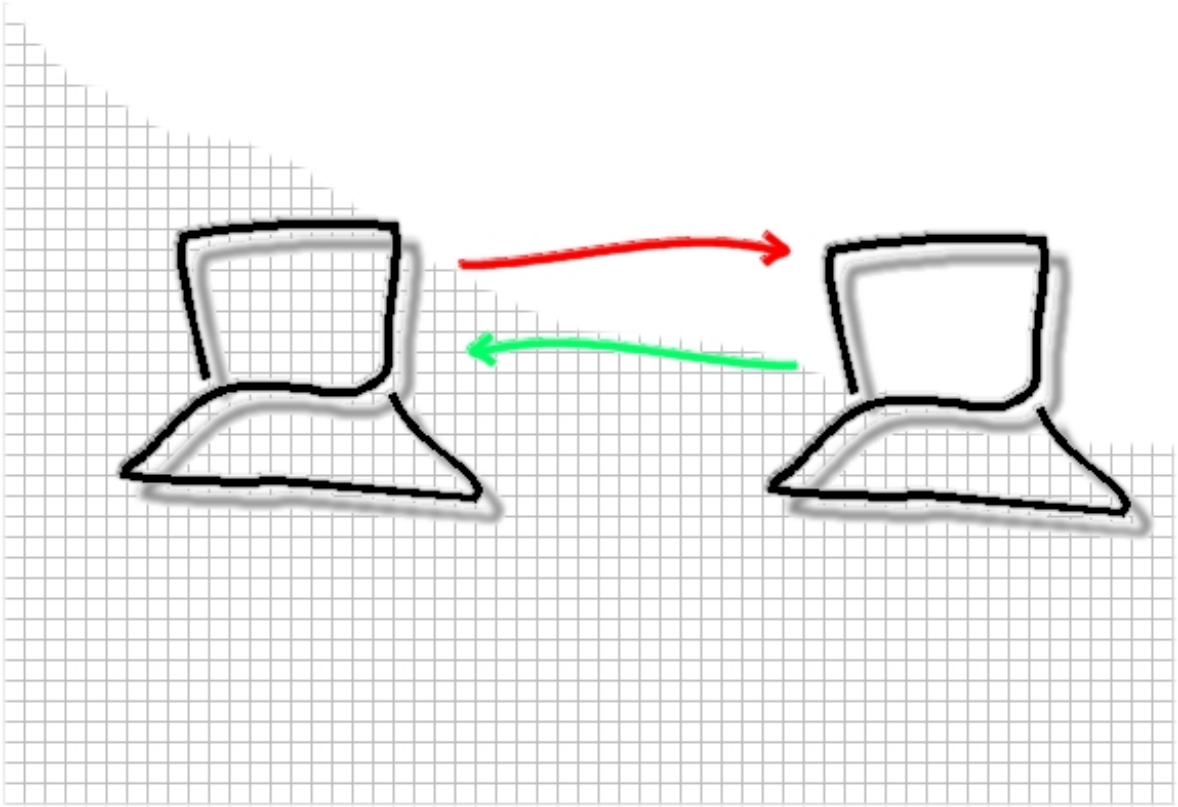
TCP / IP

Yann Duchemin
E-mail: yann.duchemin@free.fr



Pré-requ

Objectifs :

Apporter les notions essentielles pour l'interconnexion de réseaux dans des environnements de communications hétérogènes basés sur TCP/IP.

Version 1.0 - avril 2000

Table des matières

Historique

Les travaux de l'*ARPA (Advanced Research Project Agency)* débutèrent au milieu des années 1970 dans le but de développer un réseau à commutation de paquets afin d 'échanger plus facilement courrier et données entre les différends centres. Le but étant de construire un réseau résistant à d'éventuelles attaques militaires ou à des catastrophes naturelles, il ne fallait pas de points névralgiques, dont la neutralisation pouvait entraîner l'arrêt complet du réseau.

C'est ainsi que le réseau **ARPANET** fût conçu sans nœud particulier le contrôlant et de telle sorte que si une voie de communication venait à être détruite, le réseau soit capable de déterminer un nouveau chemin d'acheminement des données. En 1974 TCP est créé afin d'améliorer et de standardiser le mode de connexion entre machine hétérogènes. En 1978, TCP est fragmenté en TCP/IP, et c'est vers 1980 que le réseau **INTERNET** est apparu, à ce moment là l'ARPA commença à faire évoluer les ordinateurs en utilisant des protocoles de communication plus élaborés tel que TCP/IP, et allant jusqu'à subventionner l'université de Berkeley pour qu'elle intègre TCP/IP à son système d'exploitation Unix-BSD. Rapidement la quasi-totalité des départements informatiques des universités américaines sont connectées. Vers 1989 est créé le WWW (World Wide Web) par Tim Bernes-lee, mais seulement vers le milieu des années 1990 pour le commerce électronique.

Comme l'ensemble des protocoles TCP/IP n'est pas issu d'un constructeur unique mais de la collaboration de milliers de personnes à travers le monde, une structure de fonctionnement a été imaginée dès le début. Après des évolutions successives c'est aujourd'hui l'*IAB* (**Internet Architecture Board**) qui est chargée de coordonner l'architecture, la gestion et le fonctionnement d'Internet. L'IAB se compose de 2 branches principales : L'*IETF* (**Internet Enginnering Task Force**) pour les problèmes techniques et l'*IRTF* (**Internet Research Task Force**) pour coordonner les recherches relatives à TCP/IP.

Les documents relatifs aux travaux sur internet, les nouvelles propositions de définitions ainsi que les standards TCP/IP, sont publiés sous la forme de *RFC* (Request For Comments). Les numéros de RFC sont attribués dans l'ordre et ne sont jamais réutilisés, lorsqu'une norme est révisée, elle reçoit un nouveau numéro. - www.ietf.org - www.eisti.fr/eistiweb/docs/normes -

Le *Modèle OSI*

L'***O**pen **S**ystem **I**nterconnection* est une norme établie par L'*International Standard Organisation* , afin de permettre aux *systèmes ouverts* (ordinateur, terminal, réseau, ...) d 'échanger des informations avec d'autres équipements hétérogènes. Cette norme est constituée de 7 couches, dont les 4 premières sont dites *basses* et les 3 supérieures dites *hautes*. Le principe est simple, la couche la plus basse (directement au dessus du support physique) ne peut communiquer directement avec une couche n+1: chacune des couches est composée d'éléments matériels et/ou logiciels chargés de « transporter » le message à la couche immédiatement supérieure.

```
7 - Application
6 - Présentation  - : Passerelle
5 - Session
4 - Transport     - : TCP
3 - Réseau        - : Routeur
2 - Liaison       - : Pont
1 - Physique      - : Répéteurs
```

1 - La Couche Physique

Cette couche définit les caractéristiques techniques, électriques , fonctionnelles et procédurales nécessaires à l'activation et à la désactivation des connexions physiques destinées à la transmission de bits entres deux entités de liaisons de données.

2 - La Couche Liaison

Cette couche définit les moyens fonctionnels et procéduraux nécessaires à l'activation et à l'établissement ainsi qu'au maintien et à la libération des connexions de liaisons de données entre les entités du réseau. Cette couche détecte et corrige, quand cela est possible, les erreurs de la couche physique et signale à la couche réseau les erreurs irrécupérables.

3 - La Couche Réseau

Cette couche assure toutes les fonctionnalités de relais et d'amélioration de services entre les entités du réseau, c'est à dire : l'adressage, le routage, le contrôle de flux, la détection et la correction d'erreurs non résolues par la couche 2 (liaison) pour préparer le travail de la couche 4.

4 - La Couche Transport

Cette couche définit un transfert de données transparent entre les entités en les déchargeant des détails d'exécution (contrôle entre l'OS et le support de transmission). Son rôle est d'optimiser l'utilisation des services de réseau disponibles afin d'assurer à moindre coût les performances requise par la couche 5 (session).

5 - La Couche Session

Cette couche fournit aux entités de la couche présentation les moyens d'organiser et de synchroniser les dialogues et les échanges de données. Il s'agit de la gestion d'accès, de sécurité et d'identification des services.

6 - La Couche Présentation

Cette couche assure la transparence du format des données à la couche 7 (application).

7 - La Couche Application

Cette couche assure aux processus d'application le moyen d'accès à l'environnement OSI et fournit tous les services directement utilisable par l'application (transfert de données, allocation de ressources, intégrité et cohérence des informations, synchronisation des applications).

Le modèle OSI

<table>
<tr><td>Couche Application
7</td><td colspan="2">Programmes,
Applications Réseau</td><td colspan="4">Messagerie, Navigateur Internet,
ping, ftp, bootp, ...</td></tr>
<tr><td>Couche Présentation
6</td><td colspan="2">Utilitaire de conversion, de cryptage,...</td><td colspan="4">Interpréteurs ASCII/EBCDIC,
Emulateurs,...
(telnet, nfs, ...)</td></tr>
<tr><td rowspan="2">Couche Session
5</td><td colspan="6">Système d'exploitation</td></tr>
<tr><td colspan="2">Network
Operating
System</td><td colspan="4">Netware IP/IPX - NetBios - DEC -
snmp, ftp, smtp, telenet</td></tr>
<tr><td>Couche Transport
4</td><td colspan="6">SPX, PCLAN, LanManager, DecNet, PC/TCP
TCP (connecté -> ftp,http,smtp,...) - UDP (non connecté -> dns, tftp, ...)</td></tr>
<tr><td rowspan="2">Couche Réseau
3</td><td colspan="6">IEEE 802.1
(ip, ipx, rip, icmp, igmp,...)</td></tr>
<tr><td colspan="6">Drivers Réseau</td></tr>
<tr><td rowspan="4">Couche Liaison
2</td><td rowspan="2">Logical
Link
Control</td><td colspan="5">IEEE 802.2</td></tr>
<tr><td>CSMA/CD</td><td>Token Bus</td><td>Token Ring</td><td>DQDB</td><td>ISO 9314
ANSI X3T9.5</td></tr>
<tr><td>Medium
Acces
Control</td><td>IEEE 802.3</td><td>IEEE
802.4</td><td>IEEE
802.5</td><td>IEEE
802.6</td><td>FDDI</td></tr>
<tr><td colspan="6">isdn, lan, ppp, slip, ...</td></tr>
<tr><td rowspan="3">Couche Physique
1</td><td rowspan="2">Physical
Layer
Signaling</td><td rowspan="2">MTU
64/1518bytes</td><td rowspan="2">MTU
32b/16Kb</td><td rowspan="2">MTU
32b/16Kb</td><td rowspan="2"></td><td rowspan="2">MTU
32b/4400b</td></tr>
<tr><td>AUI</td></tr>
<tr><td colspan="6">Ethernet, paire torsadée, ...</td></tr>
</table>

Il est important de rappeler qu'un modèle TCP/IP a été proposé dix ans avant le modèle OSI, ce dernier s 'étant inspiré fortement de certains protocoles TCP/IP. Le modèles TCP/IP est basé sur quatre couches (on parle également de modèle simplifié) :

5 - 6 - 7	**Application**	telnet, ftp, dns, snmp, ...
4	**Transport**	TCP UDP
3	**Internet**	ICMP, IP, ARP, ...
1 - 2	**Interface réseau**	Ethernet, Token, ...

Acheminement des données

La couche liaison a pour but d'envoyer et recevoir des datagrammes IP pour la couche IP, et d'envoyer des requêtes ARP ainsi que de recevoir des réponses pour le module ARP.
Les adresses physiques Ethernet sont codées sur 6 octets (48 bits) dans l'état actuel, et censées êtres uniques gérées par les constructeurs et l'IEEE. Il en existe 3 types :

Unicast : pour une adresse monodestinataire désignant un seul coupleur
Broadcast : pour les adresses de diffusion générale (tous les bits « hosts » à 1)
Multicast : pour les adresses multidestinataires (ensemble de stations d'un même groupe)

Une trame Ethernet est composée d'une adresse matérielle source et destination (mac) et d'un crc de 4 octets, les autres champs différent selon le type Ethernet ou IEEE 802 :

Pour le format Ethernet le 3ème champ contient le type de données transmises selon que c'est un datagramme IP, requête ou réponse arp ou rarp. Ensuite arrivent les données d'une taille de 46 a 1500 octets. Si les données sont trop petite on complète avec des bits de bourrages (pad).
Pour le format IEEE 802, le 3ème champ indique le nombre d'octet de la trame sans le crc. Pour la sous couche LLC, le champ DSAP (Destination Service Access Point) désigne le ou les protocoles de niveau supérieur à qui sont destinées les données de la trame et le champ SSAP (Source Service Access Point) désigne le protocole qui a émis la trame.

Les Protocoles ARP et RARP

Le protocole IP, et ses adresses pouvant êtres utilisés sur des architectures matérielles différentes (Ethernet, Token-Ring, ...) possédant leur propres adresses physiques, il y a nécessité d'établir les correspondance entre adresses IP (logiques) et adresses matérielles (physiques) des nœuds d'un réseau. ARP fournit une correspondance dynamique entre une adresse IP connue et l'adresse matérielle lui correspondant. RARP, fait tout simplement l'inverse.

Exemple:

1. Un client ftp convertit l'adresse d'un serveur FTP (nom.domaine) en une adresse IP par le fichier *hosts* ou d'un serveur DNS
2. Le client ftp demande à la couche TCP d'établir une connexion avec cette adresse
3. TCP envoie une requête de connexion au serveur en émettant un datagramme IP contenant l'adresse IP
4. Sur un même réseau local, l'émetteur doit convertir l'adresse IP (4 octets) en adresse Ethernet sur 6 octets, avant d'émettre la trame ethernet contenant le paquet IP.
5. Le module ARP envoie une requête arp dans une trame ethernet avec une destination broadcast.
6. La couche ARP de la machine visée reconnaît que cette requête lui est destinée et répond par un reponse arp contenant son adresse matérielle. Les autre machines ignorent la requête.
7. La réponse ARP est reçue par l'émetteur de la requête (adresse connue puisque celle de l'émetteur).
8. La réponse ARP est reçue par le couche ARP du client ftp, le driver ethernet peut alors émettre le paquet IP avec la bonne adresse ethernet de destination.

Adressage IP

Une **adresse IP** identifie de manière unique une machine ainsi que le réseau sur lequel elle est située. Chaque adresse est une série de quatre octets dont une partie correspond à l'**identificateur du réseau** et l'autre partie à l' **identificateur de la machine.**

Il existe différentes classes d'adresses :

Adresses de *classe A*:
Le premier octet est compris *entre 1 et 126*, la partie identificateur réseau est codée sur ce premier octet,les trois octets suivants servent à coder la partie machine.
Nombre maxi de machines par réseau : 16 Millions (2^{24} - 2).

Adresses de *classe B*:
Le premier octet varie *entre 128 et 191*, la partie identificateur réseau est codée sur les deux premiers octets, la partie identificateur machine est codée sur les deux derniers octets.
Nombre maxi de machines par réseau : 65534 (2^{16} - 2).

Adresses de *classe C:*
Le premier octet varie de *192 à 223*, les trois premiers octets identifient le réseau, le dernier octet identifie la machine.
Nombre maxi de machines par réseau : 254 (2^{8} - 2).

Remarques :

Quelle que soit la classe d'adresses, deux adresses seront toujours réservées : la première (ex : 11.0.0.0) qui sera l'adresse du réseau (*host id* à « 0 »), et la dernière (ex: 11.255.255.255) qui sera l'adresse de broadcast (*host id* à « 1 »), c'est à dire celle utilisée pour diffuser un message à tous les hôtes du réseau.

Les adresses 10.x.x.x, 172.x.x.x et 192.x.x.x ne peuvent être utilisées pour aller sur internet (elles sont dites : non routables) : elles sont réservées aux réseaux internes d'entreprises.

L'adresse 127.x.x.x ne sera pas non plus attribuée : adresse de « *loopback* » (bouclage) qui permet par exemple de tester « l'état » d'une pile IP (fonctionnement interne à la machine).

Dans tous les cas, le masque de réseau sert à **identifier la partie de l'adresse IP correspondant au réseau (trouver l'adresse du réseau) et la partie correspondant à l'hôte.** En effet, l'adresse du réseau est calculée simplement en faisant un ET logique entre l'adresse IP et le masque de réseau.

Pour cela, il faut traduire l'adresse décimale en binaire, puis faire le ET logique en respectant la table suivante:

0 ET 0 = 0
0 ET 1 = 0
1 ET 0 = 0
1 ET 1 = 1

Chaque classe d'adresses possède son masque par défaut :

A : 255.0.0.0
B : 255.255.0.0
C : 255.255.255.0

Exemple

Prenons une adresse IP 12.32.23.15
Il s'agit d'une adresse de classe A.

Quelle est l'adresse du réseau?
Le masque par défaut pour une adresse de classe A est 255.0.0.0, donc si on fait le ET logique :

1) mettre l'adresse en binaire : `12.32.23.15` ➔`00001100.00100000.00010111.00001111`
2) mettre le masque en binaire: `255.0.0.0` ➔`11111111.00000000.00000000.00000000`
3) faire le ET logique ➔`00001100.00000000.00000000.00000000`
4) retraduire en décimal ➔ `12.0.0.0` = adresse du réseau

En fait, dans le masque réseau, les bits positionnés à 1 sont associés au numéro de réseau, alors que ceux positionnés à 0 correspondent à la partie hôtes. Ceci est important puisque c'est ainsi qu'on va retrouver le nombre de machines appartenant à un réseau .

Exemple avec une adresse de classe C : 200.13.13.26 , masque : 255.255.255.0.
Si on remet le masque en binaire ➔ 11111111.11111111.11111111.00000000,
on voit que la partie hôte est codée sur 8 bits, donc 2^8 - 2 = 254 machines maxi.

Le masque de réseau permet donc de retrouver l'adresse du réseau et le nombre maxi de machines sur un réseau donné. Ceci pourrait paraître inutile puisqu'on sait qu'un réseau avec une adresse de classe A par exemple va pouvoir comporter jusqu'à 16 Millions de machines et qu'on sait également retrouver son adresse réseau rapidement (le premier octet code l'identificateur réseau).

Les masques de réseau on donc d'autres utilités, c'est grâce à eux qu'on va pouvoir :
regrouper plusieurs réseaux en un réseau unique (*Supernetting*).
séparer un réseau en plusieurs sous-réseaux (*Subnetting*).

Exemple 1 : une entreprise possède une adresse réseau 195.47.58.0, elle voudrait séparer son réseau en trois sous-réseaux de chacun 60 machines minimum.. Quel masque va elle pouvoir appliquer?

L'adresse 195.47.58.0 est une adresse de classe C, son masque par défaut est donc 255.255.255.0.
Codé en binaire : 11111111.11111111.11111111.00000000.

On va « prendre » des bits normalement associés à la partie hôtes (0), en partant du bit de poids fort (gauche) que l'on va redonner à la partie identificateur réseau (les mettre à 1).

Comme on fonctionne en mode binaire, les sous-réseaux ne peuvent être crées que par blocs de deux; si on prend :

1 bit, on obtient deux sous-réseaux (2^1 =2)
2 bits, on obtient quatre sous-réseaux(2^2=4)
3 bits, on obtient huit sous-réseaux (2^3=8), etc...

Ici, l'entreprise veut 3 sous-réseaux : comme on ne peut en avoir exactement 3, on va en créer 4. On prend donc 2 bits à la partie normalement réservée aux hôtes, ce qui nous donne un masque de: 11111111.11111111.11111111.11000000, soit 255.255.255.192.
Il reste 6 bits pour coder la partie hôtes, soit 2^6 = 64 - 2 machines = 62 machines sur chaque sous-réseau.
Les adresses sont réparties ainsi :

.sous-réseau 1 : de 195.47.58.0 à 195.47.58.63 (avec 195.47.58.0 : adresse du réseau et 195.47.58.63 : adresse de broadcast)
.sous-réseau 2 : de 195.47.58.64 (adresse du réseau) à 195.47.58.127 (adresse de broadcast)
.sous-réseau 3 : de 195.47.58.128 (adresse du réseau) à 195.47.58.191 (adresse de broadcast)
.sous-réseau 4 : de 195.47.58.192 (adresse du réseau) à 195.47.58.255 (adresse de broadcast).

Exemple 2 :

Prenons le cas d'une entreprise qui possède 700 machines et dispose de trois adresses de classe C qui se suivent : 195.47.56.0; 195.47.57.0 et 195.47.58.0 .
Elle voudrait n'avoir qu'une seule plage d'adresses (et ainsi alléger ses tables de routage), on va pour cela utiliser un **masque de sur-réseau.**

Voici comment procéder :

- coder les trois adresses en binaire :

```
11000011.00101111.00111000.0
11000011.00101111.00111 001.0
11000011.00101111.00111010.0
```

- prendre la partie commune aux trois adresses sur le troisième octet (ici en gras) : on a cinq bits identiques, qui vont constituer le masque de sur-réseau en étant positionnés à 1, ce qui donne 11111000 soit 248.

Notre masque sera donc 255.255.248.0, et pour trouver le numéro du réseau il suffit comme d'habitude d'additionner le masque et l'adresse IP de l'hôte, on trouve n° de réseau : 195.47.56.0

Le *Protocole IP*

Ce protocole assure sans connexion un service non fiable de délivrance de *datagrammes IP*. Le service est non fiable car il n'existe aucune garantie pour que des datagrammes IP arrivent à destination. Certains peuvent êtres perdus, dupliqués, retardés, altérés ou remis dans le désordre ? On parle de remise au mieux et ni l'émetteur, ni le récepteur ne sont informés directement par IP des problèmes rencontrés. Le mode de transmission est non connecté car IP traite chaque datagramme indépendamment de ceux qui le précèdent ou le suivent. Ainsi, en théorie, au moins 2 datagrammes IP issue de la même machine et ayant la même destination peuvent ne pas suivre obligatoirement le même chemin.

Processus de communication du mode connecté :

1. L'émetteur demande l'établissement d'une connexion avec un hôte.
2. Si le récepteur (ou le gestionnaire du service) refuse la connexion, celle-ci n'a pas lieu.
3. Sinon un « lien » s'établit entre l'émetteur et le récepteur.
4. Les données transitent d'un point a l'autre.
5. La connexion est libérée.

Tout ce schéma est similaire à une communication téléphonique, l'avantage principal de ce mode de fonctionnement est l'identification de l'émetteur et du récepteur ainsi que la possibilité de définir une qualité de service à l'avance. L'inconvénient est la gestion « bavarde » pour de tout petits échanges de données, d'autre part une gestion complexe, mais aussi la complication des communications multipoints.

Processus de communication du mode non-connecté :

1. Envoi d'un message sur un support.
2. Le message contient les coordonnées du destinataire.
3. Chaque récepteur potentiel possède des coordonnées uniques.
4. Le contenu de l'information est inconnu de l'émetteur.
5. Le support est inconnu des utilisateurs (applicatifs).

Ce principe rappelle davantage celui du courrier postal, aucune vérification de la disponibilité du destinataire et des intermédiaires éventuels n'est effectuée avant l'envoi. Ce sont les équipements réseaux qui s'occupent de cette gestion.
Les blocs de données sont appelés « Datagrammes ».

IP a pour principales fonctionnalités :

Définir le format du datagramme IP
Définir le routage dans l'Internet
Définir la gestion de la remise non fiable des datagrammes

Le *Datagramme IP*

Un datagramme IP est constitué d'une entête suivie d'un champ de données.
En voici les champs:

La version code sur 4 bits le numéro de version du protocole IP utilisé (actuellement la version 4). Tout logiciel IP doit d'abord vérifier que le numéro de version du datagramme qu'il reçoit est en égale avec la sienne.

La longueur d'entête représente sur 4 bits la longueur, en nombre de mots de 32 bits. Ce champ est nécessaire si de part les options ajoutées, l'entête dépasse la longueur classique de 20 octets.

Le type de service (TOS) est codé sur 8 bits, indique la manière dont l'on doit traiter le datagramme et se décompose comme suit :

Un champ priorité (0 à 7)

4 bits D,T,R et C afin de spécifier ce que l'on veut privilégier pour la transmission de ce datagramme. D pour les délai d'acheminement, T pour le débit, R pour la fiabilité, et C pour le coût. Ces 4 bits ne sont pas incontournables mais peuvent aider à améliorer la qualité du routage.

La longueur totale contient la taille en octets du datagramme, ce champ étant sur 2 octets (16bits), on en déduit que la taille complète d'un datagramme ne peut dépasser 65535 octets.

Les champs d'identification, drapeaux et déplacements de fragment interviennent dans le processus de fragmentation des datagrammes IP

La durée de vie (Time To Live) indique le nombre maximal de routeurs que peut traverser le datagramme. Elle est initialisée par la station émettrice et décrémentée de 1 par chaque routeur qui recoit le datagramme et le réexpédie. Si un routeur reçoit un datagramme dont le ttl est nul, il le détruit et renvoie à l'expéditeur un message ICMP.

Le protocole permet de coder quel protocole de plus haut niveau a servi à créer ce datagramme. Les valeurs codées sur 8 bits sont 1 pour ICMP, 2 pour IGMP, 6 pou TCP et 17 pour UDP). Ainsi la station destinatrice qui reçoit un datagramme IP pourra diriger les données qu'il contient vers le protocole adéquat.

Le total de contrôle d'entête est calculé à partir de l'entête du datagramme pour en assurer l'intégrité, qui est elle même assurée par les protocoles ICMP, IGMP, TCP et UDP qui les émettent.

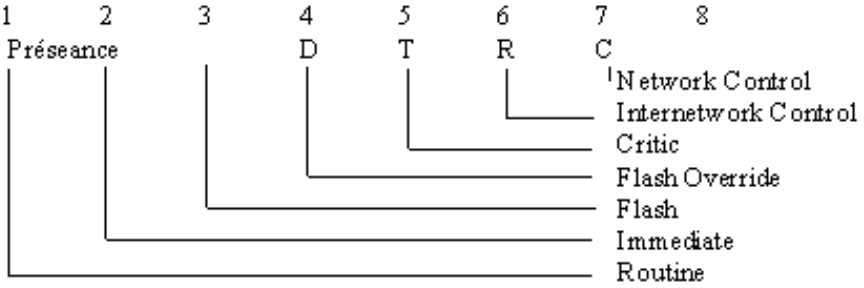
Les adresses IP sources et destinations contiennent sur 32 bits les adresses de la machines émettrice et destinataire finale du datagramme

Le champ options est une liste de longueur variable, mais toujours complémentée par des bits de bourrage pour atteindre une taille multiple de 32 bits

Exemple d'un Datagrammes IP :

Version Type d'enveloppe	Longueur de l'entête en mots de 32bits mini 5 max 15	Type de service Indication de priorité sur le routage (1)	Longueur Totale du datagramme = entête + données Taille du champs données : LongTot - Long Entete soit 2E16 = 65535
Identification Unique pour réassembler les fragments d'un même datagramme		Flags 000: frag sur le dernier 001: frag intermédiaire 010: frag interdite	Offset de fragmentation taille en octet des données du datagramme dans le segment (2)
Time To Live si = 0 alors destruction du datagramme (désincrémentation a chaque passage d'un routeur)		N° de protocole 1 icmp 5 stream 2 igmp 6 tcp 3 gcp 8 egp 4 ip 17 udp ...	CRC champ de contrôle de l'enveloppe uniquement
Adresse IP Source			
Adresse IP Destination			
Options			PAD
Données			

(1) Paramètre de gestion de la file d'attente des routeurs - de 0 à 7



(2) Sert lors de la fragmentation, ainsi lorsqu'un fragment est perdu, tous les autres sont détruits. Les réseaux hétérogènes utilisant des paquets de tailles différends, des débits différents, des taux d'erreurs variables, ...

Il existe d'autres limites à la taille que celle fixée par la valeur maximale de 65535 octets. Notamment, pour optimiser le débit, il est préférable qu'un datagramme IP soit encapsulé dans une seule trame de niveau 2 (Ethernet par exemple). Mais comme un datagramme IP peut transiter à travers Internet sur un ensemble de réseaux aux technologies différentes, on ne peut définir une taille maximale des datagrammes IP qui permette de les encapsuler dans une seule trame quel que soit le réseau (1500 octets pour ethernet, 4470 pour fddi, ...). On appelle la taille maximale d'un trame d'un réseau le MTU pour Maximum Transfert Unit et elle va servir à fragmenter les datagrammes trop grands pour les réseaux qu'ils traversent. La taille d'un fragment est choisie la plus grande possible tout en étant un multiple de 8 octets. Un datagramme n'est réassemblé que lorsqu'il arrive a destination finale, même si il traverse un réseau a plus grand MTU, les routeurs ne réassemble pas les petits fragments. De plus chaque fragment est routé de manière totalement indépendante des autres fragments du datagramme d'où il provient. Le destinataire finale qui reçoit un premier fragment d'un datagramme arme un temporisateur de réassemblage. Passé ce délai, si tous les fragments ne sont pas arrivés, ceux reçus sont détruits. En fait le destinataire décrémente de une unité selon un intervalle régulier le ttl de chaque fragment en attente, cela permet également de ne pas faire coexister au même instant 2 datagrammes avec le même identifiant.

Ce processus de fragmentation est rendu possible grâce aux champs suivants :

Le champ déplacement de fragment qui précise la localisation du début de fragment dans le datagramme. Les fragments sont des datagrammes dont l'entête est quasiment identique à celle du datagramme original.

Le champ identification est un entier qui identifie de manière unique chaque datagramme émis et qui est recopié dans chaque éventuels fragments.

Le champ longueur totale est recalculé à chaque fragment.

Le champ drapeaux comprend 3 bits dont 2 qui contrôle la fragmentation. Si le premier bit est positionné à 1 indique que l'on ne doit pas fragmenter le datagramme, et si un routeur doit fragmenter un tel datagramme, il revoir une erreur à l'expéditeur et rejette ce datagramme. Un bit appelé fragment a suivre est mis à 1 pour tous les fragments d'un datagramme sauf le dernier.

Les Protocoles TCP & UDP

Nous avons affaire ici aux deux principaux protocoles de la couche transport d'Internet. TCP, pour Transmission Control Protocol et UDP, pour User Datagram Protocol. Tous les deux utilisent IP comme couche réseau, mais TCP procure une couche de transport fiable, tandis qu'UDP ne fait que transporter de manière non fiable des datagrammes.

Le protocole UDP

Le protocole UDP utilise IP pour acheminer d'un nœud à un autre, en mode non fiable, des datagrammes qui lui sont transmis par une application. UDP n'utilise pas d'accusés de réception et ne peut donc pas garantir que les données ont bien été reçues. Ils ne réordonnent pas les messages si ceux-ci n'arrivent pas dans l'ordre dans lequel ils ont été émis et n'assure pas non plus le contrôle de flux. C'est donc à l'application utilisatrice d'UDP de gérer les problèmes de pertes de messages, duplications, retards, ...

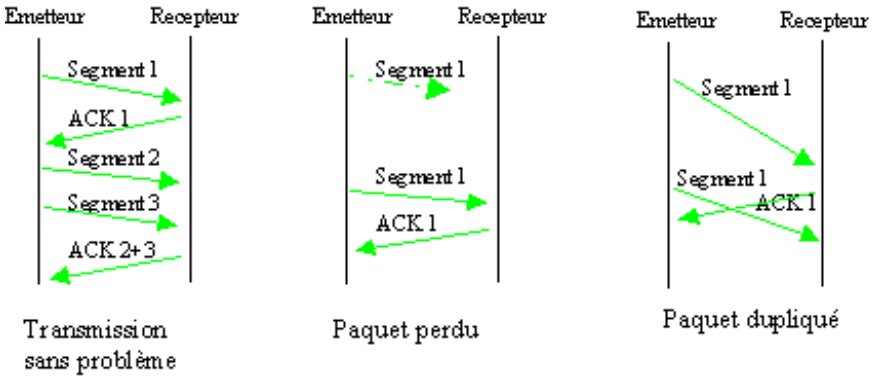
Cependant, UDP fournit un service supplémentaire par rapport à IP, il permet de distinguer plusieurs applications destinatrices sur la même machines par l'intermédiaire des « ports ». Un port est en fait une « adresse de destination » sur une machine identifiée par un numéro qui joue le rôle d'interface a l'application (ex: tftp : 69, snmp : 161, ...). Chaque datagramme émis par UDP est encapsulé dans un datagramme IP en fixant la valeur du protocole à 17. Un datagramme UDP est constitué comme suit :

Les numéros de ports sur 16 bits chacun, identifie les processus émetteurs et récepteurs.
Le champ longueur contient sur 2 octets la taille de l'entête et des données transmises
Le Checksum est en fait optionnel sur un réseau très fiable. S'il est fixé à 0 c'est qu'il n'a pas été calculé.
UDP utilise une pseudo-entête pour aboutir à un nombre d'octets total pair constituée de l'adresse IP source et destination ainsi que d'un octet (pad) afin d'aboutir à un checksum non nul stocké dans un champ de contrôle, cependant cette pseudo entête n'est pas transmise.

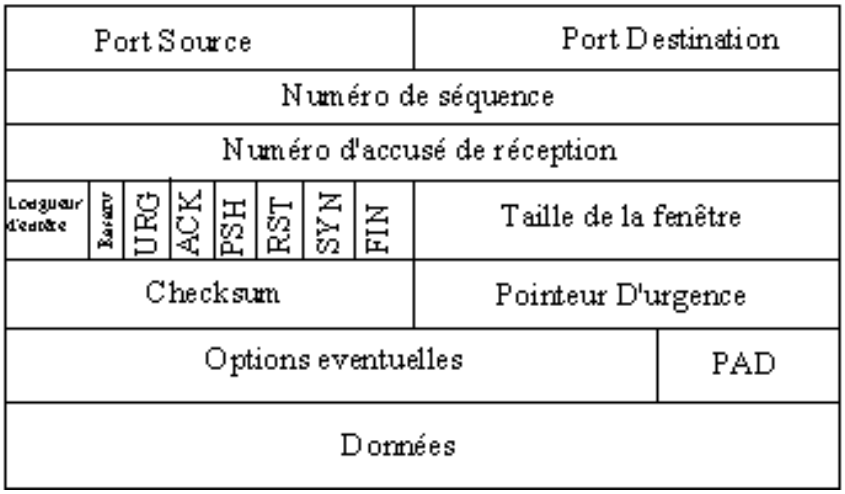
A la réception, UDP utilise l'adresse IP de destination et émettrice inscrite dans l'entête IP, afin de calculer une somme de contrôle qui permettra d'assurer que le datagramme est délivré sans erreur et à la bonne machine. Si une erreur est détectée, le datagramme UDP est détruit. Sinon, UDP oriente les données vers la file d'attente associée au numéro de port destination pour que l'application associée puisse les y lire.

Le protocole TCP utilise IP pour acheminer d'un nœud à un autre, en mode fiable. Chaque datagramme émis par TCP est encapsulé dans un datagramme IP en fixant la valeur du protocole à 6. Les applications dialoguant au travers de TCP sont considérées l'une comme serveur et l'autre comme client, elle doivent donc établir une connexion avant de pouvoir dialoguer. Il y a donc 2 extrémité communiquant l'une avec l'autre sur une connexion TCP (UDP permet de mettre en place du Broadcasting ou du multicasting). Tout au long de la connexion, TCP échange un flux d'octets sans qu'il soit possible de séparer par une marque quelconque certaines données, c'est donc aux applications de savoir gérer la structure du flot de données. Les données trop volumineuses sont fractionnées en fragments de taille appréhendés par TCP. Au contraire, TCP peut regrouper des données d'une application pour en former qu'un seul datagramme de taille convenable afin de ne pas charger inutilement le réseau. Cette unité d'information est appelée segment. Certaines applications demandent à ce que les données soient émises avant même que le tampon ne soit plein. Pour cela on utilise le principe du *push* pour forcer le transfert. Les données ainsi acheminées sont marquées par un bit particulier pour la couche TCP réceptrice de ce segment remette immédiatement les données à l'application concernée.

La fiabilité fournie par TCP consiste à remettre des datagrammes, sans pertes, ni duplication. On utilise pour cela la technique de l'accuse de réception (ACK) :



Chaque segment est émis avec un numéro qui va servir au récepteur pour envoyer un accusé de réception. De plus, à chaque envoi de segment, l'émetteur arme une temporisation qui lui sert de délai d'attente de l'accusé de réception correspondant. Mais il se peut que la temporisation expire alors que le segment a été transmis sans problème, dans ce cas l'émetteur réémet un segment alors que c'est inutile. Le récepteur conservant une trace des numéros de segments reçus, il peut éliminer les doublons.

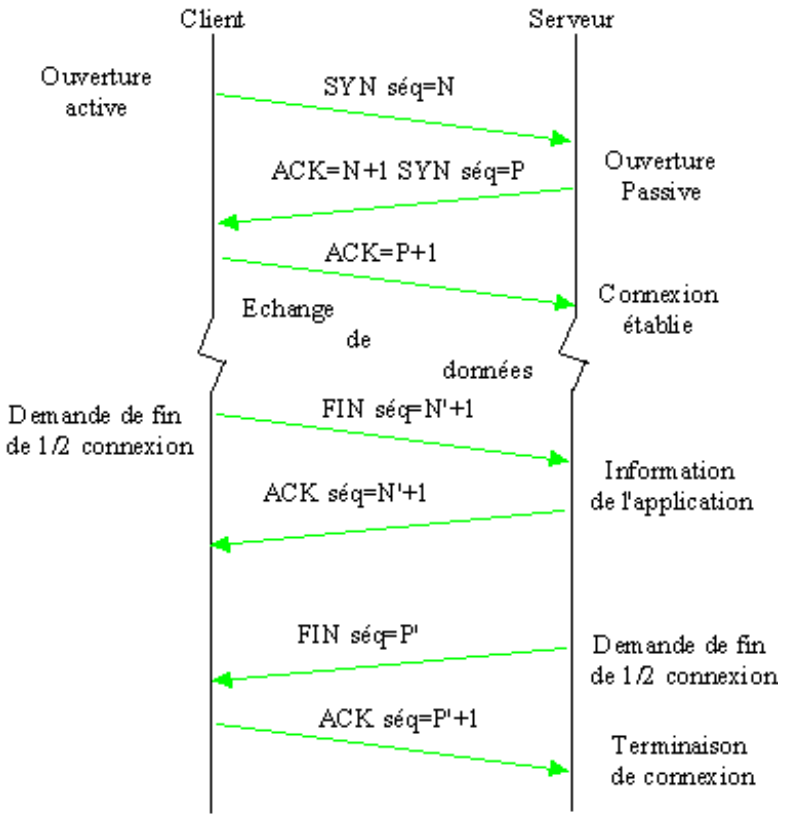


Description des différents champs :

Le port source et destination identifient les applications émettrice et réceptrices en les associant avec les numéros IP sources et destination du datagramme IP qui transporte le segment TCP.
Le numéro de séquence donne la position du segment dans le flux de données envoyées par l'émetteur.
Le numéro d'accusé de réception contient en fait le numéro de séquence suivant que le récepteur s'attend à recevoir, c'est à dire le numéro de séquence du dernier octet reçu avec succès plus 1.
La longueur d'entête, contient sur 4 bit la taille de l'entête, y compris les options présentes codées en multiples de 4 octets. Ainsi on peut avoir une entête de 20octets (pas d'option) à 60 octets (maximum d'options).
Le champ réservé sur 6 bits.
Les 6 champs bits de code permettent de spécifier le rôle et le contenu du segment TCP pour l'interprétation des champ de l'entête:

URG : pointeur de données urgente
ACK : accusé de réception
PSH : ce segment requiert un push
RST : réinitialiser la connexion
SYN : synchronisation des numéros de séquences pour initialiser une connexion
FIN : l'émetteur a atteint la fin de son flot de données

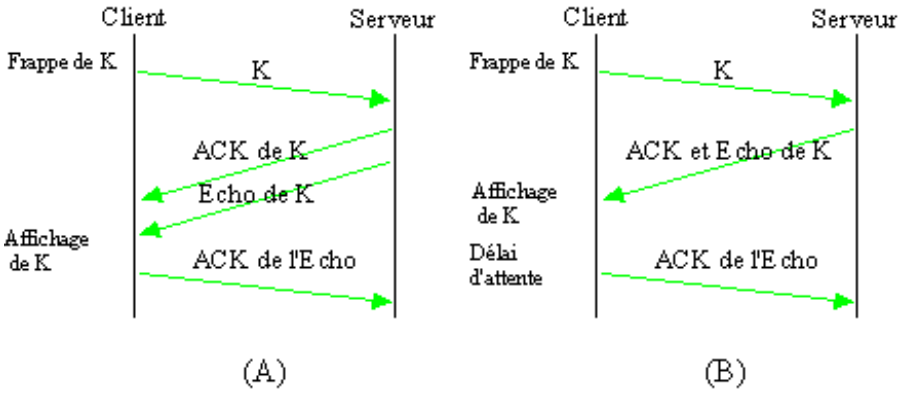
La taille de fenêtre, est un champ de 16 bits qui sert au contrôle de flux selon la méthode de la fenêtre glissante. Il indique le nombre d'octet que le récepteur est prêt a accepter jusqu'à 65535 octets. Ainsi l'émetteur augmente ou diminue son flux de données en fonction de la valeur de cette fenêtre qu'il reçoit.
Le checksum, est une somme de contrôle sur 16 bits utilisé pour vérifier la validité de l'entête est des données transmises. Il est obligatoirement calculé par l'émetteur et vérifié par le récepteur.
Le pointeur d'urgence est un offset positif qui, ajouté au numéro de séquence du segment, indique le numéro du dernier octet de données urgentes. Il faut également que le bit « URG » soit positionné à 1 pour indiquer des données urgente que le récepteur TCP doit passer le plus rapidement possible à l'application associée à la connexion.
L'option la plus couramment utilisée est celle de la taille maximale du segment TCP qu'un extrémité de la connexion souhaite recevoir.



L'établissement et la terminaison d'une connexion suit le diagramme d'échange de la figure ci-contre :

Le client demande l'ouverture de la connexion en émettant un segment « SYN » (bit « SYN » fixé à 1) spécifiant le n° du port du serveur avec lequel il souhaite se connecter.
Il expédie aussi un n° de séquence initial N.
Cette phase est appelée ouverture active et consomme un n° de séquence.
Le serveur répond en envoyant un segment avec les bits « ACK » et « SYN »positionnés à 1 : ainsi, il acquitte le premier segment reçu avec une valeur « ACK »= N+1 et indique un n° de séquence initial. Cette phase est appelée ouverture passive. Le client doit également acquitter ce deuxième segment en renvoyant un segment avec « ACK »= P+1 (pour le cas où 2 demandes de connexion auraient lieu en même temps, chacune dans un sens).

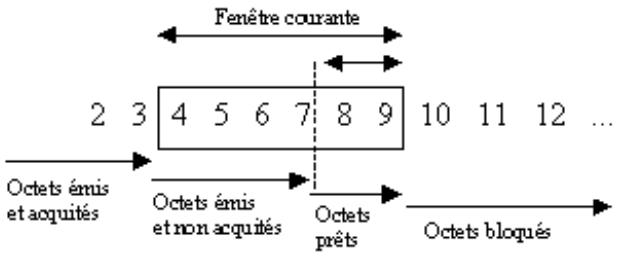
La terminaison d'une connexion peut être demandée par n'importe quelle extrémité et se compose de 2 « demi-fermetures » (des flots de données pouvant s'écouler simultanément dans les deux sens).
L'extrémité qui demande la fermeture (dans la figure le client) émet un segment où le bit FIN est positionné à 1 et où le n° de séquence vaut N'. Le récepteur du segment l'acquitte en retournant un « ACK »= N'+1 et informe l'application de la demi-fermeture de la connexion. Les données ne peuvent alors que transiter dans un sens (de l'extrémité ayant accepté la fermeture vers l'extrémité l'ayant demandée). Dans l'autre sens, seuls des accusés de réception sont transmis.
Quand l'autre extrémité veut fermer la connexion, elle agit de même ce qui entraîne la terminaison complète de la connexion.



Le transfert de données TCP est de 2 types : interactif, dans lequel chaque segment transporte très peu d'octets, et le transfert en masse où chaque segment transporte un maximum d'octets.
Cette distinction est confortée par une étude de 1991 qui indique que la moitié des paquets TCP contient des données en masse (FTP, Mail...) et l'autre moitié des données interactives (Telnet, Rlogin...).
Mais 90% des octets transmis proviennent de données en masse (90% des paquets Telnet et Rlogin comportent moins de 10 octets...)

1) Un exemple de **transfert interactif** est celui généré par rlogin lancé depuis un client vers un serveur. Dans ce cas, tous les caractères tapés par l'utilisateur sur le client sont envoyés vers le serveur en utilisant un caractère par segment, et ils sont ensuite renvoyés par le serveur pour un « echo » sur l'écran du client.
Tous les segments échangés dans ce cas là ont leur bit « PSH » fixé à 1, or tout segment doit être acquitté dans un sens comme dans l'autre, ce qui amène à la figure ci-dessus (A) :

En fait TCP gère ce type d'échanges avec la procédure d'acquittement retardé qui consiste à envoyer l'acquittement en l'incluant dans un segment qui transporte également des données (figure b). Cependant, la frappe de la commande Date sur le client provoquerait quand même l'échange de 15 datagrammes IP de 41 octets (1 octet pour le caractère, 20 pour l'entête TCP et encore 20 pour l'entête IP) : ceci pourrait être préjudiciable au bon fonctionnement d'un WAN... Cette solution sera celle utilisée pour un réseau à fort débit et très peu chargé.
Une autre solution existe qui consiste pour la couche TCP du client à accumuler de petits volumes de données et à les envoyer dans le même segment : c'est l'algorithme de Nagle.
Attention : dans le cas où de petits messages doivent être transmis sans délai (temps réel), ne pas utiliser cette méthode.



Dans le cas d'un **transfert de données en masse**, TCP utilise la technique de la fenêtre glissante pour contrôler le flux des échanges (de bout en bout) : ceci est primordial quand un micro-ordinateur dialogue avec un gros ordinateur, sinon le tampon d'entrée du micro sera très vite saturé. Il s'agit également de réguler le trafic en fonction de la charge des routeurs et du débit des réseaux traversés. Rappel : l'ensemble d'un flux de données unidirectionnel d'une machine A vers une machine B est constitué d'une séquence d'octets numérotés individuellement. La fenêtre glissante va consister à fixer quels sont les octets appartenant au flux que A peut émettre : dans la figure suivante,la fenêtre couvre les octets de 4 à 9, car la taille de la fenêtre courante est de 6 et que tous les octets jusqu'au 3ème inclus ont été émis et acquittés. A tout instant, TCP calcul sa fenêtre utilisable qui est constituée des octets présents dans la fenêtre et non-encore envoyés.

2) Pour le flot de A vers B, la taille de la fenêtre est contrôlée par B qui envoie dans chacun de ses accusés de réception la taille de la fenêtre qu'il désire voir utilisée. S'il demande une augmentation, A déplace le bord droit de sa fenêtre courante et émet immédiatement les octets qui viennent d'y entrer. S'il demande une diminution, un rétrécissement de la fenêtre est effectué avec l'arrivée des accusés de réception.

Principe de fonctionnement du routage

Attention de ne pas confondre protocole routable et protocole de routage. Par protocole routable on sous-entend un protocole de niveau 3 du modèle OSI, tel que IP, IPX, DecNet, ...
Une station qui veut communiquer vers une autre station n'appartenant pas à un même réseau logique doit solliciter un routeur afin de déterminer le chemin pour y parvenir. Cette station doit au moins bénéficier d'un logiciel réseau lui permettant de traiter des informations de niveau 3. On a alors plusieurs choix pour y parvenir : soit il existe un système dynamique permettant de découvrir le routeur, soit la station connaît l'adresse du routeur par défaut. On parle alors de routage **statique** ou de routage **dynamique**. Un protocole routable est en fait un protocole d'encapsulation qui provient de la couche transport afin d'assurer les services de la couche réseau.

Quoi qu'il en soit le routeur par défaut et la station émettrice doivent être sur le même domaine de broacast mac (physiquement liés par un média), un routeur ne prend en compte que les trames mac qui lui sont destinées (contrairement au pont qui lit systématiquement toutes les trames en circulation sur le média).

La communication entre 2 nœuds peut se faire de manière *directe*, c'est l'adresse réseau qui est comparée, si elle est identique alors la transmission du datagramme est réalisée. Si l'adresse réseau est différente, on parle de routage *indirect* alors au moins une passerelle sera traversée. Dans ce dernier cas, le datagramme est émis à la passerelle, celle-ci extrait les données, sélectionne le prochain nœud (récepteur des données, ou passerelle par défaut), remet en forme la nouvelle trame et la réémet.

Un mécanisme spécifique de résolution d'adresse (ARP), est mis en œuvre lors de l'établissement de la session de niveau 3. La station émettrice diffuse alors un paquet de broacast, à l'aide de l'adresse réseau du routeur (niveau 3), qui demande de renvoyer l'adresse mac du routeur qui détient l'adresse logique de la station réceptrice. Une fois l'adresse mac connue, la station émettrice envoie toutes les trames suivantes à cette adresse mac, donc au routeur, mais avec l'adresse logique de la station cible. Sur l'autre réseau logique (destination), la station reçoit les trames en provenance du routeur, croyant qu'il s'agit directement de la station émettrice (l'adresse mac connue est celle du routeur, et l'adresse physique est celle de la station émettrice).

Les stations associent donc l'adresse mac du routeur à l'adresse logique de toutes les stations se trouvant sur les réseaux situés de l'autre côté du routeur.

Des fonctions évoluées peuvent être mises en place par la composante logicielle. Par exemple, le contrôle de flux, qui analyse les paquets (voir les champs des datagrammes) et affecte ainsi une priorité forte aux sessions transactionnelles (ex.: telnet tcp-port 23) et moins forte pour les messages liés à un transfert de données (ex.: http tcp-port 80). On peut également effectuer des filtrages pour augmenter la sécurité, que ce soit sur les adresses, les protocoles, ou encore le type d'application.

Tout comme les trames *mac* ou *llc* permettent de connaître le protocole transporté dans le champ de données, les paquets disposent d'un champ identifiant le type d'application. Ce type de filtrage dépend donc du niveau 3 (cf. schéma page suivante).

Familles de routage

Le routage est divisé en 2 familles, d'une part le routage entre 2 systèmes autonomes différents, baptisé **EGP** pour *Exterior Gateway Protocol* (les exemples les plus courants sont EGP, BGP). L'autre famille réalise le routage entre 2 routeurs du même segment autonome, il s'agit d'**IGP** pour *Interior Gateway Protocol*. Cette dernière famille, IGP, est elle-même divisée en 2 types de protocoles :

- Les protocoles à **vecteur distance** (tels que le sont *RIP, IGRP, EIGRP, ...*)

Ils sont simples à utiliser, et nécessitent peu de ressources.

- Les protocoles à **état de liens** (tels que le sont *OSPF, ES-IS, IS-IS, ...*)

Leur convergence est rapide, la création de boucle est bien contrôlée, il est possible d'avoir des chemins d'accès multiples.

Le principal point de contrôle déterminant le chemin à emprunter pour transmettre les données est ce que l'on nomme le **métrique** :

Le métrique est implémenté dans les tables de routage, et utilisé par les protocoles de transmission de paquets, c'est un critère de comparaison.

Exterior Gateway Protocol

Il s'agit là de l'un des plus anciens protocoles de routage inter-domaines, ses fonctionnalités le rendent peu pratique pour les réseaux actuels. Il a été conçu pour rallier des sous-ensembles à un réseau unique (topologie en étoile), qui était ARPANET. Le calcul des routes se fait en communiquant la découverte de segments au voisin direct (1 seul saut) le plus proche. La gestion des boucles est donc inexistante.

Border Gateway Protocol

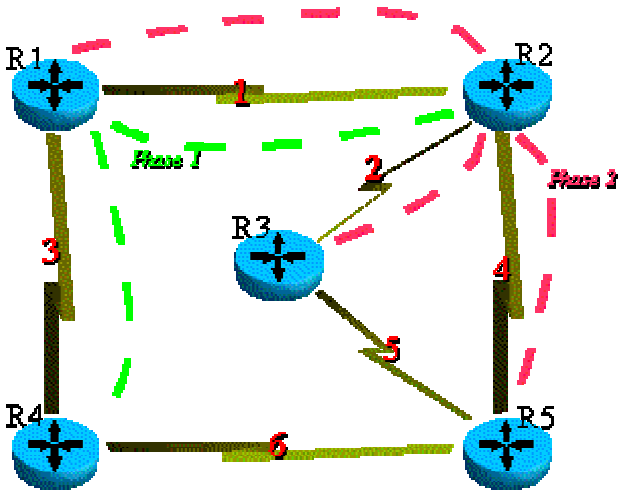
BGP est l'évolution directe d'EGP avec tout d'abord une gestion des boucles. Pour éviter une boucle dans le réseau, BGP fait transiter toute la "carte" du réseau à tous les routeurs qui composent ce réseau, notamment avec le chemin emprunté pour relier les nœuds entre eux. Si un des composants du réseau est présent plus d'une fois dans la liste, il y a vraisemblablement une boucle, donc une erreur à traiter. On peut lui reconnaître d'autres avantages (fiabilité, rapidité, ...).
Le problème de TCP est son mode de contrôle de flux : en effet en cas de problème, TCP réduit immédiatement son débit d' émission en réduisant sa fenêtre de congestion, c'est à dire le nombre d'octets qui peuvent être émis sur une connexion sans attendre un acquittement. Ce principe fonctionne bien pour la plupart des communications, mais pas dans le cas d'un protocole de routage dont la vitesse de convergence est directement dépendante. On fixera donc dans un paquet IP un code de préséance « Internetwork Control », qui modifie la priorité de traitement. A noter qu'en utilisant TCP (mode connecté, donc communication fiable), la mise à jour des tables de routage peut se faire de manière non cyclique.

Protocoles à Vecteur Distance

Dans ce cas chaque routeur d'un réseau est doté d'un numéro, on affecte également un numéro à un lien physique utilisé comme coût (en général la valeur est de 1). Au départ chaque routeur ne connaît que lui même, son vecteur de distance équivaut alors à la valeur 0, et la valeur « infinie » pour toutes les autres destinations.

- A intervalle régulier (ou quand l'état du réseau a changé) les routeurs transmettent leur vecteur distance à chacun de leurs voisins.
- Chaque routeur conserve le vecteur distance le plus récent reçu de la part de chacun de ses voisins.
- Chaque routeur examine le vecteur reçu, et recalcule son propre vecteur distance, en minimisant le coût de chaque destination.
- Le vecteur distance est recalculé à la suite des événements suivants :
  - Envoi par un voisin d'un vecteur distance contenant une information différente de la précédente.
  - Découverte de l'interruption d'une liaison vers un voisin.

Voici les différentes phases d'apprentissage:



Phase 1 : Chaque nœud résume sa propre table en diffusant son vecteur distance à son voisin.

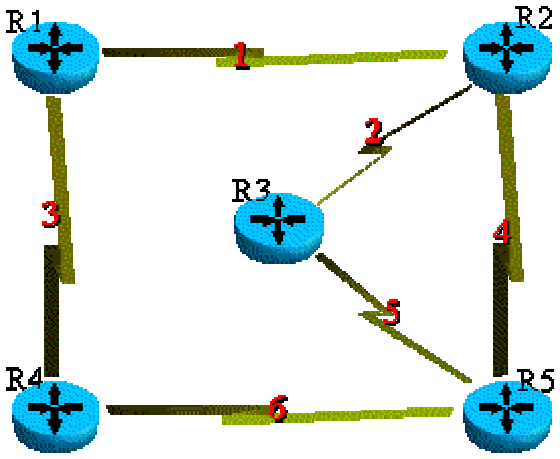
Phase 2 : Chaque nœud qui reçoit une information de mise à jour, conserve la plus petite valeur pour mettre sa propre table de routage à jour, et réémets les modifications.

Etc...

Résumé des différentes phases des tables de routage :

R1	*Lien*	*Coût*	*R2*	*Lien*	*Coût*	*R3*	*Lien*	*Coût*	*R4*	*Lien*	*Coût*	*R5*	*Lien*	*Coût*	*Phase*
R1	-	0	R2	-	0	R3	-	0	R4	-	0	R5	-	0	0
			R1	1	1				R1	3	1				1
R2	1	1				R2	2	1				R2	4	1	
R4	3	1				R1	2	2				R1	4	2	2
												R4	6	1	
			R4	1	2	R5	5	1	R2	3	2	R3	5	1	
			R3	2	1	R4	5	2	R5	6	1				3
			R5	4	1										

Dans cette configuration, chaque routeur a la responsabilité de rentrer en contact avec les routeurs voisins et d'apprendre leurs noms. Chaque routeur construit un paquet connu sous le nom de « paquet d'état de liaison », ou **LSP** (*Link State Packet*) qui contient une liste des noms et des coûts de ses voisins. Le LSP est transmis d'une manière ou d'une autre à tous les autres routeurs, et chaque routeur enregistre le LSP généré le plus récemment par chaque autre routeur. Chaque routeur, qui possède maintenant une carte complète de la topologie, calcule les routes vers chaque destination.



Dans ce cas, chacun des routeurs interroge son voisin afin de connaître son nom, après quoi il construit un paquet particulier dit paquet de liaison (Link State Packet) qui contient la liste des noms connus associés à leur coût. Chaque routeur enregistre alors le LSP le plus récent généré par un routeur donné, pour une liaison connue, ensuite chacun effectue un calcul des routes pour chacune des destinations ainsi découvertes.

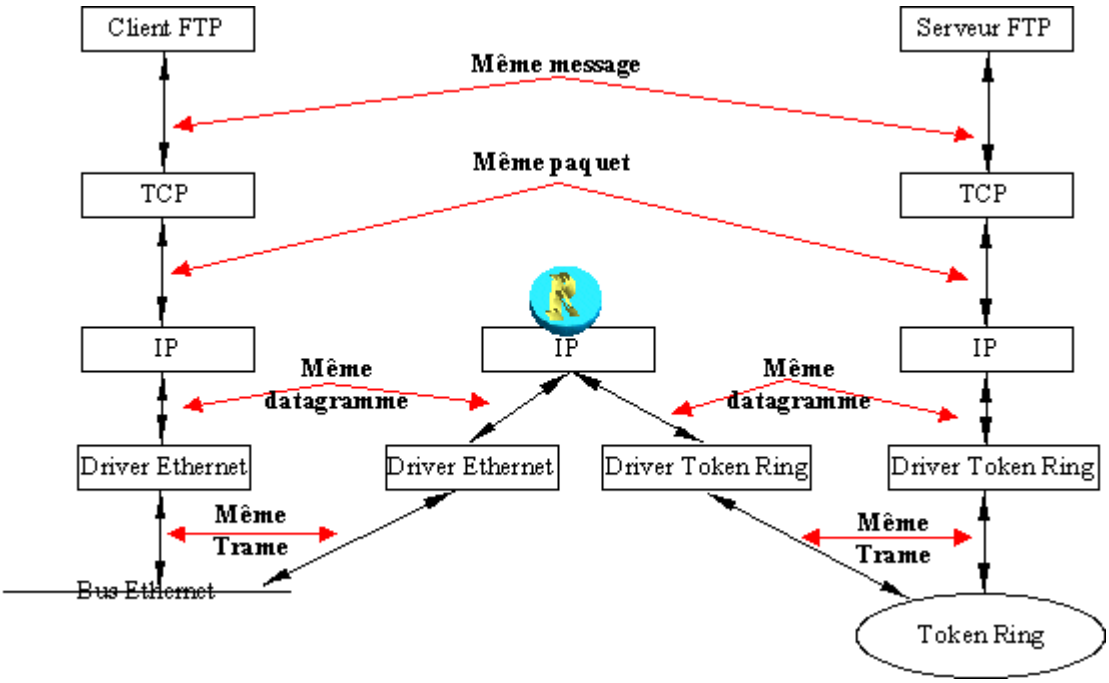
On obtient dans cet exemple :

De	***À***	***Lien***	***Coût***
R1	*R2*	*1*	*1*
R1	*R4*	*3*	*1*
R2	*R1*	*1*	*1*
R2	*R3*	*2*	*1*
R2	*R5*	*4*	*1*
R3	*R2*	*2*	*1*
R3	*R5*	*5*	*1*
R4	*R1*	*3*	*1*
R4	*R5*	*6*	*1*
R5	*R2*	*4*	*1*
R5	*R3*	*5*	*1*
R5	*R4*	*6*	*1*

La gestion des erreurs

Le protocole ICMP (Internet Control Message Protocol) organise un échange d'information permettant d'envoyer des messages d'erreurs à d'autres nœuds. ICMP est « au dessus » de IP, cependant étant indispensable à tous les routeurs on le place au niveau de la couche IP. Le but ICMP n'est pas de fiabiliser le protocole IP, mais de fournir à une autre couche IP, ou à une couche supérieure de protocole (TCP ou UDP), le compte rendu d'une erreur détectée dans un routeur. Un message ICMP est acheminé à l'intérieur d'un datagramme IP, et de ca fait sensible aux erreurs réseaux, mais la règle est qu'aucun message ICMP ne doit être délivré pour signaler une erreur relative à un message ICMP. Ceci évite un rafale d'erreurs en cas de détérioration d'un réseau.

Schématique d'interconnexion de réseaux



IP *v6*

Au début d'IP, un adresse sur 4 octets semblait convenable. Aujourd'hui il n'existe plus d'adresses disponibles en classes A et B, et la classe C s'appauvrit... De plus, plus il sera accordé d'adresse de classe C, plus les tables de routages dans l'Internet grandissent (1 entrée a chaque fois). L'IETF pris conscience du problème en 1991, et organisa des travaux sur l'élaboration d'une nouvelle norme Ipng ou IP version 6. L'adresse Internet sera sur 128 bits soit 16 octets. Cette nouvelle norme fournit une interropérabilité entre les hôtes sous Ipv4 et les hôtes sous Ipv6. Dans le principe, la migration d'une norme à l'autre doit se faire de manière transparente... De plus un certains nombre de services supplémentaires ont été et seront apportés, comme le contrôle de flux, la sécurité, etc ...

Glossaire

AB

ARP : *Adress Resolution Protocol, partie du protocol de TCP/IP qui permet de "lier" une adresse IP ou adresse logique avec une adresse MAC ou adresse physique.*

AUI : *Attachement Unit Interface, connecteur universel 15 broches.*

Bandwidth: *Bande Passante mesurée en milliers de bits par seconde Kbps ou en millions de bits par seconde Mbps, mesure relative au débits de liaisons particulières (T1=1,544 Mbps, numéris 64kbps,...).*

BGP : *Border Gateway Protocol.*

BRI : *Basic Rate Interface. Interface numéris (ISDN=RNIS) composée de 2 cannaux à 64kbps et d'un canal à 16kbps.*

CDE

Classes d'adresses:

A

```
0 1 ... 126       . 0-255     . 0-255     . 0-255
|----- RESEAU -----|---------- HÔTES ----------|
```

B

```
1 0 1 ... 128-191       . 0-255     . 0-255
|---------- RESEAU ----------|----- HÔTES -----|
```

C

```
1 1 0 ... 192-223                   . 0-255
|--------------- RESEAU ---------------| HÔTES |
```

D

```
Réservée au Multicast ( 224 à 239 )
```

E

```
Réservée à un usage particulier. ( 240 à 255 )
```

CHAP : *Challenge Handshake Authentification Protocol, protocole destiné à securiser la connexion entre deux machines.*

CSMA/CD: Carrier Sense Multiple Acces / Collision Detection

CIDR : *Classless Inter-Domain Routing.*

Datagramme: Message concernant la couche 3 du modèle OSI

DHCP : *Dynamic Host Configuration Protocol. Permet d'obtenir de manière dynamique une adresse IP avec différends critères tels que l'authentification, la durée de bail, ...*

DNS : *Domain Name Server. Permet d'attribuer des noms logiques à des machines à la place d'une adresse IP.*

Ethernet : *Est un réseau local qui connecte un ensemble d'hôte (ordianteurs, imprimantes, ...) à l'aide de câble coaxial, ou de paire torsadée.*

EGP : *Exterior Gateway Protocol.*

FGH

FTP : File Transfert Protocol

IJK

ICMP : *Internet Control Messaging Protocol, utilisé par les composants actifs d'un réseau pour rapporter les éventuelles erreurs et contrôles, lors de communications IP*

IGMP : *Internet Group Management Protocol.*

IGP : *Interior Gateway protocol.*

IP : *Internet Protocol, fait partie de TCP/IP. C'est un protocole de routage adapté à la plupart des infrastructure réseau, en terme de taille, de complexité,... . L'adresse IP permet d'identifier un hôte de manière logique sur un réseau. Elle est codée sur 4bytes soit 32bit, divisée en 2 champs, qui identifie d'une part le Réseau, et l'hôte (unique pour un même réseau).*

IPX : *Internet Packet eXchange, protocole dédiée de Netware.*

ISDN : *Integrated Services Digital Network, permet d'envoyer des données et de la voix simultannément à partir d'un même média. Celui-ci utilise plusieurs canaux de communications, 2 baptisés B pour la data d'une taille de 64kbps, puis 1 autre, le D de 16kbps pour le contrôle et la synchronisation.*

LMN

LAN : *Local Area Network*
MAC : *Media Access Control, ensemble de règles permettant la communication entre différends noeuds d'un réseau.*
MAU : Media Attachement Unit.
MAN : *Metroplolitan Area Network*
MIB : *Management Information Base, sorte de répertoire utilisant un certains nombre de conversions en noms logiques des paramètres de configuration d'un équipement informatique.*
MISSALIGNED: Collision en cours d'émission, elle est non divisible par 8, et le crc est faux
NFS : *Network File System*

OPQ

OSI : *Open System Interconnection, références à un modèle standardisé utilisé pour la description définissant le nombre de "composants" d'un réseau ainsi et du rôle des couches fonctionnelles ainsi définies.*
OSPF : *Open Shortest Path First*
Out Of Band: *Canal de communication indépendant*
PAP : *Password Authentification Protocol, utilisé pour définir un nom d'utilisateur et un mot de passe afin de sécuriser un accès.*
Paquet : Message concernant la couche 4 du modèle OSI.
Port : *TCP/IP utilise une information logique définissant une sorte d'emplacement réservé au fonctionnement d'un applicatif ou d'un processus.Ex: 21-FTP, 23-Telnet, ...*
PPP : *Point to Point Protocol, protocole de communication encapsulant (définissant) des paquets de données afin de les envoyer vers un lien unique WAN, ou par des connexions appel du client vers un host (RTC).*

RST

RARP : *Reverse Address Resolution Protocol.*
RIP : *Rounting Informatin Protocol. Protocole d'echange de table de routage basé sur le coût de chaque destination, appelé "saut".*
RPC : *Remote Procedure Call.*
RTC : *Réseau téléphonique commuté.*
SLIP : *Serial Line Internet Protocol, permet d'envoyer des paquets IP par liaison série.*
SMTP : *Simple Message Transfert Protocol, est un protocol de communication dédié à l'échange de message électronique entre utilisateurs de réseaux interconnectés.*
SNMP : *Simple Network Management Protocol, est un protocol de communication dédié à la supervision de la plupart des équipements réseaux (intégré par le constructeur). Celui-ci permet de récupérer diverses informations et de les stocker dans une base de donnée dédiée appelée MIB. Ce protocol intervient par UDP.*
Socket : *Définit un ensemble adresse IP + numéro de port + processus pour une communication TCP/IP.*
Spanning Tree: *Partie importante du "transparent bridging" pour la detection de boucles.*
TACACS: *Terminal Access Concentrator Access Control Server, protocole tres simple d'authentification (uniquement PAP) par question/reponse.*
TCP : *Transmission Control Protocol :*
Protocoles de transports :
- Transmission Control Protocol
- User Datagram Protocol

Protocoles de routage : Contrôler l'adresse et le format d'un paquet pour déterminer le meilleur chemin entre l'émetteur et le récepteur:
- Internet Protocol
- Internet Control Message Protocol
- Rounting Information Protocol
- Open Shortest Path first

Protocoles de routeurs : Permet d'échanger les informations de routage
- Exterior Gataway Protocol
- Gateway to Gateway Protocol
- Interior Gateway Protocol

.../...

Protocoles et services de réseaux : Permet de d'identifier et d'utiliser les noeuds d'un réseau :
- Domain Name Server / System
- Address Resolution Protocol
- Reverse Address Resolution Protocol

Services Utilisateurs :

- BootP
- File Transfer Protocol
- Telnet
- Network File System
- Network Information service
- Remote Procedure Call
- Simple Mail Tranfer Protocol
- Simple Network Management Protocol

TFTP : *Trivial File Tranfer Protocol*
Trame : Message concernant la couche 1.
Transparent Bridging: *Permet de relier 2 réseau de même topologie.*
Translational Bridging: *Permet de relier 2 réseau de topologie différentes.*

UVWXYZ

UDP : *User Datagram Protocol, sous partie de TCP/IP.*
Communique en mode non connecté
WAN : *Wide Area Network*
WINS : *Windows Internet Name Service, produit par MicroSoft afin de permettre aux utilisateurs d'utiliser les noms logiques de machine locales dynamiquement, ce que ne permet pas DNS.*

AB

arp : Permet de modifier le cache su système
Liaison adresse physique et adresse logique

CDE

du / df : estime l'utilisation de l'espace disque des fichiers

FGH

ftp : Permet de tranferer des fichiers d'une machine vers une autre (via le protocole ftp)
commandes : binary / ascii : change le mode de transfert
cd : changer de répertoire sur la machine distante
lcd : change de répertoire sur la machine locale
get : rapatrie un fichier distant : src dst
put : envoie un fichier distant : src dst
ls : liste les entrées distante ...

IJKLMN

netstat : Affiche les connexions actives, tables de routage, statistique sur les interfaces
options : -i Si taux d'erreurs en entrée important : probléme de câbles ?
taux d'erreurs en sortie important : problème de contrôleur ?
rapport collision/paquets.sec supérieur à 1/1000 : Surcharge réseau.
-s Si retourne des erreurs de crc, vérifier le routeur.

nfsstat : affiche les statistiques sur les connexions NFS (Network File System)
les appels RPC = taux d'erreur < 5%

nbtstat : Affiche les statistiques du protocole et les connexions TCP/IP actuelles
utilisant NBT (NetBIOS sur TCP/IP)
options : -a Nom_Machine
-A @IP_machine
-c affiche le cache
-r Nom_machine par WINS

OPQ

ping : Permet de connaître les temps de réponses entre 2 noeuds d'un réseau (via ICMP)

ps : Status des process actifs

RST

rarp : Permet de modifier le cache su système - Liaison adresse logique et adresse physique

route : Permet de modifier la table de routage
ex: route ADD @equipement MASK @subnet @Routeur -p : ajoute une entrée permanente.

telnet : Permet de se connecter pour utiliser une machine distante (via le protocole telnet)

tftp : Permer de tranférer des fichiers, en envoi et en reception, via TFTP (udp)

tracert : Permet d'afficher les liens empruntés par des paquets ICMP

UVWXYZ

Vmstat : Statistiques sur la mémoire virtuelle d'un système