République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MUSTAPHA STAMBOULI DE MASCARA

FACULTÉ DES SCIENCES ET TECHNOLOGIE

Polycopié de Cours

Architectures Réseaux

Présenté par **:**

Ce cours est destiné aux étudiants de Master spécialité Réseaux et systèmes distribués (RSD)

Algérie
2015

Avant Propos

Ce cours est destiné aux étudiants de première année Master informatique spécialité Réseaux et systèmes distribués.

Ce module s'inscrit dans la continuité du module RÉSEAUX de Licence et a pour but de sensibiliser les étudiants aux réseaux informatiques à la fois sur le plan de l'architecture physique d'un réseau, de l'architecture logicielle (en particulier de la notion de couches) et sur le plan des protocoles standard de communication TCP et UDP. Dans un premier temps, nous détaillons les principales architectures réseaux et nous donnerons un aperçu sur les réseaux de type WIFI, WIMAX, xDSL, P2P...

Ensuite nous nous intéressons au modèle de référence OSI et TCP/IP. Dans les chapitres suivants, Nous replongeons dans les protocoles TCP/IP en commençant par les protocoles de la couche IP, notamment ARP/RARP, ICMP, IGMP. Un rappel sur l'adressage avec et sans classe, les sous réseau et la translation NAT sera donné. Nous passerons par la suite au deux classes de routage statique et dynamique en décrivant les protocoles RIP et OSPF. Nous donnerons aussi une brève présentation de la version 6 du protocole IP (IPV6). Les protocoles TCP et UDP seront présentés dans le chapitre suivant. Finalement nous décrirons quelques applications réseau.

Ce module contient des modalités d'évaluation pour chaque partie du cours.

Table des matières

Chapitre IV : Couche transport

Chapitre V : Applications réseau

Chapitre I Introduction

Avant de nous attaquer aux infrastructures réseaux, reprenons quelques notions de base sur les réseaux informatiques en général.

Un réseau permet de partager des ressources entre plusieurs ordinateurs: données ou périphériques (Imprimante, sauvegarde sur bandes, modem, scanner, ...). La première partie de ce cours reprend toutes les informations permettant de connecter ces ordinateurs entre eux.

La transmission d'information entre deux programmes informatiques sur deux machines différentes passe par deux modèles: le modèle OSI ou le modèle TCP/IP. Ces deux normes permettent à chaque partie de la communication de dialoguer. Chaque modèle inclut plusieurs couches. Chaque couche doit envoyer (et recevoir pour l'autre PC) un message compréhensible par les deux parties, compatibilité des informations [1].

I.1. Principales Architectures Réseaux

I.1.1. LAN (Local Area Network ou Réseau Local)

Un tel réseau permet de relier des ordinateurs et des périphériques situés à proximité les uns des autres (dans un même bâtiment, par exemple).
C'est le type de réseau le plus répandu dans les entreprises et ne comporte pas plus de 100 ordinateurs.

I.1.2. MAN (Métropolitain Area Network ou Réseau Métropolitain)

Il s'agit d'une série de Réseaux Locaux et permet de relier des ordinateurs situés dans une même ville.

I.1.3. WAN (Wide Area Network ou Réseau Etendu)

Ce réseau sert à relier des LAN situés dans un même pays ou dans le monde.
Lorsqu'un WAN appartient à une même entreprise, on parle souvent de Réseau d'Entreprise.

I.1.4. Pair-à-Pair (P2P)

Fondé sur une relation d'égalité entre les applications qui échangent des informations sur un réseau. Plusieurs applications identiques sont installées sur des machines différentes, et échangent deux à deux des données et des services, en jouant tour à tour le rôle de client et de serveur.
L'exemple le plus connu d'applications P2P concerne les applications d'échange de fichiers sur Internet, où chaque Internaute met à disposition certains fichiers dont il dispose pour permettre aux autres de les télécharger, et en télécharge en même temps.

I.1.5. Réseaux informatiques sans fil (wireless network)

Un réseau dans lequel au moins deux terminaux (ordinateur portable, PDA, etc.) peuvent communiquer sans liaison filaire. En utilisant des ondes radio-électriques (radio et infrarouges).

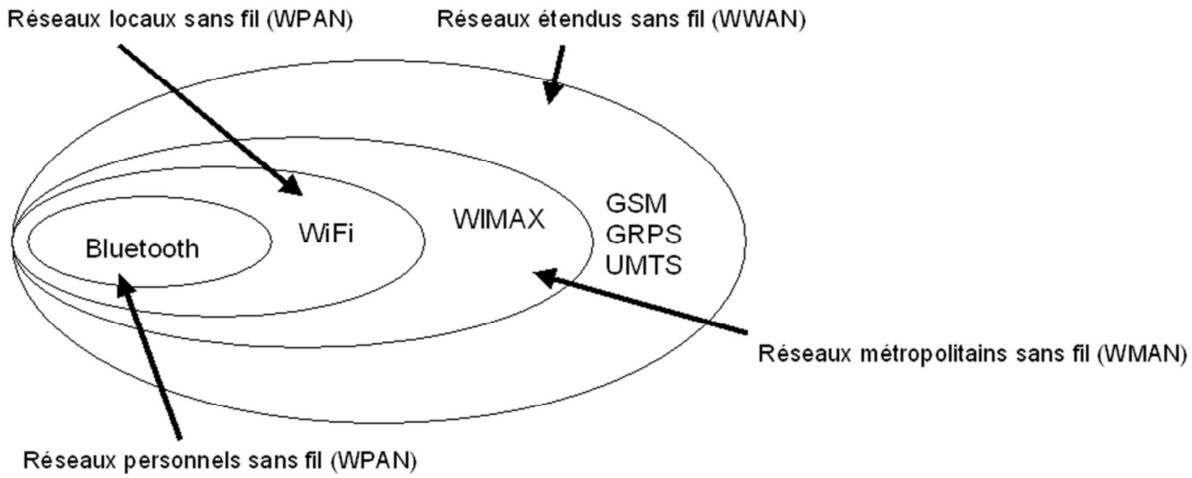


Figure I.1. *Réseaux informatiques sans fil*

WPAN (Wireless Personal Area Network), inclus les réseaux sans fil d'une faible portée (quelques dizaines mètres). Ce type de réseau sert généralement à relier des périphériques.

La technologie Bluetooth
C'est la technologie sans fil la plus connue, dans sa gamme, normalisée par l'IEEE (802.15). Elle utilise la bande de fréquences radio des 2,4GHz et un débit théorique de 1 Mbps pour une portée maximale d'une trentaine de mètres. Son gros avantage est d'utiliser peu d'énergie.

WLAN, pour Wireless Local Area Network, est un réseau qui permet une portée d'environ une centaine de mètres.

La technologie WiFi
WiFi (Wireless Fidelity) ou Ethernet sans fil : Réseau local de type Ethernet à accès sans fil. Il est devenu un moyen d'accès au haut débit à Internet, Cette technologie offre des débits allant jusqu'à 11Mbp/s sur une distance de plusieurs centaines de mètres..

WMAN, pour Wireless Metropolitan Area Network, Utilisé essentiellement par les opérateurs de télécommunication. Cette technologie propose un débit utile de 1 à 10 Mbit/s pour une portée de 4 à 10 kilomètres.

Le WiMAX (Worldwide Interoperability for Microwave Access) permet de réserver une bande-passante pour un usage donné. C'est le cas notamment de la voix sur IP (VoIP). En effet la communication orale ne peut pas accepter de coupures même minimes.

I.2. Rappel sur le modèle OSI

Le modèle OSI (Open System Interconnection Model) est un modèle théorique définit en 1977, il régit la communication entre deux systèmes informatiques selon sept couches. A chaque couche, les deux systèmes doivent communiquer "compatibles".

Dans le découpage en 7 couches, on distingue [2]:

- Les couches basses (1-4) : transfert de l'information par les différents services de transport.
- Les couches hautes (5-7) : traitement de l'information par les différents services applicatifs.

Application	⇓	Couche Application	7	Couche Application	⇑	
	⇓	Couche Présentation	6	Couche Présentation	⇑	
	⇓	Couche Session	5	Couche Session	⇑	
Transport des données	⇓	Couche Transport	4	Couche Transport	⇑	
	⇓	Couche Réseau (Network)	3	Couche Réseau (Network)	⇑	Paquet
	⇓	Couche liaison de données (Data Link)	2	Couche liaison de données (Data Link)	⇑	Trames
	⇒	Physique (Physical)	1	Couche Physique (Physical)	⇒	BIT
		Support de communication				

Figure I.2. *Représentation du modèle OSI [1].*

I.2.1. Couche application

La couche application est le point d'accès des applications aux services réseaux. On y retrouve toutes les applications de communication via le réseau communément utilisées sur un LAN ou sur internet : applications de transfert de fichiers, courriers électroniques, etc [2].

Ex : Navigateurs (HTTP), transfert de fichiers (FTP), clients email (SMTP).

I.2.2. Couche Présentation

La couche présentation, s'occupe de la mise en forme des données, c'est à dire qu'elle a en charge de structurer et convertir les données échangées ainsi que leur syntaxe afin d'assurer la communication entre nœuds disparates (différences hardware et/ou software) [2].

Ex : Codage des données, cryptage, compression des données.

I.2.3. Couche session

La couche session gère une communication complète entre plusieurs nœuds, permettant ainsi d'établir et de maintenir un réel dialogue, pouvant être constitué de temps morts pendant lesquels aucune donnée n'est physiquement transmise [2].

Ex : une connexion http avec suivi de navigation sur un même site web, une connexion FTP.

I.2.4. Couche transport

La couche transport, supervise le découpage et le réassemblage de l'information en paquets, contrôlant ainsi la cohérence de la transmission de l'information de l'émetteur vers le destinataire [2].

Ex : techniques de communication par paquets, fragmentation.

I.2.5. Couche réseau

La couche réseau, a en charge de déterminer le choix de la route entre les nœuds afin de transmettre de manière indépendante l'information ou les différents paquets la constituant en prenant en compte en

temps réel le trafic. Cette couche assure aussi un certain nombre de contrôle de congestion qui ne sont pas gérés par la couche liaison [2].

Ex : techniques de communication de données (circuits, paquets, etc.).

I.2.6. Couche liaison de données

La couche liaison de données, s'occupe de la bonne transmission de l'information entre les nœuds via le support, en assurant la gestion des erreurs de transmission et la synchronisation des données [2].

Ex : gestion des erreurs (contrôles de parité, CRC, etc.), synchronisation, multiplexage.

I.2.7. Couche physique

La couche physique, gère la communication avec l'interface physique afin de faire transiter ou de récupérer les données sur le support de transmission [2].

Ex : Interconnexion avec le support physique de transmission (paire torsadée, fibre optique, etc.).

A chacun de ces niveaux du modèle OSI, on encapsule un en-tête et une fin de trame (message) qui comporte les informations nécessaires en suivant les règles définies par le protocole utilisé. Ce **protocole** est le langage de communication pour le transfert des données (TCP/IP, NetBui, IPX sont les principaux) sur le réseau informatique.

I.3. Rappel sur le modèle TCP/IP

Le modèle TCP/IP s'inspire du modèle OSI auquel il reprend l'approche modulaire mais réduit le nombre à quatre. Les trois couches supérieures du modèle OSI sont souvent utilisées par une même application. Ce n'est pas le cas du modèle TCP/IP. C'est actuellement le modèle théorique le plus utilisé[7].

<table>
<tr><th>Protocoles utilisés</th><th>Modèle TCP/IP</th><th>correspondance en OSI</th></tr>
<tr><td rowspan="3"></td><td rowspan="3">Couche application</td><td>Application</td></tr>
<tr><td>Présentation</td></tr>
<tr><td>Session</td></tr>
<tr><td>TCP / UDP, gestion des erreurs</td><td>Couche Transport</td><td>Transport</td></tr>
<tr><td>IP / ARP et RARP /ICMP / IGMP</td><td>Couche Internet</td><td>Réseau</td></tr>
<tr><td rowspan="2"></td><td rowspan="2">Couche Accès réseau</td><td>Liaison de donnée</td></tr>
<tr><td>Physique</td></tr>
</table>

Figure I.3. *Représentation du modèle TCP/IP [1].*

De nouveau, on ajoute à chaque niveau un en-tête, les dénominations des paquets de données changent chaque fois [1, 12]:

- Le paquet de données est appelé **message** au niveau de la couche application
- Le message est ensuite encapsulé sous forme de **segment** dans la couche transport. Le message est donc découpé en morceau avant envoi pour respecter une taille maximum suivant le MTU.
- Le segment une fois encapsulé dans la couche Internet prend le nom de **datagramme**
- Enfin, on parle de **trame** envoyée sur le réseau au niveau de la couche accès réseau

Les **couches du modèle TCP/IP** sont plus générales que celles du modèle OSI.

I.3.1. Accès réseau

Cette couche intègre les services des couches physique et liaison du modèle OSI, a en charge la communication avec l'interface physique afin de transmettre ou de récupérer les paquets de données qui lui sont transmis de la couche supérieure [2].

I.3.2. Couche internet

Cette couche correspond à la couche réseau du modèle OSI. Elle s'occupe de l'acheminement, à bonne destination, des paquets de données indépendant les uns des autres, soit donc de leur routage à travers les différents nœuds par rapport au trafic et à la congestion du réseau. Il n'est en revanche pas du ressort de cette couche de vérifier le bon acheminement.

Le protocole IP (internet protocol) assure intégralement les services de cette couche, et constitue donc l'un des points clés du modèle TCP/IP [2].

I.3.3. Couche transport

Gere le fractionnement et le réassemblage en paquets des flux de données à transmettre. Le routage ayant comme conséquence un arrivage des paquets dans un ordre incertain, cette couche s'occupe aussi du réagencement ordonné de tous les paquets d'un même message [2].

Les deux principaux protocoles pouvant assurer les services de cette couche sont les suivants :

- TCP (Transmission Control Protocol) : protocole fiable, assurant une communication sans erreur par un mécanisme question/réponse/confirmation/synchronisation (orienté connexion) ;
- UDP (User Datagram Protocol) : protocole non fiable, assurant une communication rapide mais pouvant contenir des erreurs en utilisant un mécanisme question/réponse (sans connexion).

I.3.4. Couche application

Un grand nombre de protocoles divers de haut niveau permettent d'assurer les services de cette couche [2] :

- Telnet : ouverture de session à distance ;
- FTP (File Transfer Protocol) : protocole de transfert de fichiers ;
- HTTP (HyperText Transfert Protocol) : protocole de transfert de l'hypertexte ;
- SMTP (Simple Mail Transfer Protocol) : protocole simple de transfert de courrier ;
- DNS (Domain Name System) : système de nom de domaine ;
- Etc.

I.4. Exercices

Questions de cours

- Qu'est-ce qu'un réseau local?
- Qu'est-ce qu'un protocole?
- Pour chaque couche du modèle TCP/IP, citez quelques protocoles.
- Dans quel sens se fait l'encapsulation des données dans la pile TCP/IP.
- Quel est le rôle du NAT.

Exercice 1

Un réseau de classe B dispose du masque de sous-réseau 255.255.240.0.

- Quel est le nombre maximum d'ordinateurs que l'on peut raccorder à chaque sous-réseau ?
- Combien de sous-réseaux y a-t-il ?

- Définissez l’adresse broad cast du sous réseau N° 2.
- Définissez l’adresse de la machine N°2 du sous réseau N° 3.

Exercice 2

En général, les ordinateurs en réseau sont dotés d’une carte réseau Ethernet, et utilisent le protocole TCP/IP pour la communication.

a) Quand se sert-on d’une adresse Ethernet (physique), et quand se sert- on d’une adresse IP ?

b) A quel moment doit-on utiliser une correspondance entre les deux types d’adresses ? Et quel est le protocole responsable de cette étape.

Chapitre II : Protocoles de la couche IP

C'est le réseau Internet qui a introduit le protocole IP. Ce protocole a été ensuite repris pour réaliser des réseaux privés, tels les réseaux intranet et extranets ou les réseaux mis en place pour la domotique.

II.1. Protocoles ARP/RARP

Internet propose l'interconnexion de réseaux physiques par des routeurs. Pour obtenir l'interfonctionnement des différents réseaux, la présence du protocole IP est nécessaire dans les nœuds qui effectuent le routage entre les réseaux. Globalement, Internet est un réseau à transfert de paquets. Les paquets traversent plusieurs sous-réseaux pour atteindre leur destination, sauf bien sûr si l'émetteur se trouve dans le même sous réseau que le récepteur. Les paquets sont routés dans les passerelles situées dans les nœuds d'interconnexion. Ces passerelles sont des routeurs. De façon plus précise, ces routeurs transfèrent des paquets, en déterminant pour chaque paquet la meilleure route à suivre.

Le réseau Internet a été développé pour mettre en relation des machines du monde entier, auxquelles on a pris soin d'attribuer des adresses IP. Ces adresses IP n'ont aucune relation directe avec les adresses des cartes coupleurs qui permettent aux PC de se connecter au réseau. Ces dernières sont des adresses physiques. Pour envoyer un datagramme sur Internet, le logiciel réseau convertit l'adresse IP en une adresse physique, utilisée pour transmettre la trame. La traduction de l'adresse IP en une adresse physique est effectuée par le réseau sans que l'utilisateur s'en aperçoive.

Le protocole ARP *(Address Resolution Protocol)* effectue cette traduction en s'appuyant sur le réseau physique. ARP permet aux machines de *résoudre* les adresses sans utiliser de *table statique.* Une machine utilise ARP pour déterminer l'adresse physique du destinataire. Elle diffuse pour cela sur le sous réseau une requête ARP qui contient l'adresse IP à traduire. La machine possédant l'adresse IP concernée répond en renvoyant son adresse physique. Pour rendre ARP plus performant, chaque machine tient à jour, en mémoire, une table des adresses résolues et réduit ainsi le nombre d'émissions en mode diffusion.
Ce processus est illustré à la figure II.1 [13].

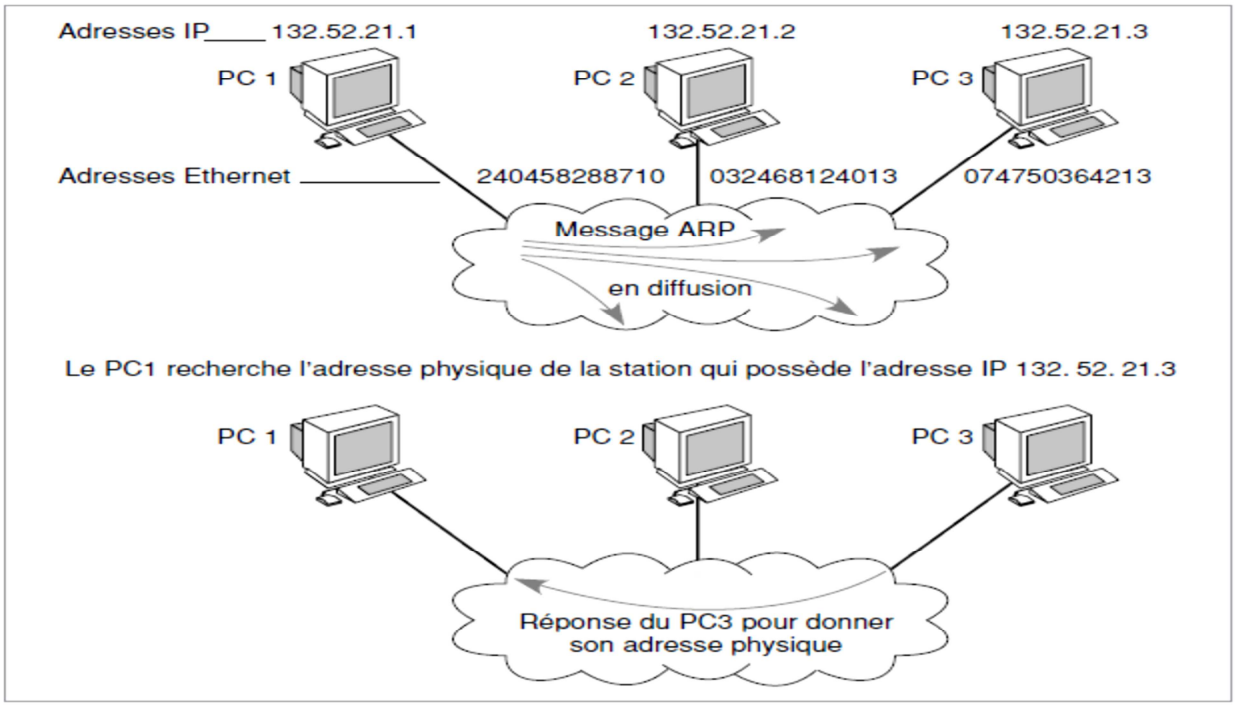


Figure II.1. *Fonctionnement du protocole ARP.*

De façon inverse, une station qui se connecte au réseau peut connaître sa propre adresse physique sans avoir d'adresse IP. Au moment de son initialisation *(bootstrap)*, cette machine doit contacter son serveur afin de déterminer son adresse IP et ainsi de pouvoir utiliser les services TCP/IP. Dans ce cas, le protocole RARP *(Reverse ARP)* permet à la machine d'utiliser son adresse physique pour déterminer son *adresse logique* sur Internet.
Par le biais du mécanisme RARP, une station peut se faire identifier comme cible en diffusant sur le réseau une requête RARP. Les serveurs recevant le message examinent leur table et répondent au client. Une fois l'adresse IP obtenue, la machine la stocke en mémoire vive et n'utilise plus RARP jusqu'à ce qu'elle soit réinitialisée.

II.2. Protocole ICMP (Internet Control Message Protocol)

La gestion et le contrôle sont des processus fortement imbriqués dans les nouvelles générations de réseaux IP. Pour permettre aux machines de rendre compte des anomalies de fonctionnement, on a ajouté à Internet un protocole d'envoi de messages de contrôle, appelé ICMP *(Internet Control Message Protocol).*
Le destinataire d'un message ICMP n'est pas un processus application mais le logiciel Internet de la machine. Ce logiciel IP traite le problème porté par le message ICMP à chaque message reçu.
Les messages ICMP ne proviennent pas uniquement des passerelles. N'importe quelle machine du réseau peut envoyer des messages à n'importe quelle autre machine. Les messages permettent de rendre compte de l'erreur en remontant jusqu'à l'émetteur d'origine. Les messages ICMP prennent place dans la partie donnée des datagrammes IP. Comme n'importe quels autres datagrammes, ces derniers peuvent être perdus. En cas d'erreur d'un datagramme contenant un message de contrôle, aucun message de rapport de l'erreur n'est transmis, afin d'éviter les *avalanches* [13].

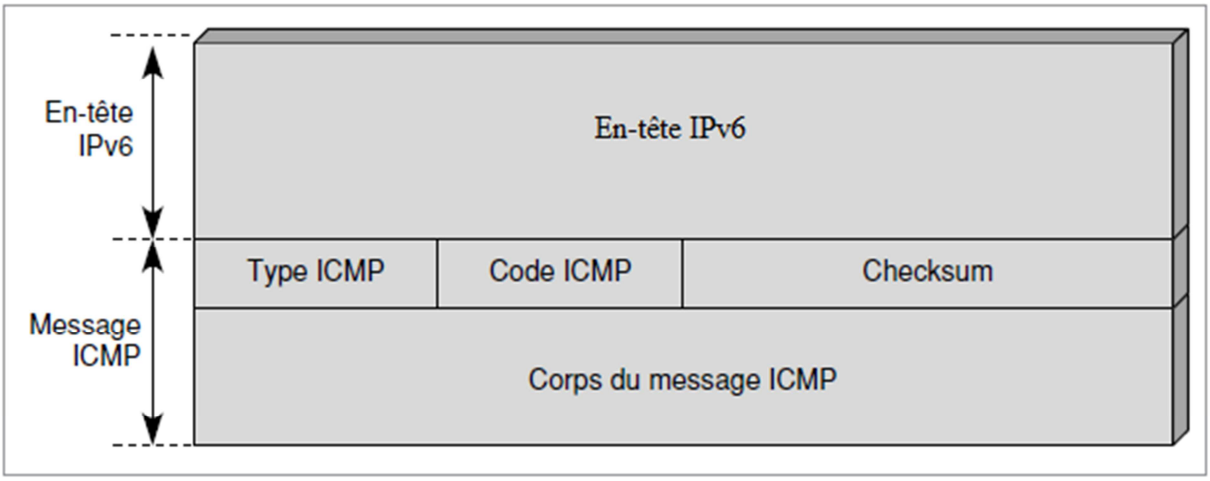


Figure II.2. *Format des messages ICMP.*

La figure II.2 illustre le format des messages ICMP. L'en-tête de la partie ICMP comprend un octet « type » de message, suivi d'un octet « code », suivi de deux octets de checksum. Le type et le code différencient les différents messages ICMP.

Il en existe 14 types différents. Ces messages sont les suivants :

- 1 : message d'erreur, impossible d'atteindre la destination ;
- 2 : message d'erreur, paquet trop volumineux ;
- 3 : message d'erreur, temps dépassé ;
- 4 : message d'erreur, problème de paramètre ;
- 128 : message de requête d'écho ;
- 129 : message de réponse d'écho ;

- 130 : requête d'entrée dans un groupe ;
- 131 : rapport sur l'entrée dans un groupe ;
- 132 : fin d'appartenance à un groupe ;
- 133 : sollicitation d'un routeur ;
- 134 : émission d'un routeur ;
- 135 : sollicitation d'un voisin (Neighbor Solicitation) ;
- 136 : émission d'un voisin (Neighbor Advertisement) ;
- 137 : message de redirection.

Le checksum ne s'applique ni au paquet IP, ni à la partie ICMP, mais à un ensemble de champs qui contiennent la partie ICMP, tels que les adresses émetteur et récepteur, la zone de longueur du paquet IP et le champ indiquant ce qui est encapsulé, c'est-à-dire la valeur 58, dans le cas présent.

II.3. Protocole IGMP (Internet Group Management Protocol)

Le protocole **IGMP** permet d'envoyer le même message à des machines faisant partie d'un groupe. Ce protocole permet également à ces machines de s'abonner ou de se désabonner d'un groupe. Ceci est utilisé par exemple dans la vidéo conférence à plusieurs machines, envoi de vidéos, ... Ce protocole permet de regrouper des stations [1].

II.4. Rappel sur l'adressage avec et sans classe (CIDR)

A chaque périphérique réseau physique ou logique correspond une adresse IP, une machine routeur a donc en général plusieurs adresses IP puisqu'il est normalement connecté à plusieurs réseaux. D'autre part à chaque réseau physique ou logique correspond une adresse de réseau, un « masque » et une adresse de diffusion.
Une adresse IP se décompose en une adresse de réseau dans les bits de poids forts (dont le nombre est à fixer par l'administrateur), les bits de poids faibles donnent l'adresse locale. Il y a 4 classes d'adresse utilisées, de A à D, qui instaurent une certaine hiérarchie. Le masque est un masque binaire de 32 bits (pour une adresse IPv4) qui possède un 1 à la position *i* si le bit *i* de l'adresse IP fait partie de l'adresse réseau. Ainsi, dans une adresse de classe C les 3 premiers octets sont la partie réseau. Le masque est donc 255.255.255.0 [14].

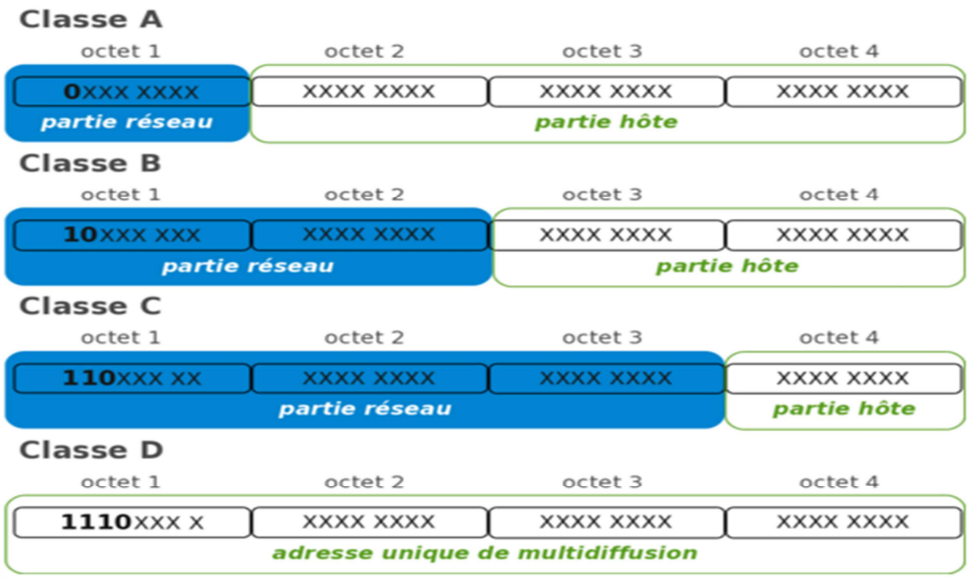


Figure II.3. *Classes d'adresses.*

Les adresses réseaux étant fixées, la partie affectée aux institutions (compagnies, universités...) peut-être gérer localement par l'administrateur. Celui-ci peut par exemple définir des sous réseaux en prenant une partie des bits réservés aux adresses de machines. Ensuite, pour les adresses de machines, on évite les adresses ayant tous les bits à 0 ou à 1. Par convention, ces adresses sont utilisées pour le broadcast (diffusion).

Par exemple, soit l'adresse 205.56.7.89 qui est de la classe C. L'adresse de réseau est 205.56.7.0, le masque 255.255.255.0. L'adresse pour la diffusion sur ce réseau est 205.56.7.255 car on prend tous les bits pour la partie machine à 1.

Depuis 1994, afin de réduire le gaspillage des adresses IP (du aux classes A et B) et la taille des tables de routage dans les routeurs inter-domaine, le système CIDR (Classless Inter Domain Routing) est utilisé pour agréger et attribuer des adresses de classe C consécutives. En fait, les adresses sont maintenant vues comme composée d'un préfixe de longueur variable et d'un nombre représentant le nombre de bits du masque de réseau. Ainsi par exemple l'adresse suivante
193.127.32.0/23 représente les adresses réseaux 193.127.32.0 et 193.127.33.0 chacun avec un masque de 255.255.255.0

Le nombre d'adresses réseaux référencées peut facilement être déduit du nombre de bits du masque. Ainsi en prenant comme exemple une adresse de classe C, /24 adresse un seul réseau, /23 adresse 2 réseaux, /22 4 réseaux et ainsi de suite.

II.5. Rappel sur le découpage en sous-réseaux

Pour illustrer le fonctionnement du découpage en sous-réseaux, on utilise un exemple pratique. On prend l'exemple de la classe C 192.168.1.0 dont le masque réseau est par définition 255.255.255.0. Sans découpage, le nombre d'hôtes maximum de ce réseau est de 254. Considérant qu'un domaine de diffusion unique pour 254 hôtes est trop important, on choisit de diviser l'espace d'adressage de cette adresse de classe C. On réserve 3 bits supplémentaires du 4ème octet en complétant le masque réseau. De cette façon on augmente la partie réseau de l'adresse IP et on diminue la partie hôte [15].

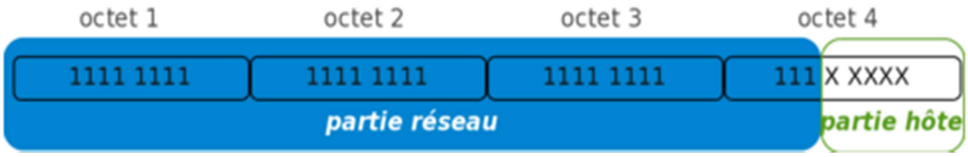


Figure II.4. *Masque réseau étendu.*

les sous-réseaux dont les bits de masque sont tous à 0 ou tous à 1 ne devaient pas être utilisés pour éviter les erreurs d'interprétation par les protocoles de routage :

- L'adresse du sous-réseau 192.168.1.0 peut être considérée comme l'adresse réseau de 2 réseaux différents : celui avec le masque de classe C (255.255.255.0) et celui avec le masque complet après découpage en sous-réseaux (255.255.255.224).
- De la même façon, l'adresse de diffusion 192.168.1.255 est la même pour 2 réseaux différents : 192.168.1.0 ou 192.168.1.224.

II.6. Translation NAT (Network Address Translation)

Le protocole IP version 4, que nous utilisons massivement actuellement, offre un champ d'adressage limité et insuffisant pour permettre à tout terminal informatique de disposer d'une adresse IP.

Une adresse IP est en effet codée sur un champ de 32 bits, ce qui offre un maximum de 232 adresses possibles, soit en théorie 4 294 967 296 terminaux raccordables au même réseau.

Pour faire face à cette pénurie d'adresses, et en attendant la version 6 du protocole IP, qui offrira un nombre d'adresses beaucoup plus important sur 128 bits, il faut recourir à un partage de connexion en utilisant la translation d'adresse, ou NAT (*Network Address Translation*).

Ce mécanisme se rencontre fréquemment à la fois en entreprise et chez les particuliers. Il distingue deux catégories d'adresses : les *adresses IP publiques*, c'est-à-dire visibles et accessibles de n'importe où (on dit aussi routables sur Internet), et les *adresses IP privées*, c'est-à-dire non routables sur Internet et adressables uniquement dans un réseau local, à l'exclusion du réseau Internet. Le NAT consiste à établir des relations entre l'adressage privé dans un réseau et l'adressage public pour se connecter à Internet.

Dans le cas d'un réseau purement privé, et jamais amené à se connecter au réseau Internet, n'importe quelle adresse IP peut être utilisée. Dès qu'un réseau privé peut être amené à se connecter sur le réseau Internet, il faut distinguer les adresses privées des adresses publiques. Pour cela, chaque classe d'adresses IP dispose d'une plage d'adresses réservées, définies comme des adresses IP privées, et donc non routables sur Internet. Ces plages d'adresses IP, sont indiqué dans la figure II.5 [13].

Classe d'adresses	Plages d'adresses privées	Masque réseau	Espace adressable
A	10.0.0.0 à 10.255.255.255	255.0.0.0	Sur 24 bits, soit 16 777 216 terminaux
B	172.16.0.0 à 172.31.255.255	255.240.0.0	Sur 20 bits, soit 1 048 576 terminaux
C	192.168.0.0. à 192.168.255.255	255.255.0.0	Sur 16 bits, soit 65 536 terminaux

Figure II.5. *Plages d'adresse privées*

Dans ce cadre, et avant d'introduire la notion de NAT, les utilisateurs qui possèdent une adresse IP privée ne peuvent communiquer que sur leur réseau local, et non sur Internet, tandis qu'avec une adresse IP publique, ils peuvent communiquer avec n'importe quel réseau IP.

La figure II.6 illustre un exemple d'adressage mixte, dans lequel on distingue les différentes communications possibles, selon un adressage de type privé ou public.

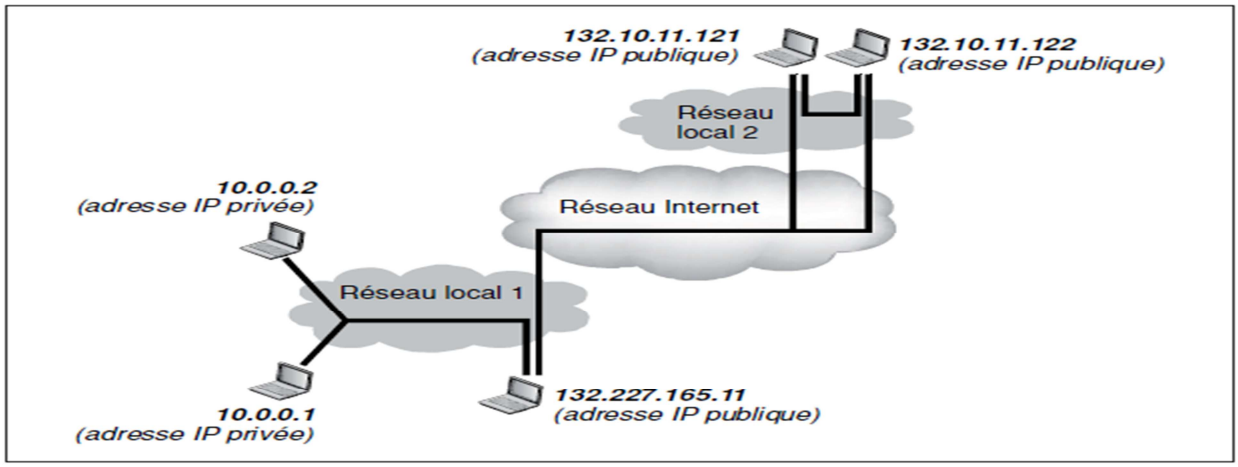


Figure II.6. *Adressage mixte privé-public*

Pour permettre à un terminal disposant d'une adresse IP privée de communiquer avec le réseau public, le processus de NAT fait intervenir une entité tierce entre un terminal, ayant une adresse IP privée, et tout autre terminal ayant une adresse IP publique. Ce mécanisme consiste à insérer un boîtier entre le réseau Internet et le réseau local afin d'effectuer la translation de l'adresse IP privée en une adresse IP publique.

La figure II.7 illustre un exemple dans lequel une passerelle NAT réalise une translation d'adresses pour quatre terminaux. Cette passerelle possède deux interfaces réseau. La première est caractérisée par une adresse IP publique (132.227.165.221). Connectée au réseau Internet, elle est reconnue et adressable normalement dans le réseau. La seconde interface est caractérisée par une adresse IP non publique (10.0.0.254). Connectée au réseau local, elle ne peut communiquer qu'avec les terminaux qui possèdent une adresse IP non publique de la même classe.

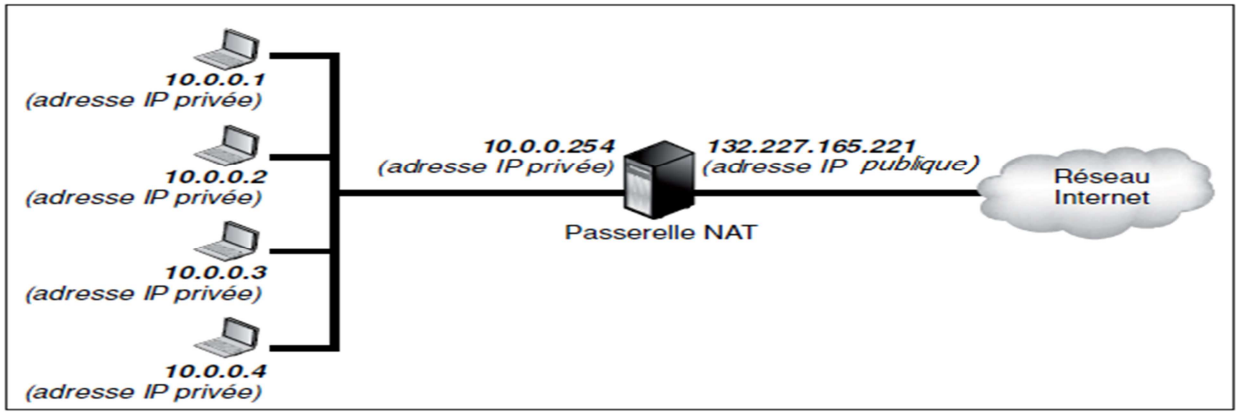


Figure II.7. *Translation d'adresse.*

Lorsqu'un terminal ayant une adresse IP privée tente de se connecter au réseau Internet, il envoie ses paquets vers la passerelle NAT. Celle-ci remplace l'adresse IP privée d'origine par sa propre adresse IP publique (132.227.165.221). On appelle cette opération une translation d'adresse. De cette manière, les terminaux avec une adresse IP privée sont reconnus et adressables dans le réseau Internet par une adresse IP publique.

La translation d'adresse est bien sûr réalisée dans les deux sens d'une communication, afin de permettre l'émission de requêtes aussi bien que la réception des réponses correspondantes. Pour cela, le boîtier NAT maintient une table de correspondance des paquets de manière à savoir à qui distribuer les paquets reçus.

II.7. Exercices

Exercice 1

Convertir l'adresse IP 134.77.88.34 en format binaire.
A quelle classe appartient cette adresse ? Quelle est l'adresse du réseau ? Quelle est l'adresse de diffusion ? Combien y a t-il d'adresse IP valides pour ce réseau ?
Convertir l'adresse 10100110.10001111.01001101.01110111 en décimal. De quelle classe est elle ?

Exercice 2

Soit l'adresse de réseau 134.56.34.0 avec un masque 255.255.255.0.
Exprimer ces 2 informations sous une forme unique en utilisant le système CIDR. Si le masque était 255.255.255.192 quelle aurait été la forme CIDR ? Quels sont les réseaux adressables dans ce cas ?

Exercice 3

Un hôte a pour adresse IP 193.222.8.98 et le masque de sous-réseau associé est 255.255.255.192.
a) Quelle est la classe du réseau? b) Quelle est l'adresse du sous-réseau? c) Quel est l'@ de diffusion (broadcast) qui permet de diffuser les datagrammes sur ce réseau? d) Il faut se connecter à un serveur d'adresse IP 193.222.8.171. Appartient-il au même sous réseau que l'adresse précédente ?

Chapitre III Routage

Le routage statique consiste à indiquer l'adresse IP des réseaux que l'on cherche à atteindre. On associe à chaque adresse, le nom de l'interface du routeur ou l'adresse IP du routeur voisin se situant sur la route vers ces réseaux de destination. Si le réseau global est complexe, la configuration peut être fastidieuse et source d'erreurs. De plus, lorsqu'un nouveau réseau est ajouté, il faut reconfigurer l'ensemble. Enfin, pour prévenir tout dysfonctionnement (panne d'un routeur, ligne coupée, etc.), il faut effectuer une surveillance permanente et reconfigurer chaque routeur le cas échéant. Si la route est rétablie, il faut recommencer la manipulation.

L'idée générale du routage dynamique est la suivante : plutôt que de centraliser la configuration du routage dans les mains d'un individu dont le temps de réaction est fatalement long et les risques d'erreurs importants, nous allons délocaliser cette tâche au niveau des routeurs. En effet, chaque appareil n'est-il pas le mieux placé pour connaître les adresses des réseaux auxquels il est directement relié puisque chacune de ses interfaces possède une adresse IP ? De plus, étant directement au contact des supports de communication, il peut établir un diagnostic sur l'état des liaisons. Fort de ces informations, il n'a plus qu'à les partager avec ses voisins. De proche en proche, les nouvelles se répandront à chaque routeur du réseau. L'intervention humaine se situera en amont dans la définition de directives et de règles à appliquer par les routeurs pour la diffusion des routes.

III.1. Rappel sur les algorithmes de routage de base

Il existe deux grandes classes de routage :

- vecteurs à distance,
- état des liaisons.

III.1.1. Vecteurs à distance

Les protocoles de ce type ont un fonctionnement assez simple [6] :

Tous les routeurs du réseau transmettent à intervalles réguliers (les Hello-Time) des mises à jour (des "updates") sur toutes leurs interfaces. Ces updates sont des paquets IP, émis en broadcast, et contenant presque la totalité de leur table de routage.

Tous les routeurs qui reçoivent ces updates en comparent le contenu avec les entrées de leur table de routage. Ils considèrent trois paramètres :

- ***la destination*** : si une destination apparaît dans un update et qu'elle n'existe pas dans leur table, ils ajoutent l'entrée à leur table. L'entrée sera constituée de l'adresse de destination indiquée, du coût de route indiqué, et de l'adresse du routeur qui a émis l'update permettant d'ajouter l'entrée. Cette adresse est en fait l'adresse IP source du paquet IP contenant l'update de référence.

- ***le coût*** : si une destination annoncée dans un update existe dans la table de routage mais avec un coût différent, le routeur procède à la mise à jour du coût.

- ***l'adresse de l'émetteur*** : si une destination existe dans la table de routage mais a été enregistrée avec une adresse de next_hop (adresse de l'émetteur de l'update) différente de l'adresse émetteur de l'actuel update, le routeur compare les coûts. Si le coût indiqué dans l'update est supérieur au coût de l'entrée existante, il ignore l'information. A l'inverse si le coût indiqué dans l'update est inférieur, il remplace l'entrée existante dans la table de routage par l'information de l'update.

Les protocoles à vecteurs de distance présentent deux inconvénients majeurs :

- La charge réseau induite par les updates. En effet, chaque routeur émet régulièrement sur le réseau l'ensemble de sa table de routage. Ce trafic induit fini par ne pas être négligeable, notamment dans le cas de grands réseaux.

- Le protocole n'est pas très réactif. On parle de "latence" ou de "délai de convergence" longs.

Il a donc été nécessaire d'imaginer un autre type de fonctionnement.

III.1.2. Etats de liens

Le principe est le suivant [6, 11]:

Chaque routeur transmet, comme précédemment, des updates à ses voisins (les routeurs voisins, pas les machines IP bien sûr !).

Ces mises à jour sont placées dans des paquets IP émis en multicast. Selon la nature des informations émises l'adresse de multicast peut être différente, elle permet de s'adresser à des routeurs ayant des rôles particuliers dans le réseau. Ces updates indiquent non plus des destinations associées à des coûts, mais des descriptions de liens (des link states). Comme chaque lien est forcément associé à une adresse IP réseau il est identifié par cette adresse. Il est ensuite décrit par sa nature (LAN, Point à point, Multipoint), son coût d'accès (notion très subjective, variant d'un protocole à l'autre), son état (actif, down, shut, etc.), etc. Enfin le lien est également associé à un identifiant de routeur qui lui donne accès (le routeur qui est branché sur le lien).

Ces informations sont stockées dans une base de données du routeur. Le routeur va déterminer quelles suites successives de liens et de routeurs sont possibles pour atteindre une destination, c'est le calcul de routes. Si pour une même destination il y a plusieurs routes possibles, il validera, dans sa table de routage, celle de coût le plus faible. Le coût d'une route correspond à la somme des coûts des liens la composant.

A chaque update, le routeur examine les changements pour chaque lien par rapport aux précédentes mises à jour. S'il n'y a pas de changement il ne recalcule pas les routes et n'émets pas de mises à jour à ses voisins. Par contre s'il y a un changement, il ne relaie vers ses voisins que l'information ayant changée.

Donc les différences fondamentales avec un protocole à vecteurs de distances sont :

- Les routeurs n'émettent des updates que lorsque des liens changent d'états (coûts, indisponibilité, etc.)
- les updates émis ne contiennent que des descriptions de liens ayant changé d'état, ils sont donc moins volumineux
- Grâce aux updates d'états de liens, les routeurs stockent dans leurs bases de données une véritable carte topologique du réseau et de son état
- Les routeurs retransmettent immédiatement les mises à jour à leurs voisins, ils compilent leur table de routage après, le protocole est donc beaucoup plus réactif !

Malheureusement, il y au moins trois inconvénients à l'utilisation de ce type de protocole :

- Il faut des routeurs possédant des performances CPU et des capacités mémoires supérieures à celles nécessaires pour les vecteurs de distance.

- Ce type de protocole très réactif nécessite des réglages précis des timers de transmission car sa réactivité peut justement nuire à la stabilité du routage (si on change toutes les tables pour une micro-coupure de liens, par exemple !).

- Enfin, Ces protocoles sont assez complexes à mettre en œuvre.

III.1.3. Quels protocoles pour quelle famille ?[6]

A. Les protocoles à vecteurs de distances

- ***RIP Routing Information Protocol (version 1 & 2)*** : Le plus ancien, le plus simple mais encore très utilisé !

- ***IGRP Interior Gateway Routing Protocol*** : Protocole propriétaire Cisco, mais comme Cisco est très implanté dans le monde du réseau, on le trouve très souvent.

- ***E-IGRP Extended IGRP*** : Apparut avec la version logicielle 11 de Cisco, il supplante peu à peu l'IGRP. Il offre des fonctions plus élaborées.

B. Les protocoles à états de liens

- ***OSPF Open Shortest Path First*** : très complet, très performant, très élaboré, mais très compliqué et consommateur de ressources dans les routeurs. On le réserve généralement à de grands réseaux.

III.2. Routage à vecteurs de distance, exemple : le protocole RIP

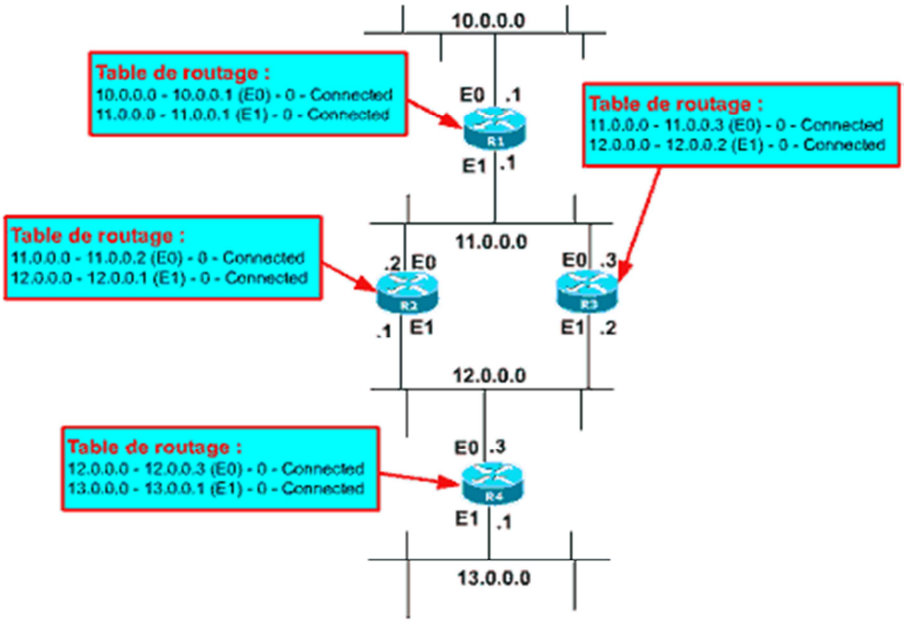
RIP est un protocole à vecteurs de distances, il émet la totalité de la table de routage des routeurs toutes les 30 secondes (Hello-time). Le coût RIP est un nombre de sauts. Au-delà de 15 sauts la route est invalide. La meilleure route est celle présentant le moins de sauts (le coût le plus faible) [6, 11].

A. Exemple sur le fonctionnement de RIP

Phase 1 : Programmer le réseau

Soit le réseau ci-contre composé des routeurs R1, R2, R3 et R4 interconnectant les réseaux IP 10.0.0.0 à 13.0.0.0. Après avoir installé les routeurs sur site, les avoir mis sous tension et avoir raccordé les câbles réseaux, il est temps de les programmer !

Vous allez définir dans un premier temps, sur chacune des interfaces les adresses IP et les masques associés. Si vous ne programmez rien d'autre et si vous examinez les tables de routage vous obtenez le contenu du schéma ci-dessous.

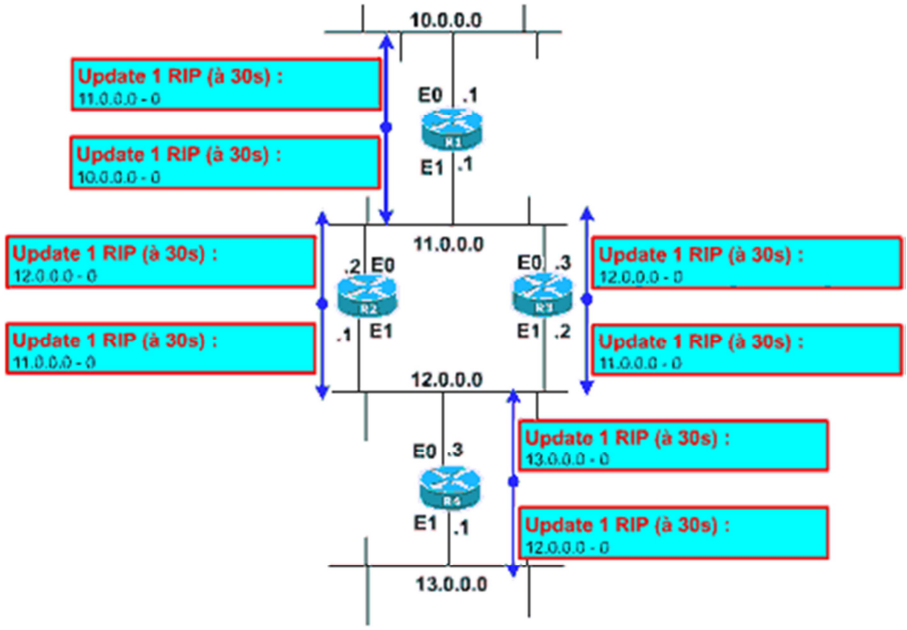


Phase 1 : Définition du plan d'adressage

Vous remarquez que **les routeurs connaissent les réseaux qui leurs sont directement raccordés** mais rien d'autre ! Notamment R2 et R3 ignorent la présence des réseaux 10.0.0.0 et 13.0.0.0, de la même manière que R1 ignore l'existence des réseaux 12.0.0.0 et 13.0.0.0, et R4 ignore les réseaux 10.0.0.0 et 11.0.0.0. Si des paquets leurs étaient envoyés avec ces adresses destinations, ils ne sauraient pas quoi en faire et les jetteraient !

Phase 2 : Activation de RIP

Pour remédier au problème vous accédez de nouveau à l'administration de chaque routeur et vous activez le process RIP. Immédiatement, ou au plus tard dans les 30 secondes, chaque routeur émet un premier update. Bien sûr les updates ne sont pas forcément émis en même temps par tous les routeurs. Le schéma ci-dessous présente le contenu de ces updates.



Phase 2 : Première émission d'update RIP

Vous remarquez :

- que les updates contiennent la table de routage de chaque routeur. L'update est formé d'entrées (*lignes*). Chaque entrée est constituée de l'adresse réseau annoncée (*la destination*) et d'un coût associé. Ce coût est celui indiqué dans la table de routage (*ici tous les coûts sont à 0, puisque les réseaux annoncés sont tous directement raccordés aux routeurs émettant les updates*). Le coût pour RIP correspond à un nombre de routeurs à traverser.
- que les updates sont émis sur toutes les interfaces de chaque routeur
- que le numéro du réseau sur lequel est émis l'update n'est pas inscrit dans l'update. Par exemple R1 n'annonce pas le réseau 11.0.0.0 dans l'update émis sur 11.0.0.0. En effet, les routeurs R2 et R3 n'ont pas besoin de cette information puisqu'ils ont également les pieds dans le réseau 11.0.0.0.

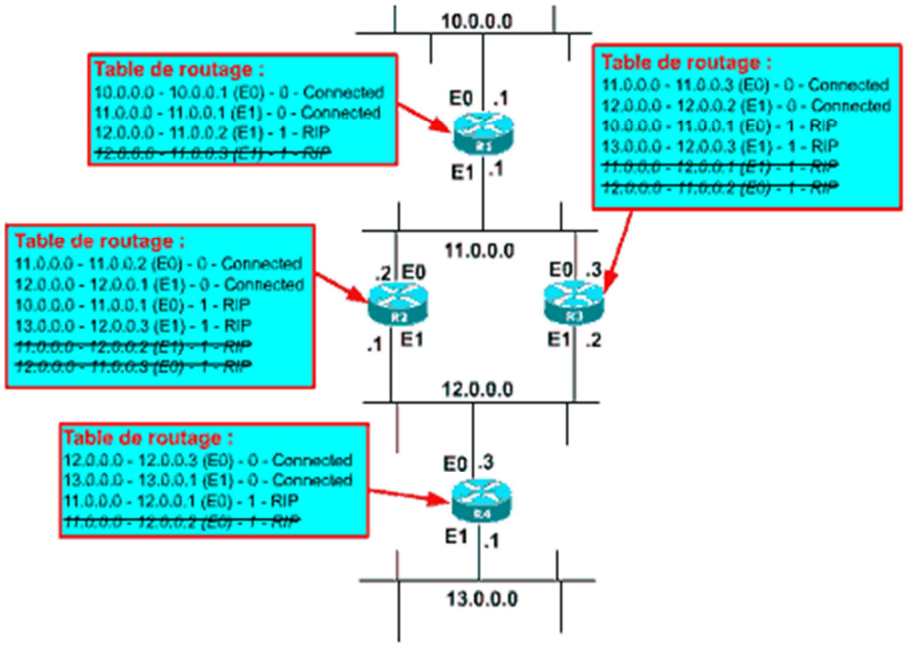
Phase 3 : Première initialisation des tables de routage

Chaque routeur examine les updates reçus. Ils incrémentent systématiquement de 1 les coûts annoncés puisqu'ils considèrent que les routeurs émettant les updates devront être franchi pour atteindre les destinations indiquées.

Tous les routeurs ajoutent dans leurs tables les destinations qui ne leurs sont pas encore connues. Par contre, si une entrée est déjà connue, le routeur compare le coût annoncé et l'adresse de l'émetteur de l'update **:**

- si le coût de l'update est inférieur à celui de la table de routage, l'entrée de l'update remplace celle de la table
- si le coût est supérieur ou identique, l'update est ignoré

Le schéma ci-dessous illustre le contenu des tables de routage après traitement du premier update par les routeurs.



Phase 3 : Première mise à jour des tables de routage

Vous remarquerez que R1 a reçu deux annonces pour 12.0.0.0, une de la part de R2 et une autre de R3. L'entrée de R3 a été ignorée par R1 (*barrée dans le schéma*) car son coût n'était pas meilleur que l'entrée annoncée par R2 qui a été traitée avant (*mais cela aurait pu être l'inverse !*). Il en est de même pour R4 vis à vis de l'annonce simultanée de 11.0.0.0 par R2 et R3.

A noter également que R2 reçoit, par son interface E1, une annonce de R3 pour le réseau 11.0.0.0, bien sûr R2 l'ignore (*barrée dans le schéma*) puisqu'il a accès à ce réseau directement par son interface E0 ! C'est d'ailleurs la même chose pour R3 à qui R2 annonce 11.0.0.0 !

Chaque entrée de la table de routage qui est initialisée suite à réception d'un update est construite telle que suit :

Destination - adresse de l'émetteur de l'update - coût de la route - protocole qui a annoncé la route

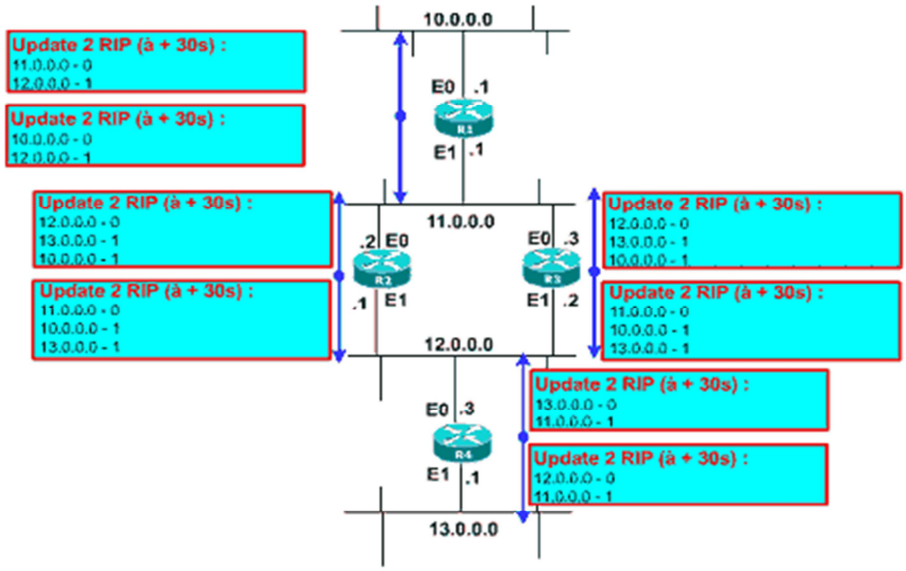
Vous remarquerez enfin que R1 ne connaît toujours pas le réseau 13.0.0.0 et que R4 ne connaît pas 10.0.0.0. Il va falloir attendre le prochain update !

Phase 4 : Emission du deuxième update

Comme précédemment, les routeurs émettent leurs updates contenant la totalité de leur table de routage, sur toutes leurs interfaces.

Pour l'anecdote, vous remarquerez que l'update émis par R1 sur le réseau 10.0.0.0 est totalement inutile puisqu'il n'y a aucune machine pour interpréter son annonce.

Vous remarquerez que l'update émis sur le réseau 11.0.0.0 par R1 contient le réseau 12.0.0.0. Ceci suppose que R1 peut donner accès à 12.0.0.0, ce qui est totalement faux ! 12.0.0.0 n'est accessible que par R2 et R3 !

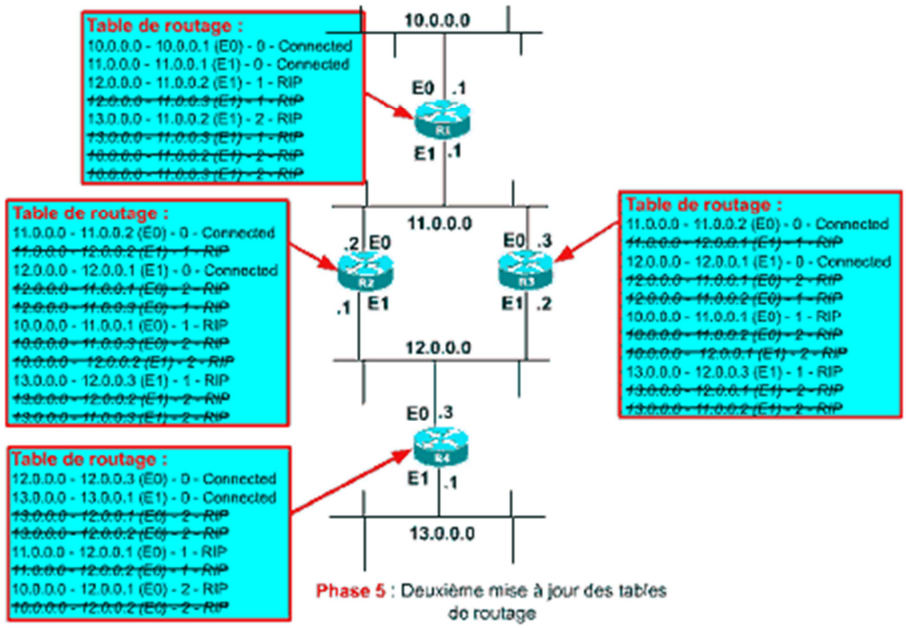


Phase 4 : Deuxième émission d'update RIP

Comme précédemment les réseaux 11.0.0.0 et 12.0.0.0 sont annoncés l'un à l'autre par R2 et R3 ce qui est inutile !

Phase 5 : Deuxième mise à jour de table

Vous remarquerez dans le schéma suivant, que chaque routeur ajoute les entrées qui lui sont inconnues et ignore les entrées présentant un coût supérieur ou égal. C'est finalement très simple !



Phase 5 : Deuxième mise à jour des tables de routage

Au final, après ce traitement :

- les tables de R2 et R3 n'ont pas été modifiée, elles avaient déjà toutes les informations nécessaires
- R1 a appris la route pour 13.0.0.0 (via R2) et R4 a appris la route pour 10.0.0.0 (via R2 également).
- tous les réseaux sont connus par tous les routeurs

B. Limites du protocole RIP

- La limite des 15 routeurs maximum sur une route (ne s'implémente pas sur de grand réseaux), le délai de convergence long, l'émission des tables de routages complètes sur le réseau toutes les 30 secondes.

- **des notions de coûts simplistes** (nombre de sauts pour RIP) qui avaient pour conséquence de limiter la taille des routes possibles et d'interdire l'optimisation du routage. RIP préfère ainsi une route composée de 2 routeurs séparés par une liaison à 64 Kbps à une route de 3 routeurs séparés par deux liaisons 34 Mbps ! Pourtant la seconde route est sans contexte, la plus rapide et la moins sujette à la congestion !

III.3. Routage à Etas de liens, exemple : le protocole OSPF

Résumons ici les principales caractéristiques d'OSPF [6] :

- C'est un protocole ouvert (non propriétaire) et standardisé par la RFC 1247. Il est le fruit d'un groupe de travail de l'IETF auquel a d'ailleurs largement contribué DIGITAL.
- Il autorise le routage par type de service (champ TOS du datagramme IP).
- Il assure automatiquement un équilibrage de charge (load balancing), si plusieurs routes de même coût sont définies vers un même réseau pour un même TOS.
- Il permet de diviser le réseau en zones, toutes autonomes, et invisibles pour les zones voisines. OSPF implémente une notion d'aires. Vous divisez votre réseau en plusieurs zones correspondantes à des aires (par exemple, l'aire Sud-Ouest, l'aire Nord-Est, etc....). Les aires sont interconnectées entre-elles par une aire spécifique appelée "l'aire backbone". Ceci permet d'optimiser le routage en annonçant des résumés de routes. Cette hiérarchisation va plus loin en permettant d'indiquer des natures de réseaux IP différentes, selon que le réseau supporte un seul routeur ou plusieurs routeurs (notion de stub). Chaque routeur maintient une base de données (la LSDB : Link State DataBase) indiquant la topologie de sa propre zone.
- Il est capable de gérer les routes de machine à machine, les routes vers les réseaux, ainsi que le routage vers les sous-réseaux.
- Les échanges entre routeurs sont authentifiés, ce qui permet de s'affranchir des problèmes de malveillance.
- Sur un réseau à accès multiples, il y a possibilité d'avoir une seule passerelle diffusant les messages d'état de lien (notion de Directory Router et Backup Directory Router) aux autres routeurs, ainsi les média sont moins chargés par des messages de protocole de routage.
- Utilise l'adressage multicast IP. Celui-ci est moins polluant que le broadcast, dans le sens ou seules les machines implémentant OSPF analyse le contenu des paquets IP émis en multicast (aux adresses correspondantes au trafic OSPF).
- Il réagit rapidement aux changements de topologie sans générer trop de trafic. OSPF fonctionne très différemment de RIP. Lorsqu'un lien change d'états (liaison d'interconnexion devenant indisponible ou inversement), OSPF émets des updates (LSA : Link State Advertisement) à tous les routeurs voisins en "multicast IP". Les updates ne décrivent que le changement d'état du lien et pas l'ensemble de la table de routage du routeur ! Tous les routeurs retransmettent immédiatement ces mises à jours à tous leur voisins également. Tout le réseau est ainsi immédiatement informé du changement d'état d'un lien ! Chaque routeur peut modifier sa table de routage en conséquence.
- Un coût de route plus souple : OSPF autorise le paramétrage des coûts sur les interfaces sortantes. Ceci vous permet donc de tracer vos routes aussi sûrement que si vous aviez mis en place un routage statique, mais en conservant l'intérêt du routage dynamique (la reconfiguration automatique du routage). Par défaut, si vous n'appliquez pas de coût spécifique, le coût d'OSPF est calculé par rapport à la bande passante du support (nous y reviendrons plus tard !).
- Il autorise l'utilisation des masques de sous-réseaux de longueur variable (VLMS : Variable Length Mask Subnet)

III.4. Protocole IPv6

IPv4 a été initialement conçu pour un réseau comprenant une centaine d'ordinateurs. Avec le web, Internet a vu son utilisation augmenter de manière exponentielle si bien que la saturation du réseau a été initialement prévue pour 1994. Des solutions comme le NAT et le CIDR ont permis de ralentir considérablement l'explosion de la taille des tables de routages d'Internet et ont permis de mettre au point un nouveau protocole Internet appelé Ipv6.
IPv6 est conçu pour remédier aux limitations d'IPv4 et en même temps répondre à de nouvelles exigences techniques apparues au niveau des réseaux ou des applications [16].
Voici ce à quoi ressemble un datagramme IPv6 :

Voici la signification des différents champs :

- **Le champ Version** est toujours égal à 4 bits pour IPv6. Pendant la période de transition de IPv4 vers IPv6, les routeurs devront examiner ce champ pour savoir quel type de datagramme ils routent.

- **Le champ Classe de trafic** (codé sur 8 bits) est utilisé pour distinguer les sources qui doivent bénéficier du contrôle de flux des autres. Des priorités de 0 à 7 sont affectées aux sources capables de ralentir leur débit en cas de congestion. Les valeurs 8 à 15 sont assignées au trafic temps réel (les données audio et vidéo en font partie) dont le débit est constant. Cette distinction des flux permet aux routeurs de mieux réagir en cas de congestion. Dans chaque groupe prioritaire, le niveau de priorité le plus faible correspond aux datagrammes les moins importants.

- **Le champ Identificateur de flux** contient un numéro unique choisi par la source qui a pour but de faciliter le travail des routeurs et de permettre la mise en oeuvre les fonctions de qualité de services comme RSVP (*Resource reSerVation setup Protocol*). Cet indicateur peut être considéré comme une marque pour un contexte dans le routeur. Le routeur peut alors faire un traitement particulier : choix d'une route, traitement en "temps-réel" de l'information, ...
  Le champ identificateur de flux peut être rempli avec une valeur aléatoire qui servira à référencer le contexte. La source gardera cette valeur pour tous les paquets qu'elle émettra pour cette application et cette destination. Le traitement est optimisé puisque le routeur n'a plus à consulter que cinq champs pour déterminer l'appartenance d'un paquet. De plus, si une extension de confidentialité est utilisée, les informations concernant les numéros de port sont masquées aux routeurs intermédiaires.

- **Le champ Longueur des données utiles** (en anglais *payload*) sur deux octets, ne contient que la taille des données utiles, sans prendre en compte la longueur de l'en-tête. Pour des paquets dont la taille des données serait supérieure à 65536 ce champ vaut 0 et l'option jumbogramme de l'extension de "proche en proche" est utilisée.
- **Le champ En-tête suivant** a une fonction similaire au champ *protocole* du paquet IPv4 : Il identifie tout simplement le prochain en-tête (dans le même datagramme IPv6). Il peut s'agir d'un protocole (de niveau supérieur ICMP, UDP, TCP, ...) ou d'une extension.

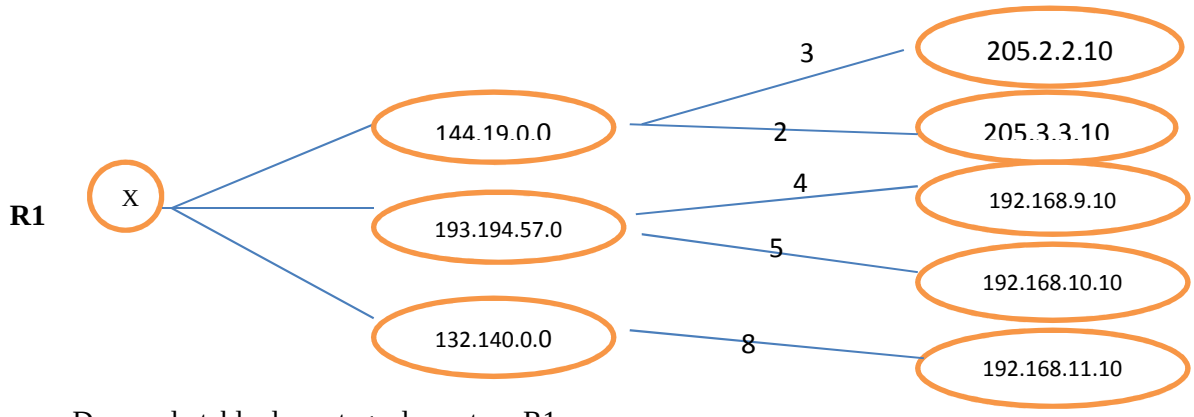
- **Le champ Nombre de sauts** remplace le champ *"TTL"* (*Time-to-Live*) en IPv4. Sa valeur (sur 8 bits) est décrémentée à chaque noeud traversé. Si cette valeur atteint 0 alors que le paquet IPv6 traverse un routeur, il sera rejeté avec l'émission d'un message ICMPv6 d'erreur. Il est utilisé pour empêcher les datagrammes de circuler indéfiniment. Il joue le même rôle que le champ *Durée de vie* d'IPv4, à savoir qu'il contient une valeur représentant le nombre de sauts ou de pas (hops) qui est décrémenté à chaque passage dans un routeur. En théorie, dans IPv4, il y a une notion de temps en seconde mais aucun routeur ne l'utilisant comme tel, le nom a changé pour refléter l'usage actuel.
- Viennent ensuite les champs **Adresse source** et **Adresse de destination**.

III.5. Exercices

Questions de cours

- Quelles sont les deux grandes classes de routage ?
- Expliquez les princes de la technique Hold down.
- Quel est le champ du datagramme IP qui évite qu'un datagramme ne circule indéfiniment dans le réseau.

Exercice 1



Donnez la table de routage du routeur R1.

Exercice 2

Dans le protocole OSPF indiquez :

- les causes de déclenchement des mises à jour.
- La façon dont la fiabilité est assurée.

Dans le protocole Rip indiquez :

- Les causes de déclenchement des mises à jour.
- Ses Faiblesses et comment y remédier.

Exercice 3

Simplifiez les adresses suivantes :

- fe80 :0000 :0000 :0000 :1000 :4cff :fe4f :4f50
- 2011 :0000 :0000 :1f80 :0000 :0000 :0000 :0fc5

Chapitre IV Couche transport

IV.1. Rôles de la couche Transport

La couche Transport remplit plusieurs rôles destinés à assurer correctement la transmission des informations [5, 8] :

IV.1.1 Suivi des conversations

Tout hôte peut héberger plusieurs applications qui communiquent sur le réseau. Chacune de ces applications communique avec une ou plusieurs applications hébergées sur des hôtes distants. Il incombe à la couche transport de gérer les nombreux flux de communication entre ces applications.

IV.1.2. Identification des applications

Afin d'identifier les applications fonctionnant sur une même machine, on affecte à chacune de ces applications un numéro unique : c'est le n° de port.
Il est choisi parmi ceux non utilisés, entre 1024 et 65535 (les 1024 premiers sont réservés).

L'ensemble Adresse IP + n° de port est appelé : SOCKET

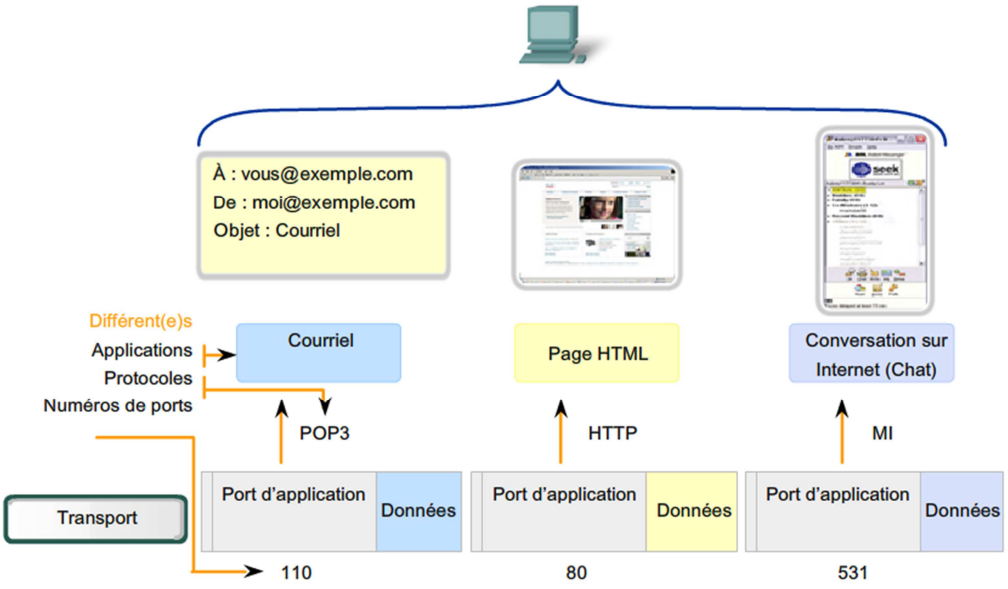


Figure IV.1. *Identification des applications.*

IV.1.3. Segmentation des données

La segmentation des données permet d'envoyer et de recevoir des données tout en exécutant plusieurs applications simultanément sur un ordinateur. En l'absence de segmentation, une seule application, par exemple la lecture vidéo en continu, pourrait recevoir des données. Il serait impossible de recevoir des courriels, de parler sur une messagerie instantanée ou d'afficher des pages Web tout en visualisant la vidéo.

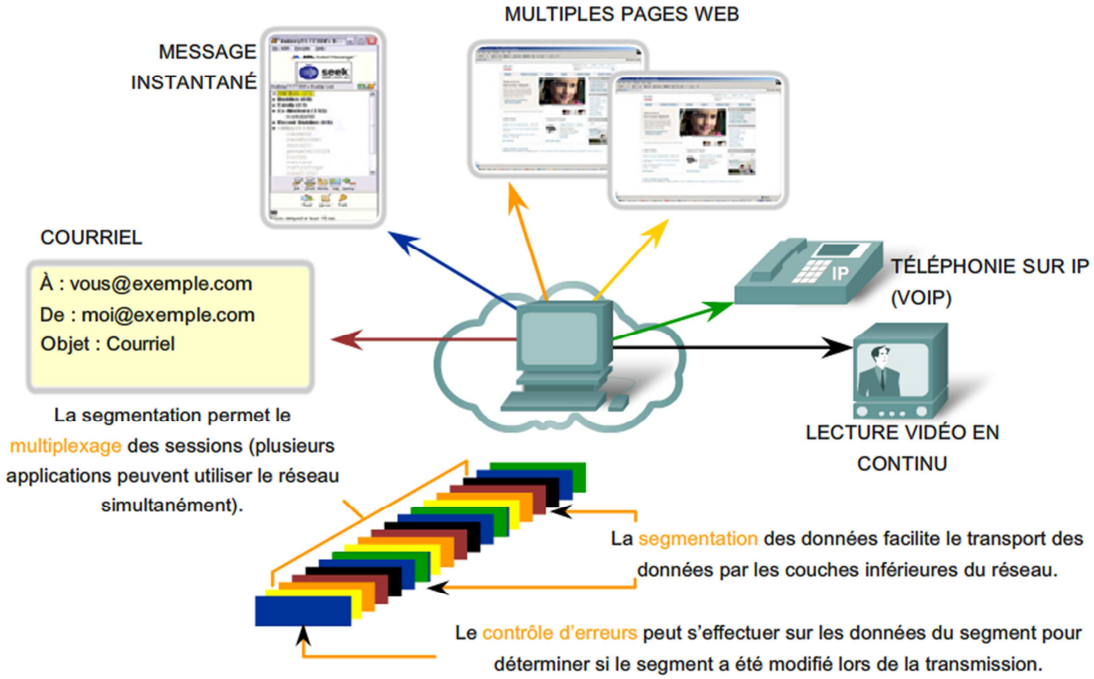


Figure IV.2. *Segmentation des données.*

IV.1.4. Reconstitution des fragments

Quand des services envoient des données à l'aide du protocole TCP, il arrive que les segments parviennent à destination dans le désordre. Pour que le destinataire puisse comprendre le message d'origine, il faut que les données contenues dans ces segments soient ré-agencées dans leur ordre d'origine. Pour cela, des numéros sont affectés à l'en-tête de chaque paquet.

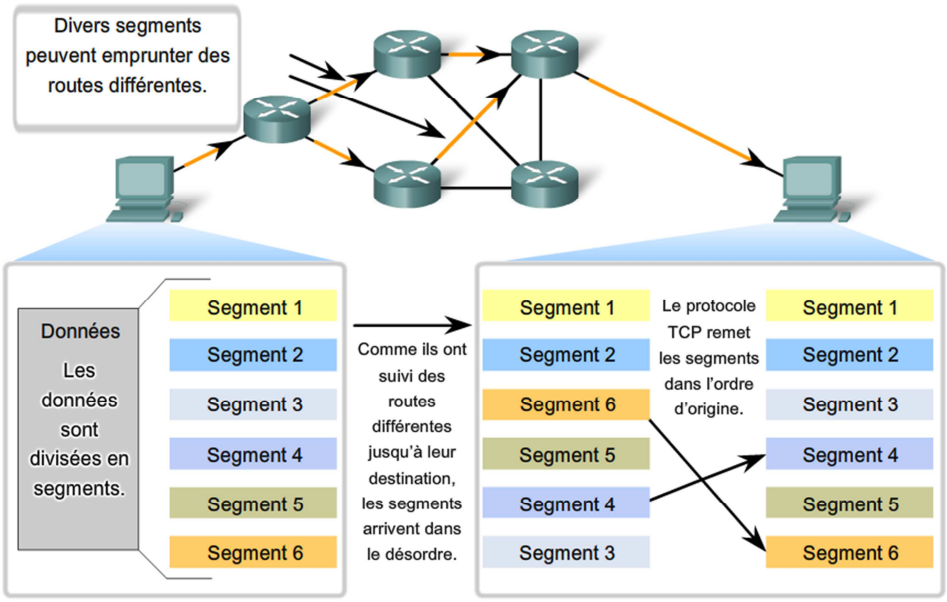


Figure IV.3. *Reconstitution des fragments.*

IV.1.5. Fiabilité et contrôle de flux

Bien des circonstances peuvent entraîner la corruption ou la perte d'un bloc de données lors de son transfert sur le réseau. La couche transport veille à ce que tous les blocs atteignent leur destination en demandant au périphérique source de retransmettre les données qui ont pu se perdre.

Les hôtes du réseau disposent de ressources limitées, par exemple en ce qui concerne la mémoire ou la bande passante. Quand la couche transport détermine que ces ressources sont surexploitées, certains protocoles peuvent demander à l'application qui envoie les données d'en réduire le flux.

IV.2. Protocoles de la couche Transport

Au niveau de la couche transport, les trois opérations de base en matière de fiabilité consistent à [5, 9] :

- Effectuer le suivi des données transmises ;
- Accuser réception des données ;
- Retransmettre toute donnée n'ayant pas fait l'objet d'un reçu.

Ce suivi des informations engendre un trafic plus ou moins important sur le réseau, ce qui peut compromettre sa stabilité.
Certaines applications, comme les bases de données, les pages Web et les courriels, ont besoin que toutes les données envoyées arrivent à destination dans leur état d'origine afin que ces données soient utilisables.
A l'inverse, d'autres applications tolèrent mieux la perte de petites quantités de données. Si, par exemple, un ou deux segments d'une lecture vidéo n'arrivent pas, cela ne fera que créer une interruption momentanée du flux. Ceci pourra se traduire par une distorsion de l'image que l'utilisateur ne remarquera peut-être même pas.

La couche Transport propose donc deux niveaux de fiabilité. Le plus élevé effectue un suivi total des connexions alors que le plus faible n'effectue pas de contrôles.
A chaque niveau de fiabilité évoqué ci-dessus correspond un protocole spécifique.

A. Protocole TCP (Transmission Control Protocol)

Le protocole TCP est un protocole avec connexion. Le protocole TCP impose une surcharge pour accroître les fonctionnalités. Le protocole TCP spécifie d'autres fonctions, à savoir la livraison dans l'ordre, l'acheminement fiable et le contrôle du flux. Les blocs sont appelés Segments. Le protocole TCP est utilisé notamment par des applications de :

- Navigateurs Web
- Courriel.
- Transfert de fichiers

En-tête TCP

La figure ci-dessous présente l'en-tête ajouté par le protocole TCP lors de l'encapsulation des données de la couche supérieure.

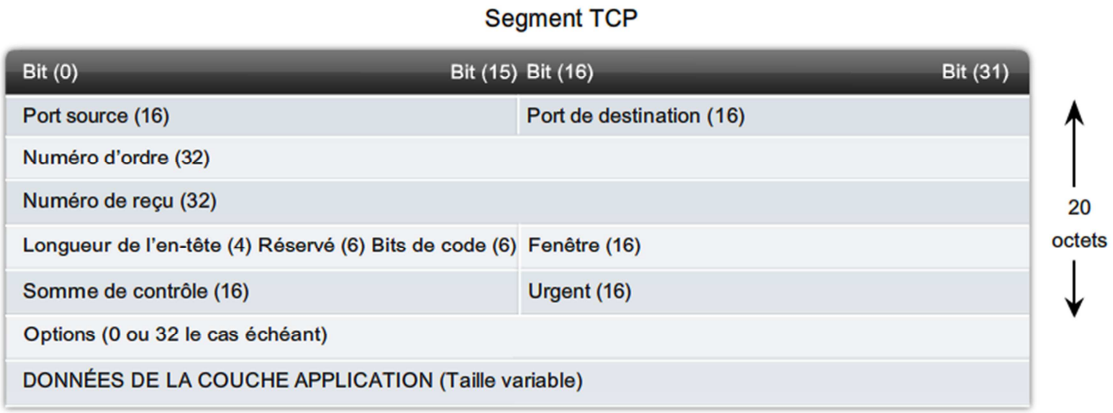


Figure IV.4. *En-tête TCP*

On y trouve notamment les informations suivantes :

- ***N° des ports Source et Destination :*** Ces numéros permettent d'identifier l'application utilisée sur le client et le serveur.
- ***N° d'ordre et n° de reçu :*** Ces numéros permettent d'identifier les segments destinés à une application. Le n° d'ordre est le n° affecté à un segment (SEQ). Le n° de reçu est le n° du prochain segment attendu (ACK).
- ***Longueur de l'en-tête :*** Indique la longueur de l'entête du segment en octet.
- ***Bits de code, aussi appelés Flags ou Drapeaux :*** Ces bits permettent de savoir par exemple s'il s'agit d'une demande de connexion, d'un accusé de bonne réception, d'une demande de réémission, …
- ***Fenêtre :*** Ce nombre indique la quantité de données qui peut être reçue en un seul segment.
- ***Somme de contrôle :*** Utilisé pour contrôler les erreurs d'entête de données.
- ***Pointeur d'urgence*** : Uniquement utilisé avec un indicateur URG (urgent).
- ***Options :*** Informations facultatifs.
- ***Données :*** Données d'application.

Suivi d'une connexion par TCP

L'une des fonctions de TCP est de veiller à ce que chaque segment atteigne sa destination. Les services TCP sur l'hôte de destination accusent réception des données reçues à l'application source.
Nous avons besoin de 3 informations :

- Le n° du segment : **SEQ**
- Le n° d'acquittement : **ACK**
- La taille des données transmises : **Lenght**

Ouverture de connexion

Lorsque deux hôtes communiquent à l'aide de TCP, une connexion est établie avant que les données ne puissent être échangées. Une fois la communication terminée, les sessions sont fermées et il est mis fin à la connexion.

Pour établir une connexion fiable, TCP recourt à un mécanisme, connu sous le nom de « three-way handshake » (3 temps)

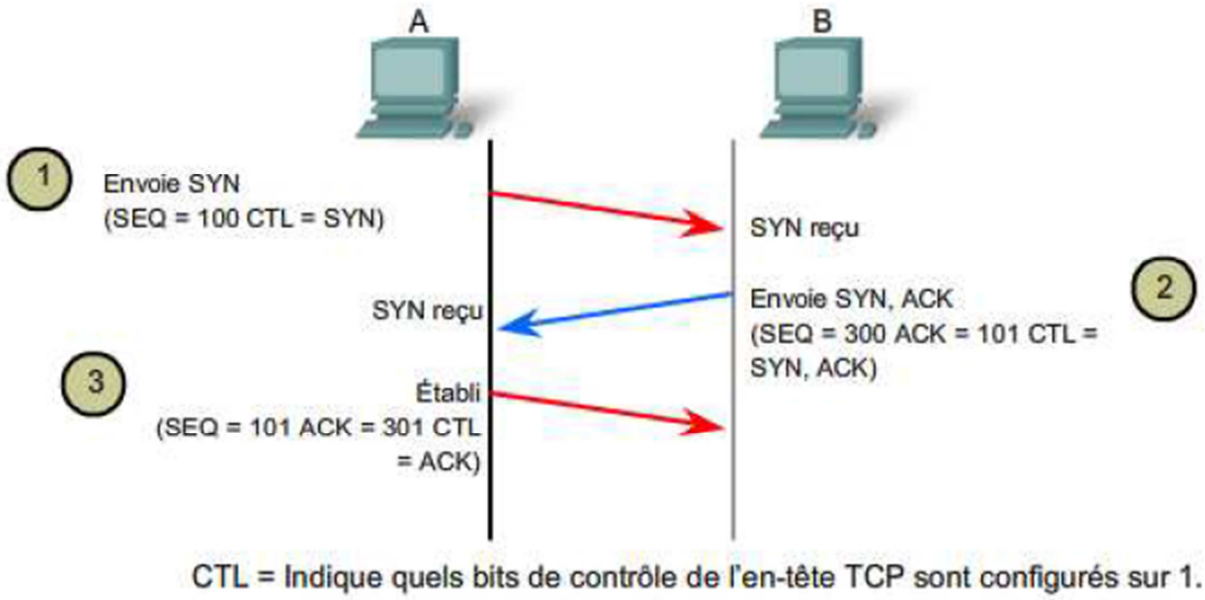


Figure IV.5. *Etablissement d'une connexion TCP.*

Fenêtre glissante

En réalité, chaque segment n'est pas acquitté. En effet, cela impose que l'émetteur attendre une confirmation après chaque envoi, ce qui se traduit par une communication lente et une surcharge du réseau. La solution est d'envoyer plusieurs segments avant qu'un ACK ne soir reçu. C'est le pincipe de la fenêtre glissante.

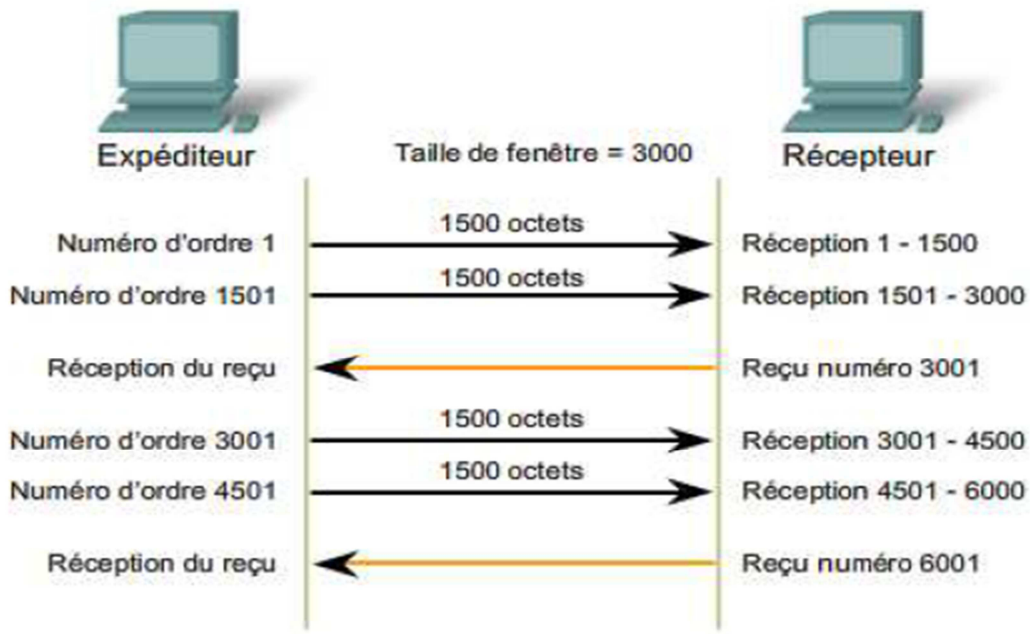


Figure IV.6. *Fenêtre glissante.*

Retransmission des informations

Lorsqu'un segment s'est perdu, il faut que l'émetteur en soit informé afin de le réémetrre. Cela se produit si le récepteur n'accuse pas réception dans un délai imparti.
Toutefois, si le segment n'est pas perdu et qu'il arrive tout de même à destination, la machine réceptrice saura grâce au numéro d'ordre qu'il s'agit d'un doublon et ne conservera que le dernier segment arrivé à destination...
La fenêtre verra alors sa taille diminuer. Quand un certain temps se sera écoulé sans perte de données ni contraintes excessives sur les ressources, le destinataire pourra recommencer à augmenter la taille de fenêtre.

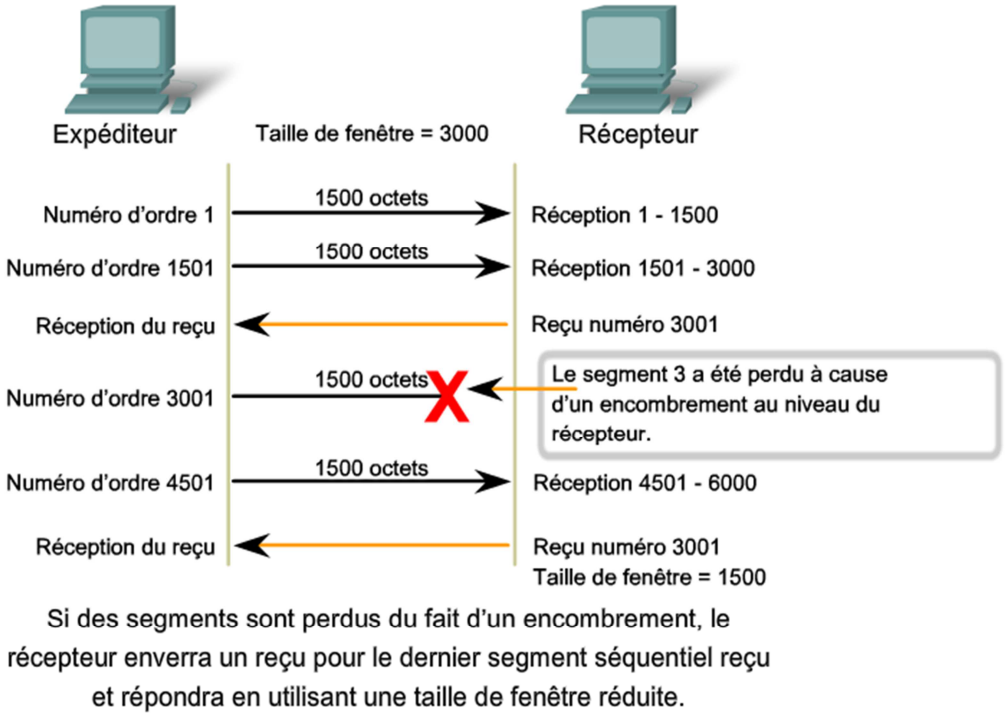


Figure IV.7. *Retransmission des informations.*

Fermeture de connexion.

De la même manière, chaque fin de connexion se fait de manière "courtoise", chacun prévenant l'autre de la fermeture de la connexion.

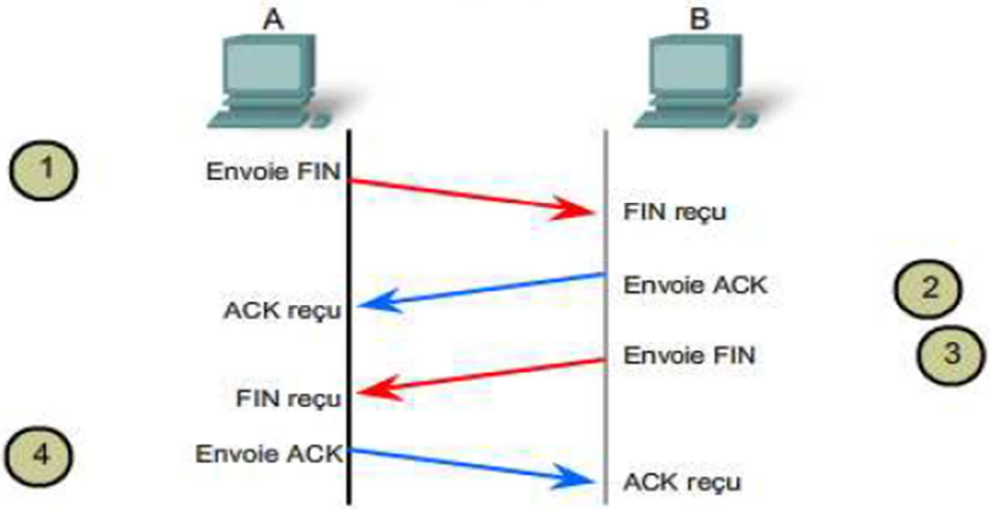


Figure IV.8. *Fermeture de connexion.*

B. Protocole UDP (User Datagram Protocol)

Le protocole UDP est un protocole simple offrant des fonctions de couche transport de base. Il crée beaucoup moins de surcharge que le protocole TCP car il est sans connexion et ne propose pas de mécanismes sophistiqués de retransmission, de séquençage et de contrôle de flux. Les blocs de communications utilisés dans le protocole UDP sont appelés des datagrammes.

Mais cela ne signifie pas que les applications utilisant le protocole UDP manquent toujours de fiabilité. Cela signifie simplement que ces fonctions ne sont pas fournies par le protocole de la couche transport et qu'elles doivent être implémentées à un autre niveau, le cas échéant.

Bien que le volume total de trafic UDP trouvé sur un réseau typique soit relativement faible, des protocoles importants de la couche application utilisent le protocole UDP, notamment :

- DNS (Domain Name System)
- SNMP (Simple Network Management Protocol)
- DHCP (Dynamic Host Configuration Protocol)
- RIP (Routing Information Protocol)
- TFTP (Trivial File Transfer Protocol)
- Jeux en ligne

Le protocole UDP se contente donc de réassembler les données dans l'ordre dans lequel elles ont été reçues, puis de les transmettre à l'application. Si l'application attache une grande importance à l'ordre des données, elle devra identifier l'ordre correct des données et déterminer leur mode de traitement.

En-tête UDP.

IV.3. Exercices

Questions de cours

- Quels sont les services de base de la couche transport ?
- Expliquez le principe de fenêtrage.
- A quoi sert le Checksum (CRC) du segment TCP.
- Quelle est la différence entre le mode connecté et le mode non connecté.
- Dans quel mode fonctionne le protocole UDP.

Exercice 1

Soit deux hôtes qui communiquent en utilisant le protocole TCP.

- Décrire le processus de communication, en cas de transfert d'un fichier de taille 1024 Ko de la machine A vers la machine B.

- Le buffer est de taille 128 Ko.

Chapitre V Applications réseau

V.1. Connexion à distance

La connexion à distance permet une utilisation interactive de machines distantes grâce à un protocole en mode connecté tel que TCP. Elle fonctionne selon le modèle client-serveur; c'est à dire qu'un utilisateur active un programme client qui établit une connexion TCP entre la machine locale et le serveur.
Le but de la connexion à distance est d'avoir accès à toutes les commandes disponibles sur le système distant [3, 10].

V.1.1. TELNET [3]

TELNET est un protocole de connexion à distance de la famille des protocoles TCP/IP.
TELNET permet d'établir depuis sa machine une connexion TCP avec le serveur d'une autre machine. Il transmet ensuite les caractères frappés par l'utilisateur à la machine distante et il achemine les réponses du serveur.
C'est un protocole transparent car il donne l'illusion que le terminal de l'utilisateur est directement relié à la machine distante.
TELNET offre 3 services de base. En premier lieu, il définit un "terminal de réseau virtuel" (NVT Network Virtual Terminal) qui offre une interface normalisée aux systèmes distants.
Ensuite, il offre un mécanisme de négociation d'options entre le client et le serveur. Enfin, il permet une symétrie des 2 extrémités de la connexion.

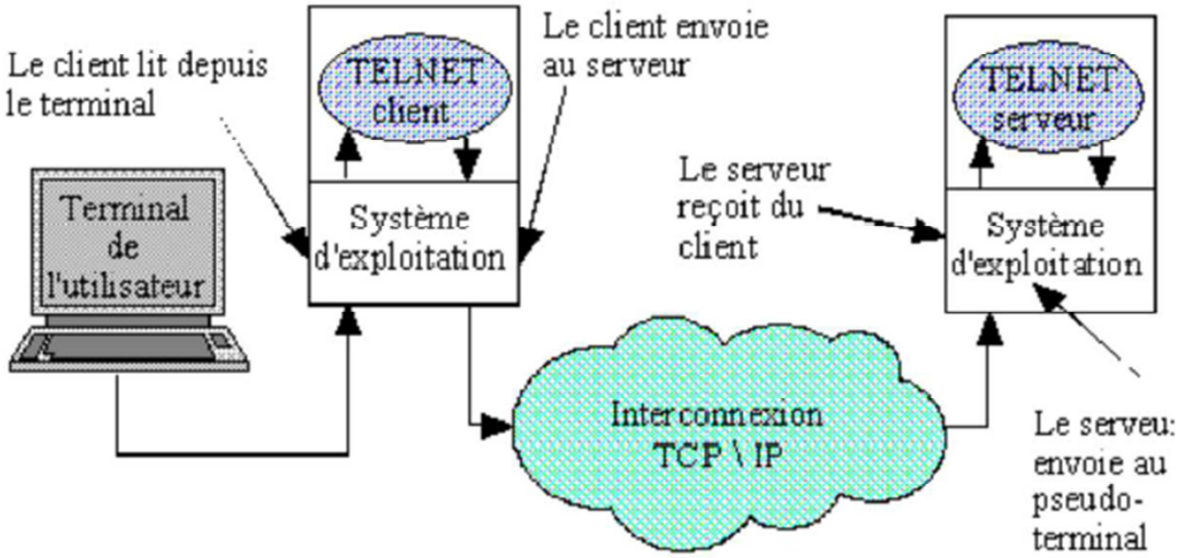


Figure V.1. *Itinéraire des informations dans une session TELNET*

L'utilisateur invoque un programme client qui établit une connexion TCP avec un serveur. Les caractères émis par le serveur TELNET comme s'ils étaient émis d'un terminal.

Le serveur TELNET est réalisé à l'aide d'un programme d'application ce qui rend les modifications et le contrôle plus faciles car le serveur ne se trouve pas dans le noyau. Mais l'efficacité est diminuée à cause du nombre d'échanges entre le client et le serveur. Pour obtenir l'interfonctionnement du plus grand nombre de systèmes d'exploitation, TELNET doit prendre en compte l'hétérogénéité.

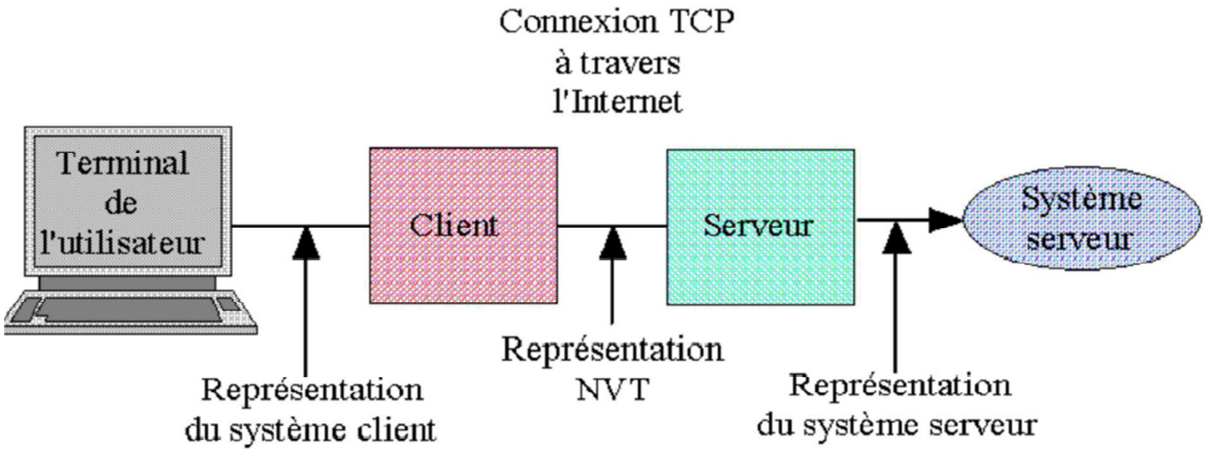


Figure V.2. *Structure terminal réseau virtuel de TELNET*

Pour résoudre ce problème, TELNET utilise la représentation terminal virtuel (NVT Network Virtual Terminal). Ainsi le client désirant émettre convertit les données de la représentation du système client en représentation NVT et le serveur recevant ces données les convertit de la représentation NVT en représentation du système serveur. Le protocole NVT fonctionne avec des mots de 8 bits mais il utilise le code USASCII 7 bits pour représenter les données et réserve les octets dont le bit de poids fort est à 1 pour les séquences de commandes.

V.1.2. Rlogin

Rlogin est un service de connexion à distance spécifique à Unix contrairement à TELNET qui est indépendant du type de machine et du système d'exploitation.
Rlogin définit un ensemble de machines sur lesquelles les noms d'utilisateurs et les droits d'accès sont partagés. Rlogin établit des équivalences entre les noms d'utilisateurs. Un utilisateur peut avoir un nom X sur une machine et un nom Y sur une autre.
Rlogin exporte l'environnement de l'utilisateur vers le serveur, donc les sessions de connexion à distance ont un comportement apparemment identique à celui des sessions locales [3].

V.*1.3. SSH*

SSH signifie *Secure SHell*. C'est un protocole qui permet de faire des connexions sécurisées (i.e. chiffrées) entre un serveur et un client SSH.

Ce protocole est largement utilisé de nos jours visant à garantir l'authenticité, la confidentialité et l'intégrité des données. En utilisant SSH, la transmission des informations au sein d'un réseau est chiffrée et ne peut donc, en théorie, être que difficilement compromise.

SSH possède plusieurs fonctionnalités. Il permet par exemple d'établir des sessions sécurisées sur un ordinateur distant, de transférer des fichiers sécurisés ou d'établir des tunnels sécurisés. SSH apporte donc une amélioration indéniable aux protocoles non sécurisés tels que Telnet, Rlogin, FTP.. [4].

V.2. Transfert de fichiers

Dans de nombreux réseaux les machines accèdent aux fichiers à distance. L'accès à distance permet de minimiser les coûts globaux. Certaines machines sans disque stockent leurs données sur un serveur distant unique et central. D'autre part, des machines dotées de disques envoient certaines copies de fichiers sur un service d'archivage distant. Enfin certaines organisations proposent le partage de fichiers entre plusieurs programmes ou plusieurs utilisateurs.
Il existe donc deux formes de partage de fichiers l'accès en ligne (on-line access) et la duplication de fichiers (whole-file copying) [3]:

- En ce qui concerne l'accès en ligne, plusieurs programmes peuvent accéder à un même fichier ce qui implique une modification immédiate du fichier. Le système permet l'accès de fichiers distants comme s'il s'agissait de fichiers locaux, c'est à dire que n'importe quelle application peut utiliser les fichiers distants sans que l'utilisateur n'ait à invoquer un programme client.

  L'avantage : un accès transparent. Inconvénient : l'accès aux fichiers dépend du réseau.
  D'autre part, l'accès en ligne est difficile à mettre en œuvre à cause de la correspondance des noms entre les deux systèmes, des notions de propriété, d'autorisation, de protection, et à cause du changement de représentation.

- L'autre forme du partage de fichier est le partage par transfert. Grâce à cette méthode, l'utilisateur obtient une copie locale d'un fichier avant de pouvoir le manipuler. Le transfert n'appartient pas au système de fichier, mais il y a invocation d'un programme client qui contacte le serveur sur la machine distante. Ensuite une autorisation est nécessaire pour l'accès au fichier.

  L'avantage du partage par transfert est l'efficacité car la manipulation d'un fichier local est beaucoup plus rapide. Malgré tout, il existe des problèmes concernant l'autorisation et la représentation. En effet, il y a parfois des difficultés à inverser des conversions.

V.2.1. FTP (File Transfert Protocol)

FTP représente une part importante du trafic sur les réseaux. FTP utilise le protocole de bout en bout TCP, et offre plusieurs options. En premier lieu, FTP offre l'accès interactif au serveur distant. Ensuite, il permet la spécification de la représentation des données. Enfin, il permet la vérification de l'authentification (nom et mot de passe). En ce qui concerne le modèle de traitement de FTP, il y a un serveur maître unique qui attend les demandes et crée un processus fils pour gérer chaque demande.

Le processus fils serveur accepte et gère l'établissement d'une connexion de contrôle qui servira à transférer les commandes. Il y a cependant un autre processus qui gérera la connexion de transfert de données. Les connexions de transfert et les processus qui les accompagnent sont créés dynamiquement, à la demande, mais la connexion de contrôle est maintenue pendant toute la durée de la session.
La connexion de contrôle permet également au client et au serveur de s'accorder sur les numéros de port TCP alloués dynamiquement afin d'établir la connexion de transfert. De plus, plusieurs clients peuvent communiquer avec un même serveur car TCP utilise les numéros de port des deux extrémités pour identifier les connexions [3].
Exemple de session FTP anonyme :
Dans cet exemple, l'utilisateur souhaite une copie du fichier fichier1 dans le répertoire /users/ sur la machine serveur.fr.

```
$ ftp www.serveur.fr
connected to serveur.fr
220 serveur.fr server FTP Ready.
Name (serveur:utilisateur_A) anonymous
Password guest
331 Guest login OK, access restrictions apply.
ftp> get /users/fichier1 fichier1
200 PORT command okay.
150 Opening data connection for /bin/ls
(172.194.46.2, 2363) (789088 bytes)
226 Transfer complete.
8272793 bytes received in 98.04 seconds
(82 Kbytes/s)
ftp> close
221 Goodbye.
ftp> quit
```

L'utilisateur précise le nom de la machine en paramètre de la commande "ftp". Ensuite il précise son nom et son mot de passe (il s'agit ici d'un "FTP anonyme"). La commande "get" lui permet d'obtenir la copie du fichier désiré. L'utilisateur tape "close" pour libérer la connexion vers le serveur, et il tape "quit" pour quitter le programme client. Les messages de contrôle échangés entre le client et le serveur débutent par un nombre de 3 chiffres destiné au logiciel et sont suivis d'un texte destiné à l'utilisateur. La commande PORT indique qu'un nouveau numéro de port a été alloué à la connexion de transfert de données [2].

V.2.2. TFTP (Trivial File Transfert Protocol)

FTP est le protocole de transfert de fichiers le plus utilisé mais il est également un des plus complexes. Certaines applications ne supportent pas cette complexité et n'ont pas besoin de toutes les fonctions de FTP. TFTP offre un service de base limité au transfert seul de fichiers et ne fait pas de contrôle d'accès. TFTP est donc moins volumineux mais aussi moins coûteux.
TFTP s'exécute au-dessus d'UDP ou tout protocole non fiable.
Le premier paquet envoyé indique la demande de transfert et initialise l'interaction. Chaque paquet doit être acquitté. Un bloc de moins de 512 octets indique la fin du fichier. Un message d'erreur différent d'un message de perte de paquet entraîne la terminaison du programme.
Le serveur utilise l'adresse IP et le numéro de port UDP du client pour identifier les opérations suivant la demande de lecture ou d'écriture [3].

V.2.3. rcp et scp

Permet de faire la copie à distance (*remote copy*) de fichiers (ou d'archives). **scp** en est la version sécurisée associée au paquet ssh. Ces deux commandes autorisent le transfert d'une machine vers une autre. scp devrait toujours être utilisé au lieu de rcp pour des raisons de sécurité [4].

V.3. Exercices

Questions de cours

- Quel est le but de la connexion à distance?
- Expliquez le principe du service DNS.
- Quel est le numéro de port dédié au http.

Exercice 1

Représentez sous forme d'un arbre de nommage les liens suivant :

www.univ-mascara.dz

www.univ-mascara.info.dz

www.info.univ-mascara.dz

On suppose que chaque faculté et chaque département à son propre nom de domaine.

VI. Références

[1] http://www.ybet.be/hardware2_ch2/hard2_ch2.php

[2] http://info.arqendra.net/Files/_Rsx+OSI+TCPIP_cours.pdf

[3] Didier Liroulet, ***« DESS Architecture des Systèmes d'Information et Communication»*** http://www.httr.ups-tlse.fr/pedagogie/exposes/tcp/tcp-9.1.html

[4] Stephane Brinster, Guillaume Lecomte, A Y Meric Bernard, « ***Nouvelles Technologies Reseaux Ssh – Ssl – Tls*** », Cour, Universite De Marne La Vallée Filière Informatique Reseau, 2003.

[5] Cours Télécommunication et réseau : http://sen-bretagne.net/Documents/Vannes-Stjoseph/

[6] Cours le monde des réseaux : http://www.gatoux.com/SECTION3/p6.php

[7] D. Comer, ***«TCP/IP : Architecture, protocoles, applications»***, 4ième édition, Dunod, ISBN 2-10-008181-0.

[8] Andrew Tanenbaum, ***« Réseaux»,*** 4ième édition, Pearson Education, ISBN 2-7440-7001-7.

[9] William STALLINGS, ***«Data and Communication Networks»***, 6th edition, Prentice Hall, 1999.

[10] Andrew Tanenbaum, ***« Réseaux: cours et exercices»,*** 3eme édition, Dunod, 1999.

[11] *Jean*-Luc Montagnier, ***« Pratique des réseaux d'entreprise »***, Eyrolles, 1998.

[12] *Douglas Comer*, ***« TCP/IP: architectures, protocoles et applications »,***5ième Ed, Pearson Education2006.

[13] Guy Pujoll, « Cours Réseaux et Télécoms », 3eme édition, Eyrolles, ISBN : 978-2-212-12414-9.

[14] C. Pham, Introduction aux Réseaux, Université de Pau.

[15] http://www.inetdoc.net/articles/adressage.ipv4/adressage.ipv4.subnet.html

[16] Benoît LE BONHOMME, « Le protocole Ipv6 », Mémoire de DESS. Université Paris 7, 2005.