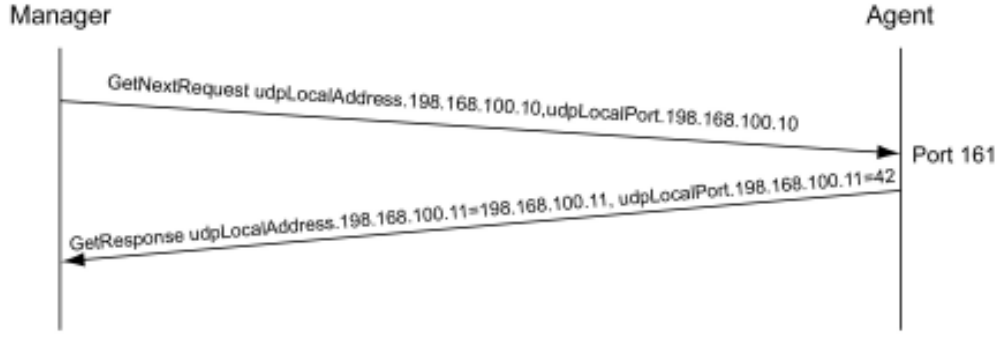


Figure 14.8 Demande d'information avec GetNextRequest

Le *Manager* peut ainsi obtenir tout le contenu de la table. Il n'en connaît pas la taille. Il continue donc à transmettre des messages GetNextRequest jusqu'à ce qu'il reçoive des paramètres différents de ceux de la table ou bien un message d'erreur. De la sorte, il pourra établir la fin de la table.

14.2.3.3 Le message SetRequest

Le message SetRequest est transmis par le *Manager* pour modifier la valeur d'une instance d'un objet. Le champ *variablebindings* comporte donc les valeurs qui doivent être transmises au MIB. Comme dans le cas de GetRequest et de GetNextRequest, plusieurs paramètres peuvent être transmis. Toutefois, si un seul d'entre eux génère une erreur aucun de ces paramètres ne sera mis à jour.

Les opérations de mise à jour ne peuvent se faire que si l'ACCESS de l'objet se trouve en mode *Write*. (*Écriture*)

Il est possible d'effacer une ligne dans une table avec cette commande en attribuant la valeur « *invalid* » à l'instance de l'objet. Une réponse GetResponse est toujours transmise avec les mêmes valeurs.

14.2.3.4 Le message Trap

Le seul message initié par l'*Agent* et envoyé au *Manager* est le message *Trap*. Celui-ci indique que quelque chose vient de se produire chez l'*Agent*. Les messages *Traps* sont envoyés au port destination UDP 162. Il n'y a pas de réponse à ce message. La figure 14.5 illustre le format du message.

Le champ *entreprise* dans le message *Trap* identifie le système de gestion qui amorce le message. Le champ *adresse agent* indique l'adresse IP de l'objet qui génère le message. Le champ *code générique* établit le type de *Trap*. Le champ *code spécifique* indique d'une manière plus précise la nature de la *Trap*. Le champ *estampille horaire* indique le temps écoulé dès le moment où l'entité transmettant le message a été initialisée jusqu'à celui où le message *Trap* à été transmis. Certaines autres informations figurent également dans le champ *variablebindings*.

Six types de *Trap* sont définis. Un septième peut être utilisé par le vendeur. Les *Traps* suivantes sont codées dans le champ *code générique* :

Cold start – code 0 –Indication d'une réinitialisation causée par une erreur majeure.

Warm start – code 1 –Réinitialisation n'affectant pas le protocole SNMP.

Link down – code 2 – Indication qu'un lien est en panne. Ce lien figure dans le champ *variablebindings*.

Link up – code 3 – La panne de lien est réparée.

Authentication failure – code 4 – l'authentification qui à lieu avec les autres messages SNMP n'a pas réussi. Ce *Trap* nous en donne l'indication.

Egp neighbor loss – code 5 – Indication qu'un lien externe est en panne.

Le septième message *Trap* est défini par l'entreprise qui a conçu l'entité en question. Il s'agit de *Entreprise specific – code 6*.

14.2.4 Agent mandataire (Proxy)

Certains systèmes, en raison de leur taille, ne soutiennent pas les normes de gestion de réseau. Ils ont leur propre méthode pour faire état des fautes. À titre d'exemple, nous prenons une banque de modems ou celle des multiplexeurs qui

ne soutiennent pas les protocoles TCP/IP ou SNMP. Par contre, il faut permettre au gestionnaire de réseau d'exercer un contrôle sur ces équipements.

Une solution pratiquée est le recours à un Agent mandataire (Proxy). Cet Agent représente les équipements auxquels le *Manager* i ne peut pas avoir directement accès. Ce dernier passe alors par l'Agent mandataire pour obtenir indirectement des informations de gestion. L'Agent mandataire traduit la demande du *Manager* par une commande précise de l'équipement visé et il la lui transmet. Les réponses reçues de l'équipement sont également traduites et relayées par l'Agent mandataire au *Manager*.

14.2.5 SNMPv2

La version 2 de SNMP ne remplace pas SNMPv1 mais y apporte des améliorations. Celles-ci sont surtout apparentes dans la structure SMI ainsi que dans les nouvelles opérations qui ont été ajoutées. Dans cette seconde version, il est possible aussi d'établir une communication entre deux *Manager* en vue d'échanger des informations.

Le SMI appliqué au SNMPv2, détaillé dans le RFC 1902, étend la macro qui établit le type d'objet. Selon le RFC 1212 la nouvelle macro prend la forme ASN.1 suivante et est compatible avec celle détaillée dans le RFC 1155 en regard de la première version de SNMP :

```
OBJECT-TYPE MACRO ::=
BEGIN
        TYPE NOTATION ::=
                                "SYNTAX" TYPE(ObjectsSyntax)
                                "ACCESS" Access
                                "STATUS" Status
                                Descrpart
                                ReferPart
                                IndexPart
                                DefValPart
        VALUE NOTATION ::=
                        value   (VALUE ObjectName)

        Access ::= "read-only"
                        | "read-write"
                        | "write-only"
                        | "not-accessible"
```

```
Status :: = "mandatory"
                    | "optional"
                    | "obsolete"
                    |   "deprecated"

DescrPart ::=
            "DESCRIPTION" value
            | empty

ReferPart ::=
            "REFERENCE" value
            | empty

IndexPart ::=
            "INDEX" "(" IndexTypes ")"
            | empty

IndexTypes ::=
            IndexTypes | IndexTypes "," IndexType
IndexType ::=
            Value (indexobject ObjectName)
                    | type (indextype)
DefValPart ::=
"DEFVAL" "(" value (defvalue ObjectSyntax) ")"

END
```

D'autres définitions de module et de notification se trouvent dans le RFC 1902 ainsi que de nouveaux termes qui ont été rajoutés au SMI de SNMPv1.

Les MIBs en rapport avec le SNMPv2 figurent dans le RFC 1907.

14.2.5.1 Les messages SNMPv2

Le protocole SNMPv2 comprend les messages dont la définition se retrouve également dans le RFC 1905. Il importe de savoir que le format des messages de SNMPv2 GetRequest, GetNextRequest, SetRequest ainsi que le format de SNMPv2 Response, sont similaires aux messages correspondants de SNMPv1.

De nouveaux messages y sont ajoutés dont : GetBulkRequest, InformRequest, Report et Trap de SNMPv2. Le message Trap de SNMPv1 n'est plus utilisé. Le GetResponse de SNMPv1 a été renommé Response dans SNMPv2.

Les messages GetRequest et GetNextRequest de SNMPv2 conservent le même format que ceux de SNMPv1. La différence des versions se fait sentir dans la réponse. Dans le cas de SNMPv1 nous avons vu que si seul un des paramètres demandés est erroné le message de réponse transmet une erreur globale et qu'aucune valeur n'est transmise. Par contre, avec SNMPv2 la réponse renferme les valeurs des paramètres qui ne sont pas erronés.

Le message SetResponse de SNMPv2 est, à toute fin pratique, identique à celui de SNMPv1 en ce qui a trait au format et à la sémantique.

Un nouveau message est ajouté à SNMPv2. Il s'agit de GetBulkRequest. Un message GetBulkRequest est l'équivalent de plusieurs GetNextRequest. Il a été conçu et élaboré pour réduire le nombre d'échanges lors de l'obtention d'un nombre d'informations important.

Le message GetBulkRequest dont le format est illustré à la figure 14.9 inclut les champs *non-repeaters* et *max-repetitions*. Parmi les paramètres définis dans le champ *variablebindings* certains pourraient n'être simplement rapportés qu'une seule fois tout comme avec le message GetNextRequest alors que d'autres pourraient être répétés plusieurs fois en se conformant à l'ordre lexicographique. Le champ *non-repeaters* indique le nombre des premiers paramètres répertoriés dans le champ *variablebindings* paramètres qui ne doivent pas être répétés, mais transmis une seule fois. Le champ *max-repetitions*, indique par contre, le nombre de répétitions selon l'ordre lexicographique qui doit être effectué relativement aux paramètres restants.

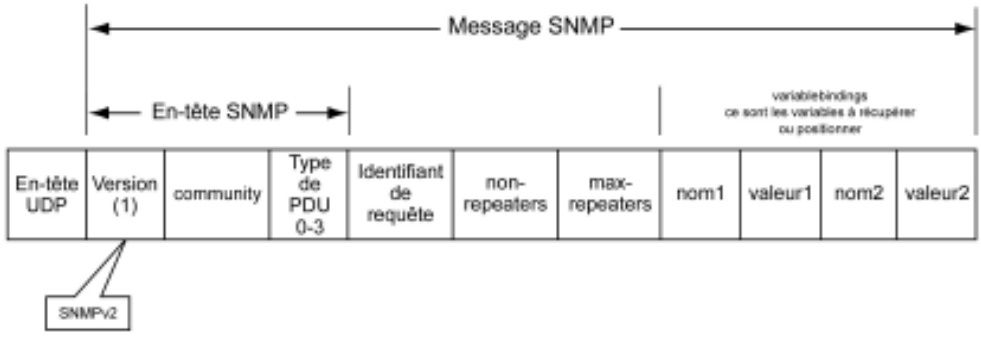


Figure 14.9 Message GetBulkRequest

Prenons l'exemple utilisé pour la description des messages GetRequest et GetNextRequest. Il s'agit du groupe UDP avec les valeurs suivantes assignées aux instances des objets du groupe.

udpInDatagrams=25,
udpNoPorts=80,
udpInErrors=4,
udpOutDatagrams=27

La table du groupe UDP comprend 3 entrées avec les valeurs suivantes :

udpLocalAddress	udpLocalPort
198.168.100.10	10
198.168.100.11	42
198.168.100.12	25

Le message GetBulkRequest, avec les paramètres suivants dans leur champ correspondant, est transmis à l'*Agent* :

non-repeaters = 1
max-repetitions = 2
variablebindings : udpInDatagrams, udpLocalAddress, udpLocalPort

La valeur de udpInDatagrams ne sera retournée qu'une seule fois alors que la valeur de udpLocalAddress et udpLocalPort sera retournée deux fois pour chacun selon l'ordre lexicographique. La réponse sera donc :

Message : Response

Valeur dans le champ *variablebindings* :

udpInDatagrams=25,

udpLocalAddress.198.168.100.10=198.168.100.10,
udpLocalPort.198.168.100.10=10

udpLocalAddress.198.168.100.11=198.168.100.11,
udpLocalPort.198.168.100.11=42

Un seul message et une seule réponse ont été essentiels pour l'obtention de ces informations. Le même message avec une valeur de *max-repetitions* plus grande pourrait être transmis en vue d'obtenir tout le contenu de la table udpTable par une seule réponse.

Le message SNMPv2 Trap conserve le même rôle que son prédécesseur de la version SNMPv1. Seul le format change et est maintenant identique à celui utilisé par les autres messages à l'exclusion de GetBulkRequest.

Un autre message, nouveau celui-là a été ajouté à SNMPv2. Il s'agit de InformRequest. Ce message est transmis d'un *Manager* à un autre, lui demandant les valeurs d'instances des objets répertoriés dans le champ *variablebindings*. Le *Manager* qui reçoit la demande adresse une réponse avec les valeurs demandées.

Il existe une troisième version de SNMP, SNMPv3 ainsi que le SMIv2. Cette version vient compléter la version 2 du protocole SNMP. Cette version comprend des aspects portant sur la sécurité. Quatre algorithmes cryptographiques figurent dans le fonctionnement de cette version.

14.3 Exercices

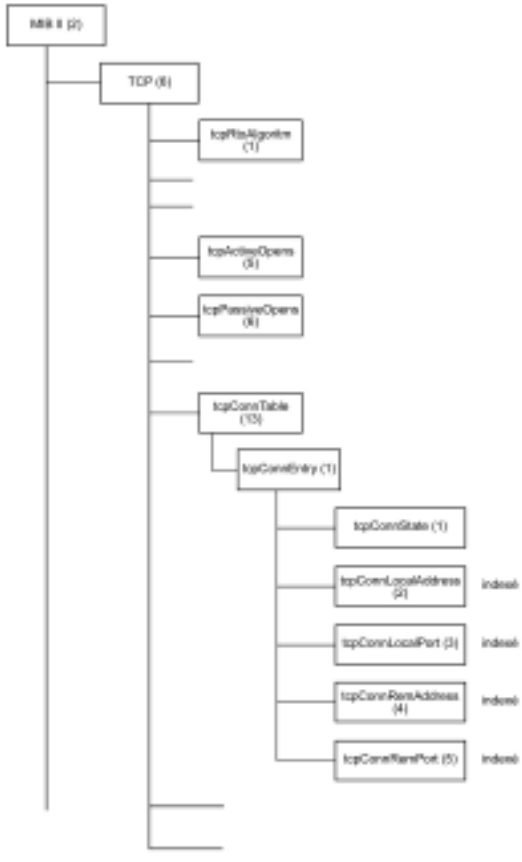
14.1

Indiquez l'identificateur d'objet (*OBJECT IDENTIFIER*) pour les objets du groupe TCP. Vous devez retrouver le RFC définissant le groupe puis l'arbre d'informations pour terminer l'exercice.

14.2

En utilisant le message GetRequest, indiquez tous les paramètres auxquels on doit avoir recours pour obtenir les valeurs des objets tcpActiveOpens, tcpPassiveOpens, et deux instances de valeurs de tous les éléments de la table tcpConnTable. Envisagez des valeurs pour les réponses.

Le groupe MIB de tcp est en partie le suivant :



14.3

En utilisant le message GetNextRequest, indiquez tous les paramètres auxquels on doit avoir recours pour obtenir les valeurs des objets tcpActiveOpens, tcpPassiveOpens, et deux instances de valeurs de tous les éléments de la table tcpConnTable. Envisagez des valeurs pour les réponses.
Le groupe MIB de tcp se trouve illustré à l'exemple 14.2.

14.4

En utilisant le message GetBulkRequest, indiquez tous les paramètres auxquels on doit avoir recours pour obtenir les valeurs des objets tcpActiveOpens, tcpPassiveOpens, et deux instances de valeurs de tous les éléments de la table tcpConnTable. Envisagez des valeurs pour les réponses.
Le groupe MIB de tcp se trouve illustré à l'exemple 14.2.

ACRONYMES

A

ABR	Area Border Router
AC	Access Control
ACFC	Address and Control Field Compression
ADSL	Asymmetric digital subscriber line
AID	Association Identity
AMP	Active Monitor Present
ANSI	American National Standard Institute
AP	Access Point
APNIC	Asie (pour Asian Pacific Network Information Center)
ARIN	American Registry for Internet Numbers
ARP	Address Resolution Protocol
ARQ	Automatic ReQuest
AS	Autonomous System
ASN	Autonomous System Number
ASN.1	Abstract Syntax Notation 1
ATM	Asynchronous Transfer Mode

B

BDR	Backup Designated Router
BEB	Binary Eponential Backoff
BID	Bridge ID
BBN	Bolt Beranek and Newman
BGP	Border Gateway Protocol
BPDU	Bridge PDU
BSS	Basic Service Set
BSSID	Basic Service Set Identification

C

CDDI	Copper Distributed Bus Data Interface.
CDP	Cisco Discovery Protocol
CFI	Canonical Format Indicator
CHAP	Challenge Handshake Authentification Protocol
CIDR	Classless Inter-Domain Routing
CCITT	Comité consultatif international pour la téléphonie et la télégraphie
CN	Correspondant Node
COA	Care of Address
CRC	Cyclic Redundancy Check
CSMA	Carrier Sense Multiple Access
CSMA/CD	Carrier Sense Multiple Access with Carrier Detection
CTS	Clear to Send

CWND	Congestion window

D

DAC	Dual Attachment Concentrator
DAS	Dual Attachment Station
DCF	Distributed Coordination Function
DHCP	Dynamic Host Configuration Protocol
DIFS	DCF Interframe Space
DISC	Disconnect
DM	Dense Mode protocol
DM	Disconnect Mode
DNS	Domain Name Service
DoD	Department of Defense
DQDB	Distributed Queue Dual Bus
DR	Designated Router
DS	Distribution System
DSSS	Direct sequence spread spectrum
DTIM	Delivery Traffic Indication Map
DVMRP	Distance Vector Multicast Routing Protocol

E

EAP	Extensible Authentification Protocol
EGP	Exterior gateway protocol
EIFS	Extended Interframe Space
E-RIF	Embedded Routing Information Field
ESS	Extended Service Set 160

F

FA	Foreign Agent – Agent visité
FC	Frame Control
FCS	Frame Check Sequence
FDDI	Fiber Distributed Data Interface
FHSS	Frequency Hopping Spread Spectrum
FLP	Fast Link Pulse
FTP	File Transfer Protocol

G

GARP	Generic Attribute Registration Protocol
GBIC	Gigabit Ethernet Interface Converter
GSM	Global System for Mobile Communications
GPRS	General Packet Radio Service
GUI	Graphical User Interface
GVRP	GARP VLAN Registration Protocol

H

HDLC	High-level Data Link Control
HA	Home Agent
HSSI	High-Speed Serial Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol

I

IA5	International Alphabet No 5
IAB	Internet Architecture Board
IANA	Internet Assigned Number Authority
ICI	Interface Control Information
ICMP	Internet Control Message Protocol
IDU	Interface Data Unit
IEEE	Institute of Electrical and Electronic Engineers
IETF	Internet Engineering Task Force
IFG	Inter Frame Gap
IGMP	Internet Group Message Protocol
IMAP	Internet Mail Access Protocol
IMP	Interface Message Processor
IBSS	Independant BSS
IP	Internet Protocol
IPCP	Internet Protocol Control Protocol
ISL	Inter Switch Link
ISM Band	Industrial, Scientific and Medical Band
ISO	International Standard Organization
ISP	Internet Service Provider

L

LAN	Local Area Network
LC	Late Counter
LCP	Link Control Protocol
LCW	Link Code Word
LE	Local Exchange/Central téléphonique local
LNPA	ligne numérique à paire asymétrique
LLC	Logical Link Control
LSA	Link State Advertisement
LSDB	Link State Database
LSP	Link State Packet

M

MAC	Medium Access Control
MAM	Medium Access Management
MAN	Metropolitan Area Network

MAU	Medium Attachment Unit
MDI	Medium Dependant Interface/Interface
MDS	Multipoint distribution system
MIB	Management Information Base
MII	Media-Independent Interface
MIT	Management Information Tree
MN	Mobile Node
MTU	Maximum Transfer Unit
MRT	multiplexage par répartition dans le temps
MSS	Maximum segment size

N

NAT	Network Address Translation
NAV	Network Allocation Vector
NBMA	Nonbroadcast Multiaccess
NCP	Network Control Protocol
NLP	Normal Link Pulse
NRZ	Non-retour à zéro
NRZI	NonReturn to Zero Inverted
NSDU	Network Service Data Unit

O

OFDM	Orthogonal Frequency Division Multiplexing
OSI	Open System Interconnection
OSPF	Open Shortest Path First
OUI	Organizational Unit Identifier

P

P/F	Poll/Final
PAM	Pulse Amplitude Modulation
PAP	Password Authentification Protocol
PCF	Point Coordination Function
PCI	Protocol Control Information
PDH	Plesiochronous Digital Hierarchy
PDU	Protocol Data
PFC	Protocol Field Compression
PIFS	PCF Interframe Space
PIM	Protocol Independent Multicast
PLS	Physical Layer Signaling
PMA	Physical Medium Attachment
POP3	Post Office Protocol – version 3
PPP	Point-to-Point Protocol
PPS	Packet per second
PSN	Packet Switch Node

Q

QoS	Quality of Service

R

RARP	Reverse Address Resolution Protocol
REJ	Reject
RFC	Request for Comment
RII	Routing Information Indicator
RIP	Routing Information Protocol
RIPE NCC	Réseaux IP Européens et pour Network Coordination Center
RNIS	Réseaux numériques d'intégration de services
RNR	Receive Not Ready
RP	Rendez vous Point
RP	Ring Purge
RPF	Reverse Path Forward
RPT	RP tree
RR	Receive Ready
RTS	Request to Send
RTT	Round Trip Time

S

SABME	Set Balance Mode Extended
SAC	Single Attachment Concentrator
SAP	Service Access Point
SAS	Single Attachment Station
SD	Start Delimiter
SDH	Synchronous Digital Hierarchy
SDU	Service Data Unit
SFD	Start Frame Delimiter
SIFS	Short Interframe Space
SLIP	Serial Line Internet Protocol
SM	Sparse Mode protocols
SMI	Structure Management Information
SMTP	Simple Mail Transfer Protocol
SNAP	SubNet & Access Protocol
SNMP	Simple Network Management Protocol
SONET	Synchronous Optical Network
SPT	Shortest path tree
SSAP	Source Service Access Point
STP	Spanning Tree Protocol
SDM	système de distribution multipoint

T

TAM	Timer Active Monitor
TBTT	Target Beacon Transmission Time
TCI	Tag Control Information
TCN	Topology Change Notification
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TFTP	Trivial File Transfer Protocol
THT	Token Holding Time
TIM	Traffic Indicator Map
TLD	Top-Level Domain-
TOS	Type of Service
TRP	Timer ring purge
TRT	Token Rotation Timer
TTRT	Target Token Rotation Time

U

UA	Unnumbered Acknowledgement
UDP	User Datagram Protocol
UI	Unnumbered Information
UIT-T	Union Internationale des télécommunications - Télécom
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
USB	Universal Serial Bus
UTP	Unshielded Twisted Pairs

V

VDSL	Very-high-data-rate digital subscriber line
VLAN	Virtual Local Area Network
VLSM	Variable Length Subnet Mask
VPID	VLAN Protocol Identification
VSAT	Very Small Aperture Terminal

W

WAN	Wide Area Network
WEP	Wired Equivalent Privacy
WI-FI	Wireless Fidelity
WWW	World Wide Web HTTP

X

XID	Exchange Information

BIBLIOGRAPHIE

[BEA00] Beau Williamson, « *Developing IP Multicast Networks* » vol 1, Cisco Press 2000.

[CAR00] Carlson J., « *PPP Design, Implementation, and Debugging* », 2nd ed., Addison Wesley, Reading, MA, 2000.

[COM01] Comer E. D., « *Computer Networks and Internet* », 3rd edition, Prentice Hall, Upper Saddle River, NJ, 2001.

[GEI99] Geier J., « *Wireless LANs – Implementing Interoperable Networks* », Macmillan Network Architecture & Development series, USA, 1999.

[HAL01] Halsall F., « *Multimedia Communications* », Addison Wesley, Reading, MA, 2001.

[IEE90] ANSI/IEEE « *Std 802.4, Part 4: Token Passing Bus Access Method and Physical Layer Specifications* », IEEE Computer Society 1990.

[IEE98E] ANSI/IEEE « *Std 802.5, Part 5: Token Ring Access Method and Physical Layer Specifications* », IEEE Computer Society 1998.

[IEE98] ANSI/IEEE « *Std 802.2, Part 2: Logical Link Control* », IEEE Computer Society 1998.

[IEE99] ANSI/IEEE « *Std 802.11, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications* », IEEE Computer Society 1999.

[IEE00] IEEE « *Std 802.3, Part 3: Carrier Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications* », IEEE Computer Society 2000.

[IEE03] IEEE « *Std 802.1Q: Virtual Bridged Local Area Networks* », IEEE Computer Society 2003.

[KRI01] Krishnamurthy B. & Rexford J., « *Web Protocols and Practice* », Addison Wesley, Reading, MA, 2001.

[KUR03] Kuroze J. F.and Ross W. K., « *Computer Networking: a top-down approach featuring the Internet* », 2nd ed, Addison Wesley, Reading, MA, 2003.

[LEO00] Leon-Garcia A., Widjaja I., « *Communication Networks* », McGraw Hill, USA, 2000.

[LOS00] Loshin P., « *Essential Email Standards: RFC and Protocols Made Practical* », John Willey & Sons, 2000.

[NUS87] Nussbaumer H., « *Téléinformatique I* », Presses Polytechniques Romandes, 1987.

[NUS872] Nussbaumer H., « *Téléinformatique II* », Presses Polytechniques Romandes, 1987.

[OHA99] O'Hara B., Petrick A., « *802.11 Handbook – A Designer's Compagnion* », IEEE Press, NY, NY 1999.

[ROS90] Rose M. T., « *The Open Book – A Practical Perspective on OSI* », Prentice Hall, Upper Saddle River, NJ, 1990.

[ROS96] Rose M. T., « *The Simple Book* », 2nd edition, Prentice Hall, Upper Saddle River, NJ, 1996.

[SIM99] Simoneau P. « *SNMP Network Management* » McGraw Hill, USA, 1999.

[STA99] Stallings W, « *SNMP, SNMPv2, SNMPv3 and RMON 1 and 2* », 3th ed., Adisson Wesley, Reading, MA 1999.

[STA04] Stallings W, « *Data and Computer Communications* », 7th ed., Pearson, Prentice Hall, N.J. 2004.

[STE94] Stevens W. R., « *TCP/IP Illustrated, vol 1, The protocols* », Addison Wesley, Reading, MA, 1994.

[SUB00] Subramanian M., « *Network Management – Principles and Practice* », Addison Wesley, Reading, MA, 2000.

[TOU99] Laurent Toutain, « *Réseaux locaux et Internet – des protocoles à l'interconnexion* », 2e édition, Hermes 1999.

[UDU99] Udupa D. K., « *TMN Telecommunication Management Network* », McGraw Hill, USA, 1999.

[X200] Study Group VII « *Reference Model of Open Systems Interconnection for CCITT Applications* », Blue Book Volume VIII – Fascicle VIII.4, ITU-T 1988.

[X212] Study Group VII « *Data Link Service Definition for Open Systems Interconnection for CCITT Applications* », Blue Book Volume VIII – Fascicle VIII.4, ITU-T 1988

[YEA96] Yeager J. N, and McGrath R. E., « *Web Server Technology* », Morgan Kaufman, San Francisco, CA, 1996.

1

1000BASE-LX 147
1000BASE-SX 146, 147
100BASE-FX 141
100BASE-T2 141
100BASE-TX 138

4

4B/5B NRZI 138

8

8B/10B 146

A

ABR (*Area Border Router*) 388
accès Internet par fibre optique 6
accès Internet par satellite 7
accès réseau 2
acheminées vers le port (*Forwarding*) 241
acheminement par la source (*Source Routing Bridging*) 233, 243
adresse MAC 124
adresse réseau 291
adresses IP 290
ADSL (*Asymmetric Digital Subscriber Line*) 5

Agent mandataire (*Proxy*) 516
Agents de relais DHCP 352
AID 179, 181
aires de OSPF 387
algorithme de Bellman-Ford 415
algorithme de Dijkstra 411
algorithme de temporisation de retransmission (TBEB) 115
ALOHA 113
anneau à jeton 199
ANSI 209
AP 155
apprentissage d'adresse (*Address Learning*) 241
arborescence partagée (*Shared Tree*) 424
ARP 322
ARP gratuit 331
ARP mandataire 329
ARPAnet 14
ARQ (*Automatic Request*) 82

B

Backoff Delay 114
bande de base (*Baseband*) 24
BEB (*Binary Exponantial Backoff*) 114, 159
BGP (*Border Gateway Protocol*) 376, 408
Bolt Beranek and Newman (BBN) 14
BSS 155
bus à jeton 191

C

câbles coaxiaux 17
CCITT 31
CDDI 219
Center based trees 423
CFI 270
CIDR 302

circuits virtuels 55
classes d'adresses 291
client-serveur 476
codage Manchester 128
commutation de circuits 9
commutation de messages 12
commutation par paquets 10
compteur de retransmission 88
concentrateurs (*Repeater*) 231
Congestion Avoidance 464
connexion non persistante 477
connexion persistante 477
contiguïté (*Adjacency*) 403
contrôle de congestion pour le protocole TCP 461
couche Transport 36, 437
Couches fonctionnelles 33
couches OSI 45
courrier électronique 490
CSMA (*Carrier Sense Multiple Access*) 113
CSMA/CA 159
CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*) 113
CTS 164

D

Démarrage lent (*Slow Start*) 461
DHCP 341
DIFS 160
DISC 78, 80
DM 78, 80
DNS 357
DoD (*Department of Defense*) 56
dotted-decimal 290
DQDB 208
DS 156
DSSS 153
DTIM 182
Dual Attachment Station 221
durée de rotation RTT 340
DVMRP 427

E

EGP (*Exterior Gateway Protocol*) 408
éléments gérés (*Managed Objets*) 504
Encapsulation 46
Entité 44
E-RIF 269
ESS 155
état de lien (*Link State*) 375, 386

F

Faisceaux hertziens 21
Fast Ethernet 130
FDDI 208
fenêtre (couche liaison) 88
fenêtre coulissante (TCP) 457
FHSS 153
fibres optiques 18
fonction pause 132
Fragmentation 287
FTP 484
fureteur 476

G

GARP 269
gestion de réseau 501
gestion de réseau TCP/IP 502
Gigabit Ethernet 145
GO BACK N 82
GSM/GPRS (*Global System for Mobile CommunicationsGeneral Packet Radio Service*) 6
GVRP 269

H

HDCL (*High-level Data Link*/Procédure de commande à haut niveau) 41
HDLC 66
Horizon coupé 385
HTTP 475

I

IANA 361
IBSS 155
ICI (*Interface Control Information*) 47
ICMP 330
identificateur d'objet (*OBJECT IDENTIFIER*) 505
IDU (*Interface Data Unit*) 47
IEEE802.11 153
IEEE802.1Q 270
IEEE802.3 122
IEEE802.4 191
IEEE802.5 199
IEEE802.2 65
IETF 276
IGMP 428
IMP (*Interface Message Processor*) 14
IP (*Internet Protocol*) 275
IPCP 106
IPv4 283
IPv6 308
ISM (*Industrial, Scientific and Medical Band*) 6
ISO (*International Standard Organization*) 31
ISP 92
Itinérance (*Roaming*) 183

K

Karn 469

L

large bande (*Broadband*) 24
LC 214
LCP 99
Liaison par satellite 21
ligne numérique à paire asymétrique (LNPA) 5
Link Code Word 133
LLC 64
LSA (*Link State Advertisement*) 386
LSDB (*Link State Database*) 386

M

MAM (*Medium Access Management*) 121
MAU 129
méthode du vieillissement 399
MIB 503
MII 142
MIT (*Management Information Tree*) 504
moniteur actif 205
MTU 286
multi distribution (*Multicast*) 419
multiplexage par répartition dans le temps (MRT) 26
multiplexage temporel asynchrone 10

N

Nagle 455
NAT 354
NAV 169, 164, 168
NCP 97
NCP 275
NLP 134
Not so Stubby Area 388
NS 193
nslookup 368

O

Organizational Unit Identifier - OUI 122
orienté connexion (*Connection Oriented*) 51
OSI (*Open System Interconnection*) 31
OSPF 386

P

Paires symétriques ou torsadées 16
PAM 141
paquets par seconde 125
PAT 358
PCF 166
PCI (*Protocol Control Information*/Informations de contrôle de protocole) 37
PDH (*Plesiochronous Digital Hierarchy*) 26
PDU (*Protocol Data Unit*/Unité de données de protocole) 37
perte du jeton 205
PIFS 168
Piggy Backing 87
PIM DM 427
PIM SM 427
PING 338

pipelining 479
PLS (*Physical Layer Signaling*) 127
PMA 127
pontage transparent (*Transparent Bridging*) 239
ponts (*Bridges*) 233
POP3 494
ports UDP et TCP 438
Préambule 121
primitives 43
problèmes de convergence 383
propagation rayonnée 20
protocole ISL 270
protocole PPP 93
protocole RIP 377
protocole SLIP 92
protocole SNMP 501
protocole STP (*Spanning Tree Protocol*) 246
protocoles 44
PS 193
PSN 275
Pwr Mng 180

R

RARP 331
Receiver-based Multicast 423
réseaux commutés 9
réseaux étendus (WAN) 8
réseaux méropolitains 8
resolve-contention 194
RFC 276
RNIS (Réseaux numériques d'intégration de services) 1
Roaming 183
RP (*Rendez-vous Point*) 425
RTS 164

S

SABME 79
sans connexion (*Connectionless*) 51
SAP 43
SDU (*Service Data Unit*/Unité de données de service) 37
Segmentation 48
séquence à sucette 399
série X.200 31
services confirmés 68
services du LLC 66
services non confirmés 68
set-successor 194
SFD (*Start Frame Delimiter*) 121
Shortest Path Tree SPT 428
SIFS 160
Silly Window Syndrome 469
Single Attachment Station 222
Sliding Window 457
Slow start 462
SMI 504
SMTP 492
SNAP (*Sub Net Access Protocol*) 122
SNMPv2 517
solicit-successor 194
SONET (*Synchronous Optical Network*/Réseau optique synchrone) 28
Source based routing 423
sous-adressage 294
Sous couche MAC (*Medium Access Control*) 64, 119
sous-réseau 296
sous-réseautage 298
Split Horizon with poisoned reverse 385
SSID 178
Steiner trees 423
Store and Forward 490
Stub Area 388
Suite TCP / IP 321
synchronisation 21
synchronisation de bits 21

synchronisation d'octets 22
synchronisation de trames 23
système de distribution multipoint (SDM) 6

T

T1 4
TBTT 168
TCI (*Tag Control Information*) 269
TCP 437
TCP/IP 275
TCP NewReno 466
TCP Reno 466
TCP Tahoe 465
techniques à compétition 113
Temporisateurs RIP 380
TFTP 487
THT 214
TIM 182
Time slot/Intervalle de temps 113
Token Bus 191
Token Ring 191
trame balisage « *Beacon* » 206
trame BPDU 253
trame information 75
trame non numérotée 78
trame REJ 78
trame REJECT 86
trame RNR 77, 84
trame RR 77, 83
trame supervision 76
trame TEST 78
trame UI 91
trame XID 78
transmission par faisceaux hertziens 21
TRT 214
TTRT 214
Type d'objet (OBJECT IDENTIFIER) 505

U

UA 78
UDP 439
URI (*Uniform Resource Identifier*) 476
URL (*Uniform Resource Locator*) 476
UTT-T 31

V

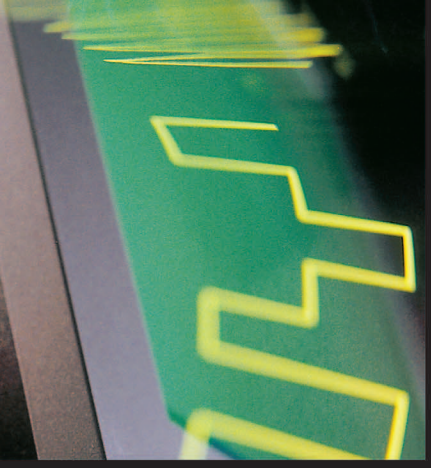
VDSL (*Very-high-data-rate Digital Subscriber Line*) 5
vecteur de distance (*Distant Vector*) 373
VLAN 262
VLSM 299
VPID 269
VSAT (*Very Small Aperture Terminal*) 7

W

WEP 180
who-follows 195
WI-FI 153

X

X.200 ..... 31, 45
X.214 45
X.222 46
X.224 46
X.227 46
X.228 46
X.229 46
X.400 490



Comme toute discipline qui se respecte, le domaine de la réseautique exige de grandes connaissances et couvre un spectre de spécialités assez vaste. C'est un domaine qui passionne autant les informaticiens que les ingénieurs en génie logiciel, les ingénieurs en télécommunication et les techniciens en réseaux.

Le présent ouvrage, *Protocoles et réseaux locaux*, a été conçu de manière à offrir les connaissances fondamentales en ce qui a trait au domaine de la réseautique, et tout particulièrement aux réseaux locaux et à l'accès à Internet.

Les protocoles des réseaux locaux comme Ethernet, les réseaux sans fil et les réseaux à jeton y sont traités. L'interfonctionnement entre réseaux locaux ainsi que l'application des réseaux locaux virtuels (VLAN) sont présentés. L'accent est mis, par la suite, sur le protocole IP et les protocoles qui le supportent, comme ARP, ICMP, DHCP, NAT et DNS. Les protocoles de routage utilisés dans Internet ainsi que les concepts du multicast sont aussi définis en détail. Les protocoles de la couche transport, notamment TCP et UDP, sont décrits en prenant en compte des méthodes de contrôle de flux et de congestion. Les protocoles d'applications typiques d'Internet comme HTTP, FTP, TFTP, SMTP et POP3, qui permettent l'accès au Web, le transfert de fichiers et le courrier électronique, ainsi que le protocole de gestion de réseau SNMP sont expliqués de façon claire et simple.

Michel Kadoch est professeur titulaire au Département de génie électrique de l'École de technologie supérieure de l'Université du Québec. Il est directeur de la maîtrise en génie logiciel et directeur du laboratoire de recherche LAGRIT (Laboratoire de gestion informatique et de télécommunication). Il est très actif en recherche et ses champs d'intérêt sont l'analyse de performance de réseaux de télécommunications, le développement de la qualité de services sur Internet, les réseaux sans fil, les réseaux Multicast, les voix sur IP, la téléphonie IP, les réseaux à hauts débits et la gestion de réseaux. Le professeur Kadoch est souvent sollicité par des entreprises pour son expertise. Il a dirigé pendant plus de 15 ans des projets sur le développement de nouvelles technologies auprès d'entreprises canadiennes. Il a été rapporteur spécial au sein de l'UIT-T (Union internationale des télécommunications) pendant plusieurs années. Il est également professeur associé à l'Université Concordia, à Montréal.

PUQ.CA

PUQ ISBN 978-2-7605-3479-7

ETS ISBN 978-2-921145-79-4

9 782760 534797