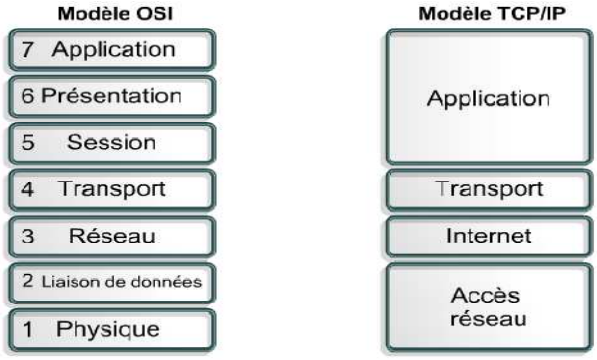
Chapitre : 6

Protocoles TCP/IP

1. Introduction et Généralité

TCP/IP (TCP : Transmission Control Protocol, IP: Internet Protocol) est un modèle mis au point par une agence de la défense des USA (DARPA). Son but était de permettre une interconnexion de réseaux quelque soit leurs technologies et leurs positions. TCP et IP sont de vieux protocoles développés dans les années 1960. Ils ont été implantés à la base du réseau ArpaNet qui est devenu, avec le temps, le cœur du réseau Internet. C'est pourquoi TCP/IP est la norme en vigueur sur l'Internet



Le modèle TCP/IP correspond à une **suite de protocoles** de différents niveaux participant à la réalisation d'une communication via un réseau informatique. Beaucoup de ces protocoles sont régulièrement utilisés par tous du fait de l'essor d'internet.

Telnet FTP HTTP SMTP DNS	application
TCP UDP	transport
IP ICMP ARP RARP	internet
PPP Ethernet ATM FDDI Token Ring *	hôte-réseau

* protocoles associés à des technologies d'architecture matérielle

2. Protocole IP

2.1. Définition

Le **protocole IP** (Internet Protocol), assure le service attendu de la couche réseau du modèle TCP/IP. Son rôle est donc de gérer l'acheminement des paquets (issus de la couche transport) entre les nœuds de manière totalement indépendante, même dans le cas où les paquets ont mêmes nœuds source et destination.

Le protocole IP offre un fonctionnement non fiable et sans connexion, à base d'envoi/réception de datagrammes (flux de bits structurés) :

- non fiable : absence de garantie que les datagrammes arrivent à destination ; les datagrammes peuvent être perdus, retardés, altérés ou dupliqués sans que ni la source ou la destination ne le sachent ;
- sans connexion (mode non-connecté) : chaque datagramme est traité et donc acheminé de manière totalement indépendante des autres.

2.2. Format du datagramme IP

Les messages transmis par IP sont appelés des datagrammes. Certains datagrammes sont des fragments d'un datagramme qui a dû être fragmenté.

Voici ce à quoi ressemble un datagramme :

<table>
<tr><th colspan="6">32 bits</th></tr>
<tr><td>Version
(4 bits)</td><td>Longueur d'en-tête
(4 bits)</td><td>Type de service
(8 bits)</td><td colspan="3">Longueur totale
(16 bits)</td></tr>
<tr><td colspan="3">Identification
(16 bits)</td><td>Drapeau
(3 bits)</td><td colspan="2">Décalage fragment
(13 bits)</td></tr>
<tr><td>Durée de vie
(8 bits)</td><td colspan="2">Protocole
(8 bits)</td><td colspan="3">Somme de contrôle en-tête
(16 bits)</td></tr>
<tr><td colspan="6">Adresse IP source
(32 bits)</td></tr>
<tr><td colspan="6">Adresse IP destination
(32 bits)</td></tr>
<tr><td colspan="6">Données</td></tr>
</table>

Voici la signification des différents champs :

- **Version** (4 bits) : il s'agit de la version du protocole IP que l'on utilise (ici on utilise la version 4 *IPv4*) afin de vérifier la validité du datagramme. Elle est codée sur 4 bits.
- **Longueur d'en-tête**, ou *IHL* pour *Internet Header Length* (4 bits) : il s'agit du nombre de mots de 32 bits constituant l'en-tête (nota : la valeur minimale est 5). Ce champ est codé sur 4 bits.
- **Type de service** (8 bits) : il indique la façon selon laquelle le datagramme doit être traité.
- **Longueur totale** (16 bits): il indique la taille totale du datagramme en octets. La taille de ce champ étant de 2 octets, la taille totale du datagramme ne peut dépasser 65536 octets. Utilisé conjointement avec la taille de l'en-tête, ce champ permet de déterminer où sont situées les données.
- **Identification, drapeaux (flags) et déplacement de fragment** sont des champs qui permettent la fragmentation des datagrammes.
- **Durée de vie** appelée aussi **TTL**, pour *Time To Live* (8 bits) : ce champ indique le nombre maximal de routeurs à travers lesquels le datagramme peut passer. Ainsi ce champ est décrémenté à chaque passage dans un routeur, lorsque celui-ci atteint la valeur critique de 0, le routeur détruit le datagramme. Cela évite l'encombrement du réseau par les datagrammes perdus.
- **Protocole** (8 bits) : ce champ, en notation décimale, permet de savoir de quel protocole est issu le datagramme
    - ICMP : 1 IGMP : 2 TCP : 6 UDP : 17

- **Somme de contrôle de l'en-tête, ou en anglais *header checksum* (16 bits)** : ce champ contient une valeur codée sur 16 bits qui permet de contrôler l'intégrité de l'en-tête afin de déterminer si celui-ci n'a pas été altéré pendant la transmission. La somme de contrôle est le complément à un de tous les mots de 16 bits de l'en-tête (champ *somme de contrôle* exclu). Celle-ci est en fait telle que lorsque l'on fait la somme des champs de l'en-tête (somme de contrôle incluse), on obtient un nombre avec tous les bits positionnés à 1
- **Adresse IP source** (32 bits) : Ce champ représente l'adresse IP de la machine émettrice, il permet au destinataire de répondre
- **Adresse IP destination** (32 bits) : adresse IP du destinataire du message

2.3. La fragmentation des datagrammes IP

Comme nous l'avons vu précédemment, la taille d'un datagramme maximale est de 65536 octets. Toutefois cette valeur n'est jamais atteinte car les réseaux n'ont pas une capacité suffisante pour envoyer de si gros paquets.

- De plus, les réseaux sur Internet utilisent différentes technologies, si bien que la taille maximale d'un datagramme varie suivant le type de réseau.
  - La taille maximale d'une trame est appelée *MTU* (Maximum Transfer Unit), elle entraînera la fragmentation du datagramme si celui-ci a une taille plus importante que le MTU du réseau.

Type de réseau	MTU (en octets)
Arpanet	1000
Ethernet	1500
FDDI	4470

- La fragmentation d'un datagramme se fait au niveau des routeurs, c'est-à-dire lors de la transition d'un réseau dont le MTU est important à un réseau dont le MTU est plus faible.
- Si le datagramme est trop grand pour passer sur le réseau, le routeur va le fragmenter, c'est-à-dire le découper en fragments de tailles inférieures au MTU du réseau et de telle façon que la taille du fragment soit un multiple de 8 octets.

- Le routeur va ensuite envoyer ces fragments de manière indépendante et les ré-encapsuler (ajouter un en-tête à chaque fragment) de telle façon à tenir compte de la

nouvelle taille du fragment. De plus, le routeur ajoute des informations afin que la machine de destination puisse réassembler les fragments dans le bon ordre.

- Pour tenir compte de la fragmentation, chaque datagramme possède plusieurs champs permettant leur réassemblage

2.4. **Adresse IP** (Internet Protocol):

C'est l'adresse numérique réelle dans le monde de l'Internet.

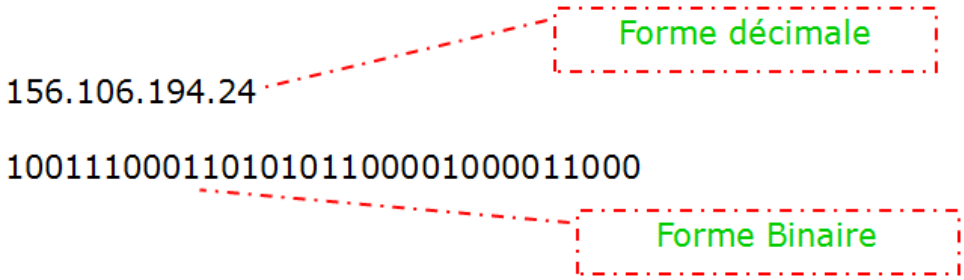
Les adresses IP ont une longueur fixe de 32 bits pour IPV4 (IP version 4). Ce sont des @ logiques à distinguer des @ physiques (les cartes Ethernet, Token Ring …etc.).

Une @ IP se compose de deux parties:

- Adresse Réseau (Netid)
- Adresse de l'hôte (hostid)

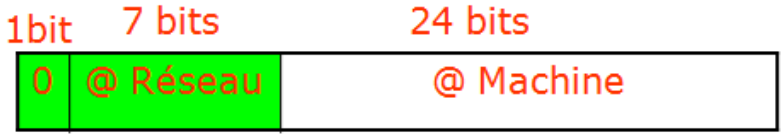
- Une @ logique doit être unique au monde c'est pourquoi seul le NIC (Network Information Center) attribue les @ réseau et la partie de l'hôte est laissée à l'appréciation de l'administrateur.
- L'@ IP est écrite par convention octet par octet séparés par un point pour la forme décimale et une suite de 32 bits pour la forme binaire.

Exemple:



2.4.1. **Les classes d'adresse IP**:

a) Classe A:

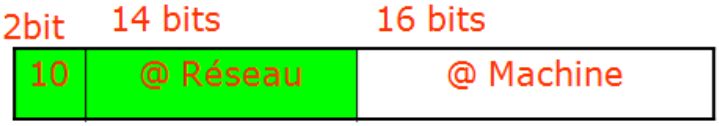


Cette classe est réservée aux très gros réseaux:

- Nbre très limité (128)
- Plus de 16 Millions d'équipements par réseau (possibles)

Remarque: Les @ IP de classe A sont expirées.

b) Classe B:



La classe B est réservée pour les gros réseaux:

- 16384 Réseaux
- 65536 machines par réseau

Remarque: Toutes Les @ IP de classe B ont déjà été attribuées (classe expirée)

c) Classe C:

- Max @ réseau 2^{21}
- max @ machine 2^{8}

C'est la classe utilisée par le grand publique (petits réseaux, particuliers).

Remarque: Cette classe est presque expirée c'est pour cela que nous avons actuellement un passage progressif vers IPV6 qui utilise un adressage sur 128 bits au lieu de 32 bits pour IPV4

d) Classe D

Cette classe est réservée à une utilisation particulière : le multicast

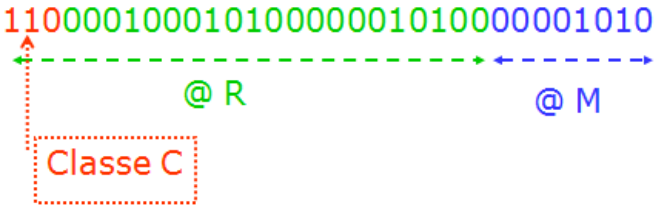
e) Classe E

Classe non utilisée à ce jour, elle est réservée pour les expériences.

Exemple:

Soit l'@ IP suivante: 194.40.20.10

Conversion binaire:



C'est une adresse d'une machine de classe C

L'@ du réseau est : 194.40.20.0

L'@ de diffusion de réseau est : 194.40.20.255

2.5.Les sous réseaux IP

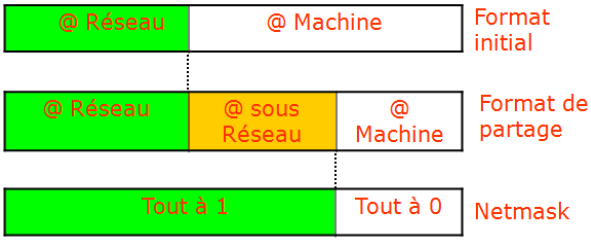
Principe

Il est possible de découper un réseau en entités plus petites appelées sous réseaux, ces sous réseaux qui ne sont pas visibles à l'extérieur du site de leurs existence sont crées par un administrateur.

Le principe de découpage d'un réseau en sous réseaux est basé sur l'utilisation des masques de sous réseaux (Netmask)

Masque

- Pour décomposer une adresse IP (c'est-à-dire séparer le Netid du hostid), il faut utiliser un masque (netmask).
- Chaque équipement effectuera une opération ET (bit à bit) entre l'adresse IP complète et le masque.
  Il suffit alors de placer des bits à 1 dans le masque pour conserver le netid et des 0 pour écraser le hostid . Un masque a donc la même longueur qu'une adresse IP.



Exemples

1-

Ex. : Soit l'adresse IP 192.168.1.72, associée au masque de réseau 255.255.255.0, abrégée en 192.168.1.72 / 24 (les 24 premiers bits définissent l'identifiant réseau, et donc les 8 restants l'identifiant d'hôte).

	192 . 168 . 1 . 72	
Adresse IP	1100 0000 . 1010 1000 . 0000 0001 . 0100 1000	192.168.1.72
Masque	1111 1111 . 1111 1111 . 1111 1111 . 0000 0000	255.255.255.0
Identifiant réseau	1100 0000 . 1010 1000 . 0000 0001 . 0000 0000	192.168.1.0
Identifiant hôte	0000 0000 . 0000 0000 . 0000 0000 . 0100 1000	0.0.0.72

Même adresse IP, associée au masque de réseau 255.255.255.224, abrégée en 192.168.1.72 / 27.

Adresse IP	1100 0000 . 1010 1000 . 0000 0001 . 010 0 1000	192.168.1.72
Masque	1111 1111 . 1111 1111 . 1111 1111 . 111 0 0000	255.255.255.224
Identifiant réseau	1100 0000 . 1010 1000 . 0000 0001 . 010 0 0000	192.168.1.64
Identifiant hôte	0000 0000 . 0000 0000 . 0000 000 0 . 000 0 1000	0.0.0.8

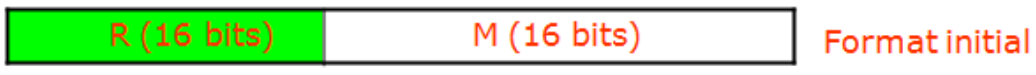
***2-* Exemple**: soit le réseau IP: 130.128.0.0

On veut faire un découpage en 30 sous réseaux.

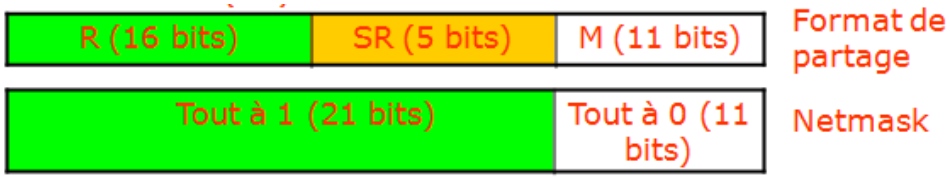
Donner le Netmask associé.

@ Binaire: 10000010100000000000000000000000

C'est la classe B



30 sous réseaux (SR) ➔ 5 bits nécessaires (30=11110)



D'où le masque: 11111111111111111111100000000000

En décimal: 255.255.248.0

2.6.Adresses interdites

Il y a des adresses interdites que l'on ne peut pas utiliser comme adresse IP pour un équipement :

- les adresses réseaux : c'est-à-dire les adresses dont tous les bits de la partie *hostid* sont à 0
- les adresses de diffusion générale (*broadcast*) : c'est-à-dire les adresses dont tous les bits de la partie *hostid* sont à 1
- l'adresse de boucle locale (*loopback*) 127.0.0.1 associé au nom *localhost*. De manière générale, toutes les adresses de ce réseau 127.0.0.0
- l'adresse 0.0.0.0 qui est utilisée par des différents services (DHCP, tables de routage, ...) et qui a souvent une signification particulière les adresses de lien local : ces adresses sont utilisables uniquement comme adresses de configuration automatique par défaut des interfaces d'hôtes (en cas d'absence de configuration manuelle explicite et de non-détection d'autres systèmes de configuration comme DHCP) : 169.254.0.0 - 169.254.255.255 (169.254/16)

2.7. Affectation des adresses IP

On distingue deux situations pour assigner une adresse IP à un équipement :

- de manière statique : l'adresse est fixe et configurée le plus souvent manuellement puis stockée dans la configuration de son système d'exploitation.
- de manière dynamique : l'adresse est automatiquement transmise et assignée grâce au protocole DHCP (Dynamic Host Configuration Protocol) ou BOOTP

3. Protocole TCP

3.1. Principe

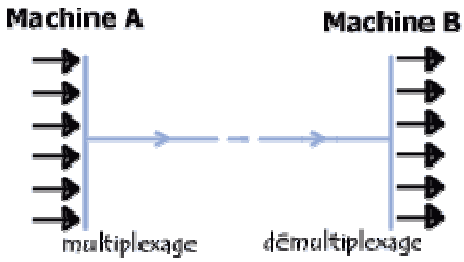
Pour pallier les problèmes suscités par le manque de fiabilité du protocole IP, un protocole fiable , appelé TCP, a été défini au niveau de la couche Transport. Nous examinerons plus loin la correction d'erreurs opérée par TCP

Le **protocole TCP** (Transmission Control Protocol [5]) assure les services attendus de la couche transport du modèle TCP/IP. Son rôle est donc de gérer le fractionnement et le réassemblage en paquets des segments de données [6] qui transitent via le protocole IP. Afin de fiabiliser la communication, TCP doit donc aussi réordonner les paquets avant de les assembler, et doit aussi gérer les paquets erronés ou perdus.

Pour cela, TCP fonctionne en mode *connecté* en usant de deux mécanismes mettant en œuvre un principe de synchronisation/question/réponse/confirmation :

- accusé de réception (/ acquittement : ACK) : tout envoi de données de la machine A vers la machine B est acquitté par B en renvoyant un acquittement à A ; cet acquittement est transporté soit par un paquet dédié, soit par un paquet transportant aussi des données à transmettre de B vers A [2] ;
  - l'acquittement doit être reçu avant l'échéance d'une temporisation amorcée par A lors de l'envoi des données ;
  - un paquet-acquittement peut transporter un acquittement cumulatif pour plusieurs envois de données distincts ;
  - l'émetteur conserve une trace des paquets émis ;
  - le receveur garde une trace des paquets reçus.

TCP permet d'effectuer une tâche importante: le multiplexage/démultiplexage, c'est-à-dire faire transiter sur une même ligne des données provenant d'applications diverses ou en d'autres mots mettre en série des informations arrivant en parallèle.



Ces opérations sont réalisées grâce au concept de ports (ou sockets), c'est-à-dire un numéro associé à un type d'application, qui, combiné à une adresse IP, permet de déterminer de façon unique une application qui tourne sur une machine donnée.

3.1. Le format des données sous TCP

Un segment TCP est constitué comme suit :

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="16">Port Source</td><td colspan="16">Port destination</td></tr>
<tr><td colspan="32">Numéro d'ordre</td></tr>
<tr><td colspan="32">Numéro d'accusé de réception</td></tr>
<tr><td colspan="4">Décalagedonnées</td><td colspan="6">réservée</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td colspan="16">Fenêtre</td></tr>
<tr><td colspan="16">Somme de contrôle</td><td colspan="16">Pointeur d'urgence</td></tr>
<tr><td colspan="22">Options</td><td colspan="10">Remplissage</td></tr>
<tr><td colspan="32">Données</td></tr>
</table>

Signification des différents champs :

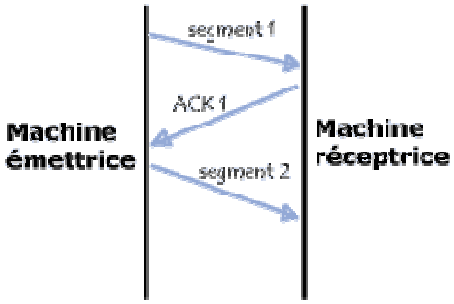
- **Port Source** (16 bits): Port relatif à l'application en cours sur la machine source
- **Port Destination** (16 bits): Port relatif à l'application en cours sur la machine de destination
- **Numéro d'ordre** (32 bits): Lorsque le drapeau SYN est à 0, le numéro d'ordre est celui du premier mot du segment en cours. Lorsque SYN est à 1, le numéro d'ordre est égal au numéro d'ordre initial utilisé pour synchroniser les numéros de séquence (ISN)
- **Numéro d'accusé de réception** (32 bits): Le numéro d'accusé de réception également appelé numéro d'acquittement correspond au numéro (d'ordre) du prochain segment attendu, et non le numéro du dernier segment reçu.
- **Décalage des données** (4 bits): il permet de repérer le début des données dans le paquet. Le décalage est ici essentiel car le champ d'options est de taille variable
- **Réservé** (6 bits): Champ inutilisé actuellement mais prévu pour l'avenir
- **Drapeaux (flags)** (6x1 bit): Les drapeaux représentent des informations supplémentaires :
    - **URG**: si ce drapeau est à 1 le paquet doit être traité de façon urgente.

    - **ACK**: si ce drapeau est à 1 le paquet est un accusé de réception.
    - **PSH** (PUSH): si ce drapeau est à 1, le paquet fonctionne suivant la méthode PUSH.
    - **RST**: si ce drapeau est à 1, la connexion est réinitialisée.
    - **SYN**: Le Flag TCP SYN indique une demande d'établissement de connexion.
    - **FIN**: si ce drapeau est à 1 la connexion s'interrompt.
- **Fenêtre** (16 bits): Champ permettant de connaître le nombre d'octets que le récepteur souhaite recevoir sans accusé de réception
- **Somme de contrôle** (Checksum ou CRC): La somme de contrôle est réalisée en faisant la somme des champs de données de l'en-tête, afin de pouvoir vérifier l'intégrité de l'en-tête
- **Pointeur d'urgence** (16 bits): Indique le numéro d'ordre à partir duquel l'information devient urgente
- **Options** (Taille variable): Des options diverses
- **Remplissage**: On remplit l'espace restant après les options avec des zéros pour avoir une longueur multiple de 32 bits

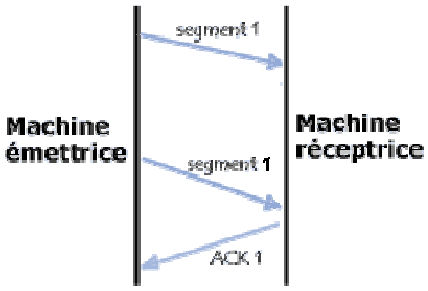
3.2.Fiabilité des transferts

Le protocole TCP permet d'assurer le transfert des données de façon fiable, bien qu'il utilise le protocole IP, qui n'intègre aucun contrôle de livraison de datagramme.

- En réalité, le protocole TCP possède un système d'accusé de réception permettant au client et au serveur de s'assurer de la bonne réception mutuelle des données.
- Lors de l'émission d'un segment, un **numéro d'ordre** (appelé aussi *numéro de séquence*) est associé.
- A réception d'un segment de donnée, la machine réceptrice va retourner un segment de donnée dont le drapeau ACK est à 1 (afin de signaler qu'il s'agit d'un accusé de réception) accompagné d'un numéro d'accusé de réception égal au numéro d'ordre précédent.



- De plus, grâce à une minuterie déclenchée dès réception d'un segment au niveau de la machine émettrice, le segment est réexpédié dès que le temps imparti est écoulé, car dans ce cas la machine émettrice considère que le segment est perdu...



- 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...

4. Protocole UDP

Le protocole UDP permet aux applications d'accéder directement à un service de transmission de datagrammes, tel que le service de transmission qu'offre IP.

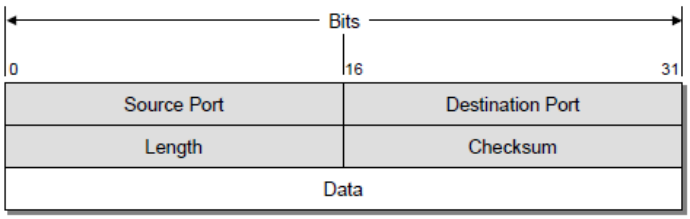
Caractéristiques d'UDP :

- UDP possède un mécanisme permettant d'identifier les processus d'application à l'aide de numéros de port UDP.
- UDP est orienté datagrammes (sans connexion), ce qui évite les problèmes liés à l'ouverture, au maintien et à la fermeture des connexions.
- UDP est efficace pour les applications en diffusion/multidiffusion. Les applications satisfaisant à un modèle du type « interrogation-réponse » peuvent également utiliser UDP. La réponse peut être utilisée comme étant un accusé de réception positif à l'interrogation. Si une réponse n'est pas reçue dans un certain intervalle de temps, l'application envoie simplement une autre interrogation.
- UDP ne séquence pas les données. La remise conforme des données n'est pas garantie.
- UDP peut éventuellement vérifier l'intégrité des données (et des données seulement) avec un total de contrôle.
- UDP est plus rapide, plus simple et plus efficace que TCP mais il est moins robuste.

Le protocole UDP permet une transmission sans connexion, mais aussi sans sécurité. Pourtant de nombreuses applications reposent sur UDP :

- TFTP
- DNS
- NFS
- SNMP
- RIP

L'en-tête a une taille fixe de 8 octets.



Le champ *Source Port* occupe 16 bits. Il indique :

- le numéro de port du processus émetteur,
- le numéro de port où on peut adresser les réponses lorsque l'on ne dispose d'aucun autre renseignement.
- si sa valeur est 0, cela signifie qu'aucun numéro de port n'est attribué.

Le champ *Destination Port* identifie le processus correspondant à l'adresse IP de destination auquel on envoie les données UDP. UDP effectue le démultiplexage des données à l'aide de numéros de port. Lorsqu'UDP reçoit un datagramme sans numéro de port, il génère un message d'erreur ICMP indiquant qu'il est impossible de contacter le port et il rejette le datagramme.

Le champ *Length* contient la longueur du paquet UDP en octets (en-tête + données). La valeur minimale est 8 et correspond à un paquet où le champ de données est vide.

Le pseudo en-tête de préfixe de l'en-tête UDP contient l'adresse d'origine, l'adresse de destination, le protocole (UDP = 17) et la longueur UDP. Ces informations sont destinées à prévenir les erreurs de routage.

5. Routage

Le routage IP est effectué sur la base du saut à saut. En effet, IP ne connaît pas, en général, la route complète entre deux machines (excepté si les deux machines sont sur le même réseau). La seule information fournie par le routage est l'adresse IP du routeur de saut suivant vers lequel le datagramme doit être envoyé.

5.1. Table de routage

- Une **table de routage** contient **l'adresse du destinataire** à atteindre (adresse de station, adresse d'un réseau) et le **prochain équipement** (**next hop**) à atteindre. Une **route par défaut** est toujours prévue
- Le routage est principalement réalisé par deux types d'équipements réseaux. Les stations, ne possédant en général qu'une seule interface réseaux et les routeurs qui possèdent plusieurs interfaces réseaux et qui gèrent la connexion entre deux ou plusieurs réseaux.
- De plus, des stations peuvent posséder plusieurs cartes réseaux et peuvent même dans certains cas être utilisées comme des routeurs (la différence est que dans ce dernier cas,

la station peut renvoyer, sur une interface réseau, un datagramme IP reçu par une autre de ces interfaces).

Tables de routage ont généralement la structure suivante.

Réseau	Interface	Prochain saut	Métrique	Age	Statut

Exemple:

Network	Interface	Next Saut	Metric	Age	Status
198.113.181.0	Ethernet0	192.150.42.177	[170/3047936]	02:03:50	D
198.113.178.0	Ethernet0	192.150.42.177	[170/3047936]	02:03:50	D
192.168.96.0	Ethernet0	192.150.42.177	[170/324608]	03:36:50	D
192.168.97.0	Ethernet0				C

Le champ Réseau (Network): désigne la destination

Le champ interface: IP consulte la table de routage pour savoir sur quelle interface envoyer le paquet. Pour effectuer cette consultation il extrait le numéro de réseau de l'adresse destination
Le champ prochain saut (Next saut): Le routage IP est effectué sur la base de saut à saut (hop to hop routing). Le message est transmis de routeur en routeur par sauts successifs, jusqu'à ce que le destinataire appartienne à un réseau directement connecté à un routeur. Celui-ci remet alors directement le message à la machine visée.
Exemple :

Le champ Métrique (Métric): La métrique (Metric) indique le coût relatif de l'utilisation de l'itinéraire pour atteindre la destination. Une métrique standard est le tronçon (ou saut), c'est-à-dire le nombre de routeurs à traverser avant d'atteindre la destination.

Le champ Age : La colonne Age de la table de routage est en général définie en seconde(s). Elle spécifie le nombre de secondes depuis que la route a été déterminée et validée. La valeur contenue dans Age est mise à jour dynamiquement et périodiquement afin que les informations utilisées par le routeur soient toujours valides.

Le champ Statut (Status): Chaque routeur émet périodiquement des informations sur son statut et sur l'état des routes qui le relient aux routeurs auxquels il est directement connecté. Ainsi, chaque routeur reçoit l'ensemble des mises à jour de ces statuts et se crée une carte de situation-réseau.

Comment sont établies les tables de routage ?
Les stratégies de routage sont:
- **statiques**: la mise à jour des tables de routage est établie par les administrateurs des différents équipements de l'Internet (Stations, passerelles...);

- **dynamiques**: la mise à jour est faite automatiquement en fonction de mesures de trafic; Les différents équipements peuvent s'échanger des informations de routage.

Il est recommandé un routage statique pour les stations et un routage dynamique (protocoles: RIP, OSPF) pour les routeurs. Il va de soi qu'un équipement a besoin d'une stratégie de routage s'il a deux adresses IP au moins.

5.2.Protocoles de routage

Différents protocoles de routage existent selon les techniques utilisées**: RIP, OSPF, IGRP, EGP, BGP**...

5.2.1. Exemple de protocole de routage : RIP

5.2.1.1. Principe et exemple

RIP signifie *Routing Information Protocol* (protocole d'information de routage). Il s'agit d'un protocole de type *Vector Distance* (Vecteur Distance), c'est-à-dire que chaque routeur communique aux autres routeurs la distance qui les sépare (le nombre de saut qui les sépare).

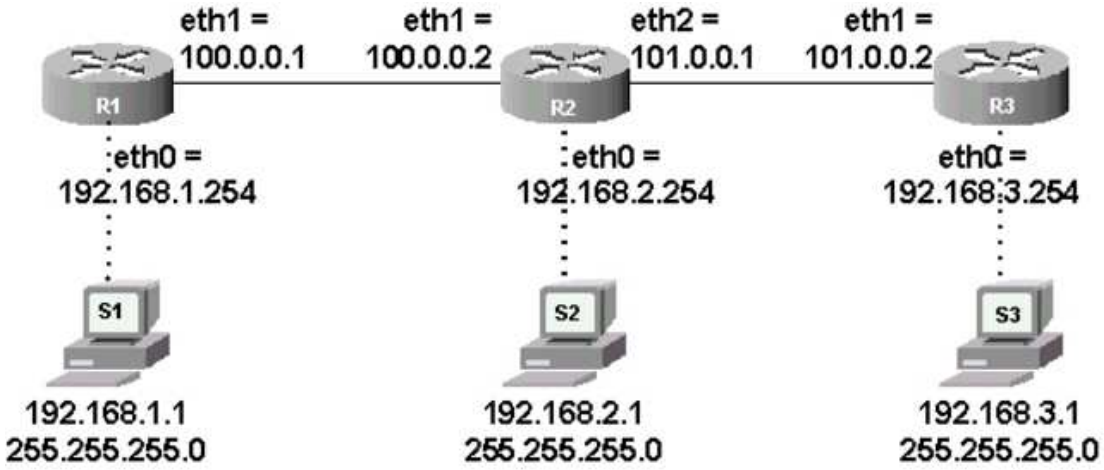
Comme toujours, pour qu'une communication puisse s'établir, chaque interlocuteur doit parler la même langue. Il a donc été nécessaire de concevoir un protocole. RIP a été défini, (on peut trouver RIP version 1 et RIP version 2). Par la suite, on ne traitera que RIPv2.

- **Quelles sont les informations de routage à échanger ?**

Le principe général est très simple. Un routeur RIP transmet à ses voisins les adresses réseau qu'il connaît (soit les adresses de ses interfaces, soit les adresses découvertes via les autres routeurs) ainsi que la distance pour les atteindre. Ces couples adresse/distance sont appelés vecteurs de distance.

- **La notion de distance**
  - La seule métrique utilisée par RIP est la distance correspondant au nombre de routeurs à traverser (hop count ou nombre de sauts) avant d'atteindre un réseau.
  - Pour chaque route, RIP calcule la distance. Ensuite, si des routes redondantes apparaissent, RIP retient celle qui traverse le moins de routeurs (donc avec la distance la plus faible).
- La norme limite la distance maximale d'une route à quinze. Cela signifie que deux réseaux ne peuvent être éloignés de plus de quinze routeurs

  Exemple



Topologie de travail

- **Situation initiale :** sur chaque routeur, toutes les interfaces réseau sont actives, aucune route statique n'est définie et le routage RIP est inactif.
- Sur R1, lorsque l'on active le processus de routage RIP, une première table est constituée à partir des adresses IP des interfaces du routeur. Pour ces réseaux directement connectés au routeur, la distance est égale à un puisqu'il faut au moins traverser ce routeur pour les atteindre. On obtient :

Adresse/Préfixe	Moyen de l'atteindre	Distance
100.0.0.0/8	eth1	1
192.168.1.0/24	eth0	1

Tableau 1. Table initiale constituée par R1

- R1 transmet à ses voisins immédiats (ici, il n'y a que R2) un seul vecteur de distance {192.168.1.0/24, 1} qui signifie : « je suis le routeur d'adresse IP 100.0.0.1 et je connais un moyen d'atteindre le réseau 192.168.1.0/24 en un saut ». Aucune information sur le réseau commun aux deux routeurs (100.0.0.0/8) n'est transmise, car R1 considère que R2 connaît déjà ce réseau.
- Ensuite, lorsque l'on active RIP sur R2, il constitue la table ci-après à partir de ses propres informations et de celles reçues de R1 :

Adresse/Préfixe	Moyen de l'atteindre	Distance
100.0.0.0/8	eth1	1
101.0.0.0/8	eth2	1
192.168.1.0/24	100.0.0.1	2
192.168.2.0/24	eth0	1

Tableau 2. table constituée par R2

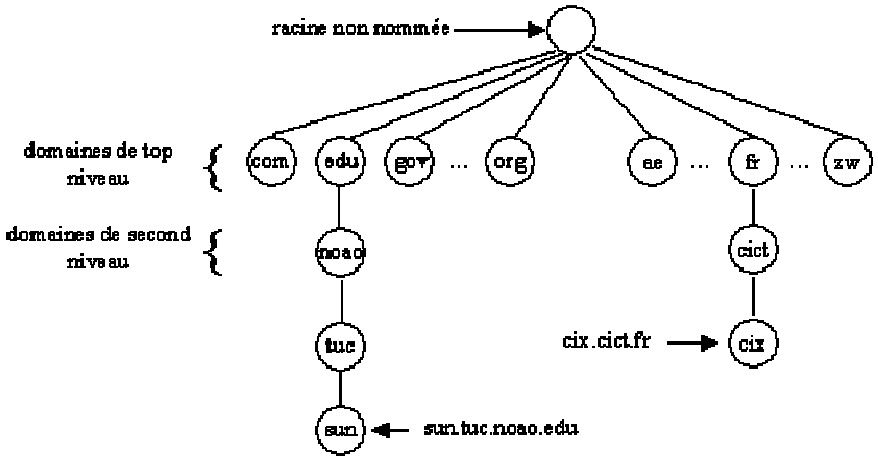
- Sur R2, RIP a calculé que la distance pour atteindre 192.168.1.0/24 est égale à deux puisqu'il faut traverser R2 puis R1. R2 a déduit le « moyen de l'atteindre » à partir de l'adresse IP de l'émetteur contenue dans le paquet RIP.

- Lorsque RIP sera démarré sur R3, la route vers 192.168.3.0/24 avec une distance de deux sera ajoutée dans la table ci-dessus.

Dans ce petit exemple, aucune restriction n'a été définie sur la diffusion des routes. Donc, à l'issue d'un certain délai appelé temps de convergence, variable selon la taille du réseau, chaque routeur connaît un moyen d'atteindre chaque réseau.

6. DNS

Pour pouvoir donner un nom sans ambigüité, un système de noms par **domaine** a été développé: le **DNS** (Domain Name System). Un nom se compose de plusieurs parties séparées par un point (ex. cix.cict.fr). En effet, c'est un nom **hiérarchique**. Il se lit de droite à gauche.
Le champ le plus à droite indique le nom de la **zone**, puis nous avons le nom de **domaine** et enfin le nom de la **machine**.
ex. asterix.ups-tlse.fr; aurore.cict.fr;



Le nom de zone peut être soit le nom du pays de rattachement (en deux lettres) soit le nom d'une catégorie d'utilisateurs (en trois lettres) souvent localisée aux Etats-Unis.

Les noms de zones en trois lettres correspondent à :
com entités commerciales
edu institutions éducatives (Universités);
gov agences gouvernementales (exemple la NASA) (uniquement USA)
mil entités militaires (uniquement USA)
net entités qui gèrent l'Internet
org organisations bénévoles
int organisations internationales

7. Protocoles principaux du modèle TCP/IP :

Niveau réseau :
IP : Internet protocol
ARP: Adresse resolution (conversion @ IP → @ Physique)
RARP: Reverse ARP (conversion @ Physique → @ IP)
RIP: Ancien Protocol de routage IP
OSFP: Nouveau Protocol de routage IP
Niveau transport :
TCP

UDP

Niveau application :

HTTP (HyperText Transport Protocol): Transport des fichiers hypertexts (pages Web)

FTP (File Transfer Protocol): Transfert de fichiers

Telnet : Système de terminal virtuel, permet d'accès distant aux applications

SMTP (Simple Mail Transfert Protocol): offre une service de courrier électronique

ICMP (Internet Control and error Message Protocol)

PPP (Point to Point Protocol

POP3 (Post Office Protocol version 3): récupération du courrier à distance

SLIP (Serial Line Interface Protocol): protocole d'encapsulation des paquets

DNS (Domaine Name System): pour les ordinateur qui se connectent momentanément sur l'Internet via un Modem, le fournisseur leur attribue une adresse temporelle (valable pour la connexion uniquement);

- Pour une connexion temporelle le DNS donne une @ temporelle.
- Pour une connexion permanente le DNS donne une @ IP.
- Et d'une façon générale le DNS permet le passage d'une @ IP à une @ de nommage et vis versa.

8. Internet

C'est un ensemble de réseaux interconnectés utilisant tous les mêmes protocoles de routage et de transport (TCP/IP).

Dans l'organisation de l'Internet on distingue:

Les opérateurs : de câblage et de transport, fournissent les points de connexion sur le réseau aux prestataires et grandes entreprises.

Les prestataires: qui sont connectés au réseau Internet, ils fournissent les adresses IP aux petites entreprises et aux particuliers ainsi que les services.

Les services: service messagerie, service de transfert de fichiers, service Web, … etc.

Illustration 1

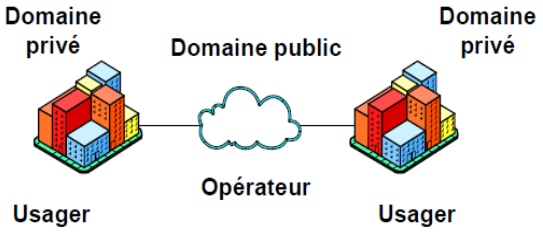


Illustration 2