

TCP/IP – Introduction aux protocoles de communication

Formation continue  
Collège de Maisonneuve  
2030, boulevard Pie IX, bureau 430,  
Montréal (Québec) H1V 2C8  
514.251-7131 poste 4800

Document proposé par Louis Savard, M. Ing. TI  
Dernière révision : Juillet 2020

# Table des matières

[Table des matières 2](#_Toc46151365)

[Une histoire de mailles 6](#_Toc46151366)

[Création d’Internet 6](#_Toc46151367)

[Vocabulaire, concepts de base et architecture 8](#_Toc46151368)

[Le modèle OSI 11](#_Toc46151369)

[Sept ou huit? 12](#_Toc46151370)

[Encapsulation 14](#_Toc46151371)

[Un autre modèle 17](#_Toc46151372)

[Couche physique 19](#_Toc46151373)

[Principes 19](#_Toc46151374)

[Bus de communication 20](#_Toc46151375)

[Affectation d’un bit lors de sa transmission 20](#_Toc46151376)

[Techniques de transmission 21](#_Toc46151377)

[Couche Liaison de données 22](#_Toc46151378)

[Principe 22](#_Toc46151379)

[Liste des services de la Couche de liaison de données 23](#_Toc46151380)

[Découpage des données en trames 23](#_Toc46151381)

[Réseau Local (*Local Area Network*) 23](#_Toc46151382)

[Adressage MAC 25](#_Toc46151383)

[Système d’adressage Ethernet 27](#_Toc46151384)

[Adresses MAC (Medium Access Control) 27](#_Toc46151385)

[Trame Ethernet 27](#_Toc46151386)

[Mode de transmission du protocole Ethernet 28](#_Toc46151387)

[Taille des trames Ethernet/IEEE 802.3 28](#_Toc46151388)

[Commutateur Ethernet 28](#_Toc46151389)

[Méthodes de transmission 29](#_Toc46151390)

[VLAN (*Virtual Local Area Network*) 29](#_Toc46151391)

[Assignation des VLAN 30](#_Toc46151392)

[Trane Ethernet 802.1Q (dot1q) 30](#_Toc46151393)

[Couche réseau 32](#_Toc46151394)

[Définition 32](#_Toc46151395)

[Le protocole IP (v4 et v6) 34](#_Toc46151396)

[Fragmentation et assemblage IP 40](#_Toc46151397)

[Masque de sous-réseau (IPv4) 43](#_Toc46151398)

[Fonctions 43](#_Toc46151399)

[Internet Protocol version 6 (IPv6) 49](#_Toc46151400)

[CIDR (Classless Inter-Domain Routing) 54](#_Toc46151401)

[Protocole ARP 55](#_Toc46151402)

[Rôle d'ARP 55](#_Toc46151403)

[Fonctionnement d'ARP 56](#_Toc46151404)

[Formats des paquets ARP 57](#_Toc46151405)

[Cache ARP 59](#_Toc46151406)

[ARP et la passerelle par défaut 59](#_Toc46151407)

[ARP en pratique 60](#_Toc46151408)

[Protocole ICMP 62](#_Toc46151409)

[Messages ICMP 62](#_Toc46151410)

[Multidiffusion 64](#_Toc46151411)

[Définition 64](#_Toc46151412)

[Avantages 64](#_Toc46151413)

[Protocoles 64](#_Toc46151414)

[Utilisation 64](#_Toc46151415)

[Protocole IGMP 65](#_Toc46151416)

[Description de l'en-tête 65](#_Toc46151417)

[Fonctionnement du protocole 65](#_Toc46151418)

[Adressage multidiffusion 67](#_Toc46151419)

[Adresse de groupe de multidiffusion 67](#_Toc46151420)

[Adresse multicast et adresse MAC 67](#_Toc46151421)

[Adresses IP de multidiffusion réservées 68](#_Toc46151422)

[Couche Transport 70](#_Toc46151423)

[Rôle et fonction de la couche transport 70](#_Toc46151424)

[Rôles de la couche transport 70](#_Toc46151425)

[Fonctions de la couche Transport 71](#_Toc46151426)

[Protocole TCP 72](#_Toc46151427)

[Objectif de TCP 73](#_Toc46151428)

[Format des données 73](#_Toc46151429)

[Fiabilité des transferts 75](#_Toc46151430)

[Établissement d'une connexion 76](#_Toc46151431)

[Protocole UDP 78](#_Toc46151432)

[Utilisation 79](#_Toc46151433)

[Protocoles de la couche Applications 81](#_Toc46151434)

[Fonctionnement général 81](#_Toc46151435)

[Fonctions 81](#_Toc46151436)

[Accès Internet 82](#_Toc46151437)

[Messagerie électronique 84](#_Toc46151438)

[Protocoles de gestion des courriels 87](#_Toc46151439)

[SMTP (*Simple Mail Transfer Protocol*) 87](#_Toc46151440)

[Accès distant 89](#_Toc46151441)

[Tunnel SSH 90](#_Toc46151442)

[Transfert de fichiers 90](#_Toc46151443)

[FTP 90](#_Toc46151444)

[Connexions passives et actives 91](#_Toc46151445)

[En résumé … par étapes 92](#_Toc46151446)

[Commandes FTP 93](#_Toc46151447)

[TFTP (*Trivial File Transfert Protocol*) 94](#_Toc46151448)

[Accès réseau 96](#_Toc46151449)

[SSH 96](#_Toc46151450)

[Tunnel SSH 96](#_Toc46151451)

[Telnet 97](#_Toc46151452)

[Protocoles de configuration des hôtes 98](#_Toc46151453)

[Protocole DHCP 98](#_Toc46151454)

[Possibilités offertes 98](#_Toc46151455)

[Fonctionnement 98](#_Toc46151456)

[Requêtes et les messages DHCP 100](#_Toc46151457)

[Port UDP 102](#_Toc46151458)

[Renouvellement du bail 103](#_Toc46151459)

[Adresse privée automatique (APIPA ou zeroconf) 104](#_Toc46151460)

[Réservation d’adresse IPv4 104](#_Toc46151461)

[Client et serveur sur des segments différents 104](#_Toc46151462)

[Protocole DNS 105](#_Toc46151463)

[Concepts de base 105](#_Toc46151464)

[Processus simplifié de résolution de noms 106](#_Toc46151465)

[Nom de domaine 106](#_Toc46151466)

[Résolveur 107](#_Toc46151467)

[Type de requêtes DNS 108](#_Toc46151468)

[Termes essentiels 109](#_Toc46151469)

[Fonctionnement de la résolution de nom 111](#_Toc46151470)

[Gestion de périphériques réseau 115](#_Toc46151471)

[SNMP (Simple Network Management Protocol) 115](#_Toc46151472)

[Protocole NTP 117](#_Toc46151473)

[Références 120](#_Toc46151474)

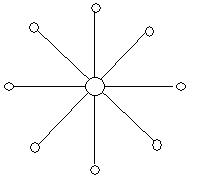
# Une histoire de mailles

## Création d’Internet

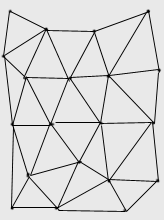
Internet a été créé au départ pour une raison bien particulière.

À l'époque, dans les années 1950, les communications étaient de point à point, c'est-à-dire qu'on ne pouvait communiquer qu'avec une seule machine à la fois. Les chercheurs qui devaient communiquer avec plusieurs autres chercheurs lors de réunions, se sont rendu compte qu'il serait intéressant de pouvoir le faire en temps réel plutôt que de passer d'un interlocuteur à l'autre successivement.

Ils ont donc cherché à créer un nouveau moyen de communication qui ne serait alors plus centralisé, mais maillé.



Cela veut dire que toute information pourrait passer par différents points, et que si certains points disparaissaient, cela n'empêcherait pas l'information de circuler. La figure suivante permet de constater qu'avec un réseau de communication maillé, si un point de communication n'est plus en état de fonctionner, l'information peut passer par un chemin différent.



Les chercheurs vont donc travailler et notamment mettre en place un réseau pour l'armée. C'est seulement au début des années 1960 que l'on voit apparaître des textes décrivant les prémices de ce que sera Internet.

À la fin des années 1960, l'Arpanet, l'ancêtre d'Internet, ne comportait que quatre machines ! Les protocoles utilisés alors ne permettaient pas d'atteindre les buts fixés, à savoir de faire dialoguer des machines provenant de différents réseaux en utilisant différentes technologies de communication.

C'est alors que les chercheurs se sont orientés vers la création d'autres protocoles de communication, et notamment TCP/IP. Internet a continué de croître au fil des années, mais c'est en 1990 qu'une révolution va permettre sa croissance réelle : le langage HTML et le protocole d'échange HTTP qui permettent la création de pages web.

Tout va s'accélérer alors avec la création des premiers navigateurs capables d'afficher des images, et la libération de l'utilisation des noms de domaine. Nous pouvons voir sur la figure suivante la progression phénoménale d'Internet dans les années 1990-2000.

# Vocabulaire, concepts de base et architecture

Le terme de réseau désigne l’organisation des connexions, appelées voies de communication, entre les différents nœuds d’un système téléinformatique. On distingue deux grandes catégories de nœuds …

* **Hôtes au sens large**;
* **Équipements remplissant des fonctions spécifiques** dans le réseau.

Le but d’un réseau est d’échanger des informations d’une entité à une autre via un canal de transmission. La transmission entre deux entités communicantes est caractérisée par …

* le **sens des échanges**unidirectionnel  
  bidirectionnel à l’alternance   
  ou   
  simultané
* le **mode de transmission**en série  
  en parallèle
* la **synchronisation**mode synchrone  
  mode asynchrone

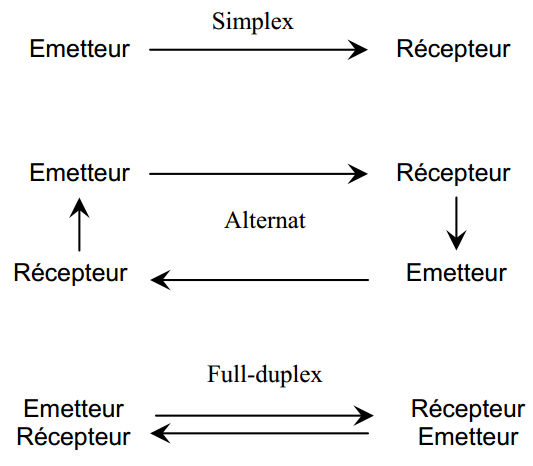
Pour que l’échange des données fonctionne …

* un **codage des signaux de transmission** doit être choisi;
* des **règles communes régissant la communication** doivent être adoptées (notion de protocole)

**Sens de transmission**

Il existe différentes possibilités de sens de transmission entre deux points …

* **Mode simplex**la transmission est unidirectionnelle, de l’émetteur vers le récepteur  
  (exemples : diffusion radio et TV)
* **Mode semi-duplex** (*half* *duplex*) ou bidirectionnel à l’alternat, permet une transmission dans les deux sens mais alternativement (jamais simultanément)
* **Mode duplex**(*full* *duplex*) ou bidirectionnel simultané permet une transmission simultanée dans les deux sens.



**Mode d’acheminement**

Deux modes sont possibles pour la communication entre deux entités …

**Le** **mode avec connexion**  
En mode connecté une entité ne peut émettre d’informations sans avoir, au préalable, demandé à son homologue avec lequel elle veut communiquer, la permission de lui envoyer les blocs d’informations.   
Le récepteur a le choix d’accepter ou de refuser la connexion.  
Le mode avec connexion fait appel à 3 phases distinctes …

* **Négociation** **et établissement de la liaison** entre les deux entités;
* **Transfert des données** de l’utilisateur d’une entité à l’autre en utilisant le chemin fixé (circuit virtuel);
* **Libération de la connexion** (suppression du circuit virtuel précédemment créé).

L’avantage de ce mode réside dans la **sécurisation du transport de l’information** : les émetteurs et récepteurs se mettent d’accord de telle sorte que l’ensemble de l’activité du réseau est contrôlable facilement du moins au niveau des nœuds extrémités. Ils peuvent de plus, au moment de la connexion, échanger des paramètres pour équilibrer la transmission : c’est la négociation de la **qualité de service** ou QoS (*Quality of Service*). Les défauts : - la lourdeur de cette mise en œuvre : même pour envoyer quelques octets, il faut mettre en place une connexion - les accès à des applications multipoints seront difficiles à mettre en œuvre : il faut ouvrir autant de connexions que de points à atteindre (pour diffuser un fichier vers 1 000 utilisateurs distants, il faudra ouvrir 1 000 connexions)

**Le mode sans connexion**

Dans un réseau à commutation de paquets, la transmission en mode non-connecté ou transmission en mode sans-connexion est une transmission de données dans laquelle chaque paquet est préfixé par un entête contenant une adresse de destination, suffisante pour permettre la livraison autonome du paquet, sans recours à d'autres instructions.

Dans une communication orientée connexion, les stations qui sont prêtes à échanger des données doivent d'abord se déclarer comme voulant effectivement le faire. Ceci est appelé l'établissement d'une connexion. Une connexion est parfois définie par une relation logique entre les parties échangeant des données.

Un avantage du mode non-connecté par rapport au mode connecté est qu'il permet les opérations de multidiffusion et de diffusion générale (*broadcast*), qui peuvent économiser encore plus de données quand la même donnée doit être transmise à plusieurs destinataires. Toutefois, une connexion est toujours de type unicast (point à point).

Dans une transmission en mode non connecté d'un paquet, le fournisseur du service de transmission ne peut garantir qu'il n'y aura pas de perte, d'insertion d'erreurs, de mauvaise livraison, de duplication, ou de dé-séquencement de la livraison des paquets. Malgré tout, ces risques peuvent être réduits en fournissant un service de transmission fiable à une couche de protocole de plus haut niveau dans le modèle de référence OSI

# Le modèle OSI

Toute communication humaine implique un certain nombre d’intervenants. On pourrait les identifier comme suit:

* l’**émetteur**  
  (celui qui envoie le message);
* le **récepteur**   
  (celui qui est l’objet du message);
* l’**encodage**   
  (manière dont le message est structuré – deux individus qui parlent français);
* le **canal**   
  (lien qui propage les données – l’air pour une communication verbale);
* les **interférences ou bruits**   
  (ce qui vient influencer négativement la transmission du message – la mauvaise qualité d’une ligne pour un échange téléphonique).

Par analogie on pourrait caractériser les communications informatiques de la même manière. Il est possible d’identifier les principaux intervenants comme étant à peu près les mêmes.

Cependant, tout comme les communications humaines, on note la présence de différents types d’encodage, de multiples canaux de communication, de plusieurs sources de bruits. Cette multiplicité dans le domaine des réseaux informatiques se matérialise pas un nombre phénoménal de standards.

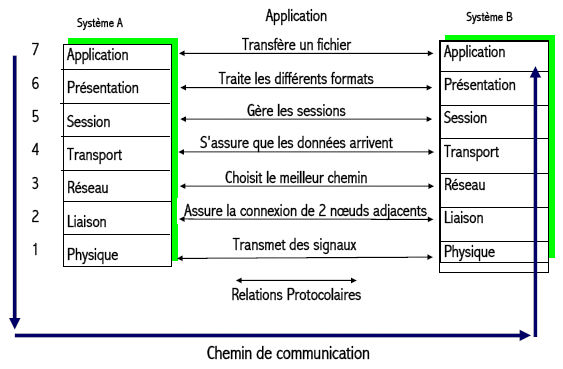
Afin de guider les différents intervenants pour le développement de leurs produits, l’organisation « International Standards Organisation » (ISO), à la fin des années 70, anticipant la nécessité de normes en matière de réseaux informatiques, établit le modèle de référence « *Open System Interconnect* ».

Ce modèle permet d'illustrer comment l'information circule dans un réseau informatique.

Originalement le modèle proposé par l’organisation de normalisation possédait sept couches, chacune des couches définissant les différentes fonctions propres à un réseau informatique.

## Sept ou huit?

La première partie de ce document vise à présenter le modèle de référence OSI.  
Les sept couches du modèle OSI original sont :



De manière générale, chacune couche du modèle OSI joue un rôle distinctif dans la préparation et la livraison des messages à transmettre, et dans le traitement des messages reçus.

Du côté émetteur (celui qui initialise la connexion), chaque protocole des couches du modèle ajoute une information qui lui est propre aux données à transmettre (un entête). On désigne cette étape comme l’encapsulation. Il passe alors les nouvelles données au protocole de la couche inférieure.

À la réception le protocole correspondant du côté du récepteur parcoure l’entête qui lui est propre, interprète l’information ajoutée qui lui indique quoi faire avec les données, retire l’entête (désencapsule) et passe les données à la couche supérieure.

Cette opération peut se produire des centaines de fois lors de l’envoi d’un courriel.

Les avantages introduit par modèle de référence sont les suivants:

* **Offrir un cadre de travail clairement établi** pour le développement d’applications et de mécanique ;
* **Réduire les difficultés propres à la mise en réseau** **par la différenciation de ses fonctions** ;
* **Séparer les différents éléments de l'interconnexion** de réseaux informatiques en sous-ensembles d'opérations distincts, plus aisés à comprendre ;
* **Isoler les différents aspects afin de prévenir que les changements dans un aspect ne modifient pas les autres** aspects, et ainsi donner la possibilité à chacun d'évoluer plus rapidement ;
* **Permettre une communication entre systèmes semblables ou dissimilaires** par l’entremise d’une architecture standard pouvant être adopté par les différents manufacturiers en matière de mise en réseau ;
* **Aider au dépannage** en précisant les principaux problèmes potentiels.

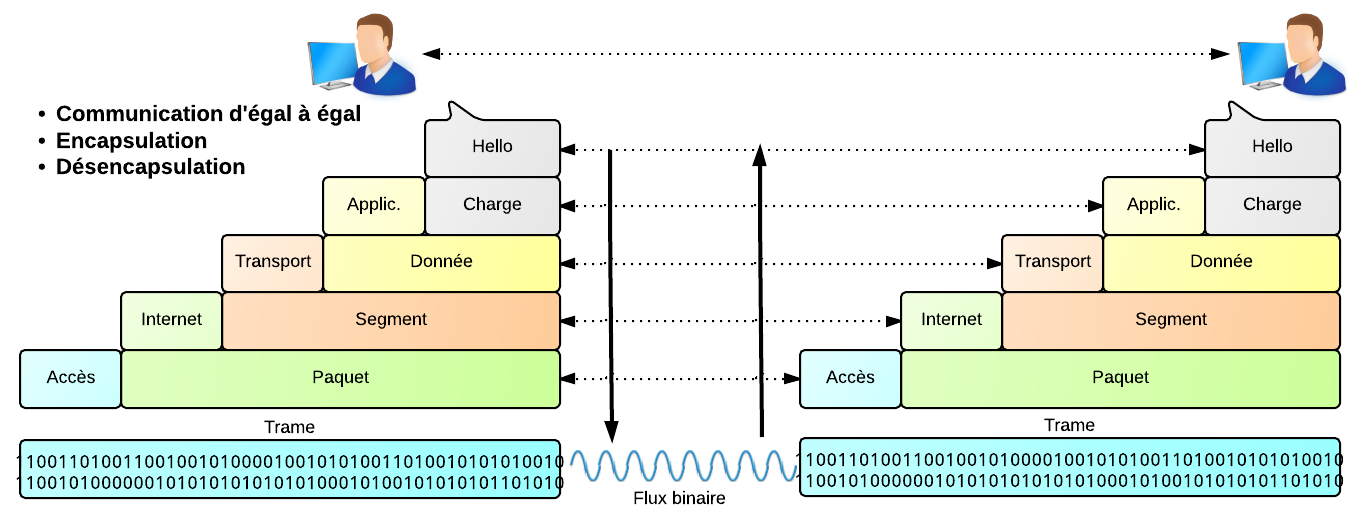
En général, les fonctions de chaque couche peuvent être brièvement décrites comme suit …

* **Couche Physique (*Physical*)**La couche Physique a deux responsabilités …   
  l’envoi et la réception de bits entre divers équipements.   
  Cette couche précise les spécifications électriques, mécaniques, procédurales et fonctionnelles des interfaces réseaux et des médias de communication.   
  Ces spécifications sont définies pour établir, maintenir, et désactiver un lien physique entre deux systèmes.  
  Au niveau de cette couche les données sont identifiées comme étant des trames.
* **Couche Liaison de données (*Data Link*)**  
  Comme son nom l’indique, la couche Liaison de données fait le lien entre la réception physique des données et les couches supérieures, et vice-versa.  
  C’est à cette couche que le destinataire physique et l’expéditeur physique sont identifiés et ce à l’aide de leur adresse MAC.  
  C’est à cette couche que se fait la mise en trame ou organisation logique des bits pour être acheminés sur le média de communication.   
  Cette couche, lors d’une modification ultérieure du modèle original (spécification IEEE 802.2) fût divisée en deux …
  + La sous-couche **Contrôle de lien logique** (LLC ou *Logical Link Control*) qui devient alors réellement le lien ou l’interface entre la couche Physique et la couche Réseau.
  + La sous couche **Contrôle d’accès au média** (MAC ou *Medium Access Control*) s’occupe quant à elle de contrôler l’accès au média, comment les données sont déposées sur le médium.   
    Cette couche détermine principalement les spécifications de la contention (Ethernet ou 802.3) et le passage de jeton (Token Ring ou 802.5)[[1]](#footnote-2).
* **Couche Réseau (*Network*)**  
  Les protocoles propres à cette couche (dont le plus universel est sans doute le protocole IP) permettent d’assurer la communication entre des nœuds informatiques branchés à différents réseaux.   
  Ils sont également responsables d’établir le meilleur chemin à prendre pour une transmission.  
  La principale fonctionnalité de cette couche est sans doute l’aiguillage des paquets par les différents intervenants entre l’émetteur et le récepteur.  
  On retrouvera également à la couche Réseau les différents protocoles qui permettent de vérifier l’état du réseau.  
  À noter que les protocoles de cette couche ne garantissent pas la livraison des paquets.
* **Couche Transport (*Transport*)**  
  La couche Transport joue un rôle de contrôle dans l’expédition des données.   
  À ce niveau, le message à transmettre est fragmenté et réassemblé une fois rendu à destination.   
  Ces fragments de message sont souvent appelés paquets ou datagrammes.   
  La transmission de ces paquets peut être effectuée avec assurance de livraison ou non.
* **Couche Session (*Session*)**  
  La couche Session établit le dialogue entre les couches Présentation de l’émetteur et du récepteur.   
  Cette couche a principalement comme fonctions la gestion de demande de service (et de réponse à ces demandes).  
  C’est à cette couche que sont établi les principales modalités de la communication (comme les modalités d’établissement du lien par exemple).
* **Couche Présentation (*Presentation*)**  
  La couche Présentation est responsable de la conversion et de la mise en forme de données afin qu'elles soient compréhensibles pour les autres couches et le destinataire final.   
  Cette couche intervient également lors de la conversion, la compression et décompression et enfin le cryptage et la remise en forme originale des données.
* **Couche Applications (*Applications*)**La couche Applications est une sorte de porte d’entrée qui permet à un programme tel qu’un fureteur (*browser*), une application de connexion ;a distance (client SSH) ou un logiciel de gestion de courriels d’accéder aux données d’une autre application par l’entremise d’un réseau.  
  Il est important de noter que la couche Applications propose des points ou protocoles d’entrée universels comme http pour les furteurs ou SMTP pour les logiciels de courriels.

**Note …**  
Pour bien illustrer qu’une communication est toujours établie entre deux intervenants et qu’elle peut être visualisée dans les deux sens (données qui descendent ou qui remontent), le tableau en annexe A vous propose une lecture de haut en bas (de la couche 7 à la couche 1).

## Encapsulation

L’encapsulation est une mécanique qui fait qu’un protocole du côté émetteur ajoute ses données de contrôle afin de guider le protocole du côté récepteur à bien interpréter le contenu des données reçues.   
Dans ces données de contrôle on peut retrouver la provenance des données ou le protocole de la couche supérieure, la version du protocole utilisée, les contrôles d’erreurs, …



Prenons un exemple …  
Si un hôte A veut envoyer des données (courriel, fichier de données, …) à un hôte B, les données sont en premier lieu préparées lors d'un processus appelé encapsulage.   
En descendant dans les couches du modèle OSI, des entêtes sont ajoutés aux données. Ces entêtes contiennent des renseignements permettant d’indiquer les spécifications propres au protocole utilisé au niveau de cette couche.

Il serait possible de résumé le processus de la manière suivante …

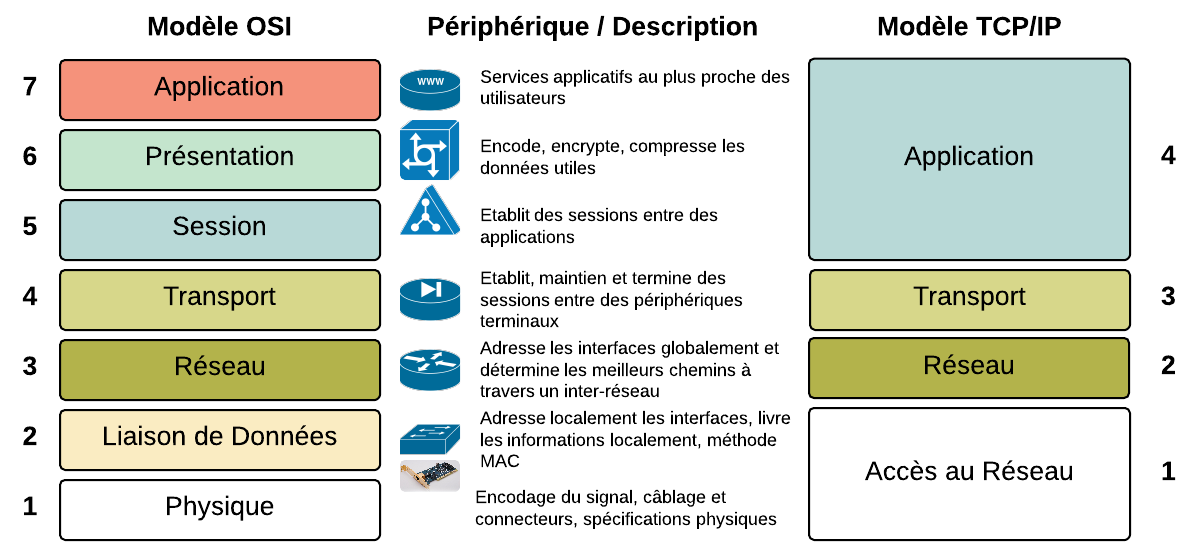
1. **Construction des données**.   
   Un usager envoie un courriel.   
   Les caractères alphanumériques qu'il contient sont convertis en données.
2. **Préparation des informations pour le transport de bout en bout**.  
   Les données sont organisées pour la livraison.   
   Les segments sont utilisés permettant ainsi de s'assurer que les hôtes de chaque extrémité peuvent échanger des données de façon sûres.
3. **Ajout de l'adresse réseau et des informations propres au protocole à l'entête**.   
   Les données sont fragmentées en paquets, chacun des paquets contenant un entête constitué des adresses logiques de source et de destination.   
   Ces adresses aident les intermédiaires à aiguiller les paquets à travers le réseau.
4. **Ajout de l'adresse locale à l'entête de Liaison de données**.   
   Le paquet est à cette étape placé dans une trame.   
   La trame permet d'établir le lien avec le prochain intermédiaire.
5. **Conversion en bits pour la transmission**.   
   La trame est enfin convertie en une série d’un et de zéro (bits) pour la transmission sur le média (généralement un fil).

On constate que des données d'entête et de fin sont ajoutées au fur et à mesure que les données descendent dans les couches du modèle.   
Toutes les informations contenues dans les différents entêtes seront décodées (ou désencapsulées) par l’hôte récepteur. De cette manière le destinataire pourra interpréter et savoir exactement ce qu’il aura à faire.

## Un autre modèle

Le modèle OSI n’est pas le seul modèle proposé. On peut en trouver bon nombre. Plusieurs sociétés, dont Microsoft, Novell, IBM, propose leur propre modèle.

Dans la seconde partie de ce document nous allons étudier un protocole de la couche Réseau. Ce protocole, IP, est un protocole de la famille TCP/IP. C’est pourquoi il est intéressant dans un premier temps de comparer le modèle TCP/IP au modèle OSI.

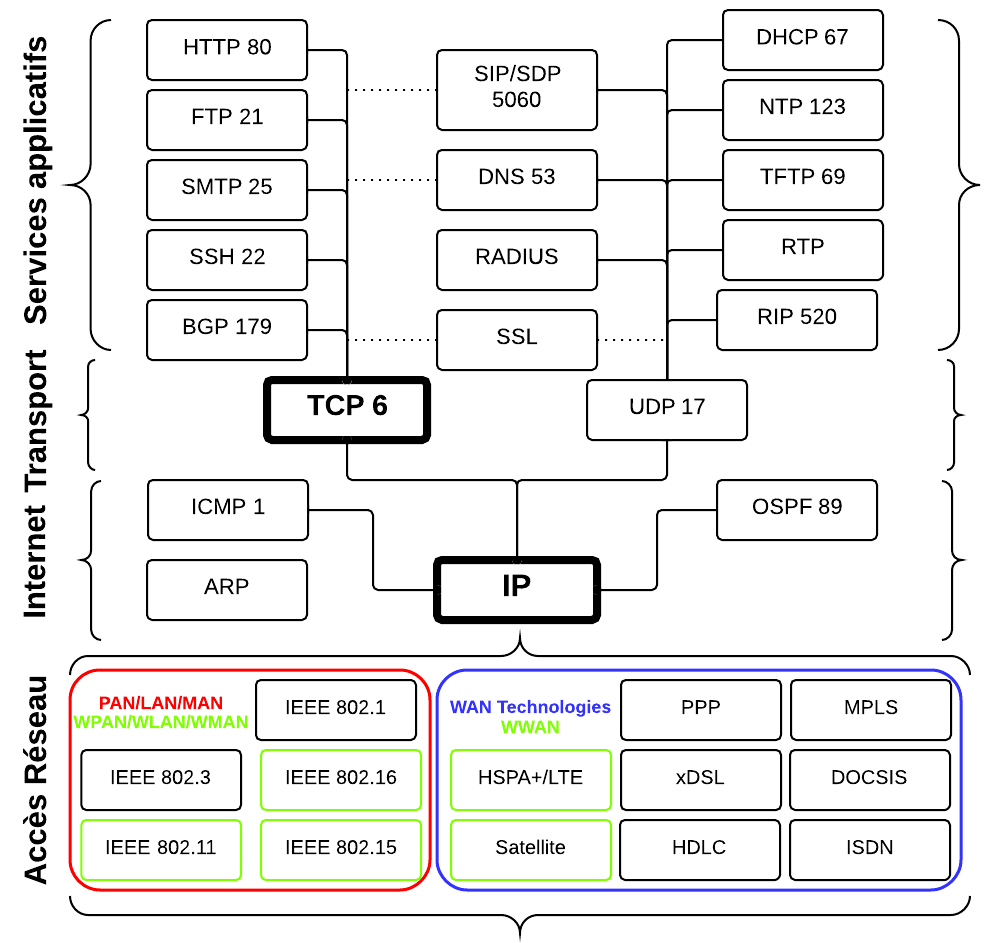


La partie de gauche représente le modèle OSI et la partie de droite représente le modèle TCP/IP et ses principaux protocoles.

Les différences pourraient se résumer comme suit …

* **Les couches 1 et 2 du modèle OSI correspondent à une seule couche du modèle TCP/IP**.   
  Malgré qu’il soit multi-platteforme, le protocole réseau IP a été conçu pour utiliser de préférence les réseaux de type Ethernet.   
  C’est pourquoi la couche accès réseau est au contraire du modèle OSI peu détaillé.
* **Les couches 5, 6 et 7 du modèle OSI sont fusionnées dans le modèle TCP/IP**.   
  Les protocoles de la couche Applications du modèle TCP/IP prennent en charge les fonctions comme l’ouverture de session, l’encodage, …, fonctions explicitement séparées dans le modèle OSI.

Il serait important de préciser à nouveau que le modèle OSI a été proposé dans le but de standardiser les étapes de communication entre deux hôtes.   
Le modèle TCP/IP quant à lui impose une série de protocoles qui par définition sont les mêmes pour tous les hôtes du réseau Internet. Ceci explique pourquoi le modèle est plus simple.



# Couche physique

Dans le domaine des réseaux informatiques, la couche physique est la première couche du modèle OSI (de l'anglais *Open Systems Interconnecti*on ou Interconnexion de systèmes ouverts).

## Principes

La couche physique est chargée de la transmission effective des signaux électriques, radiofréquences ou optiques entre les interlocuteurs.

Cette couche physique s'occupe de la transmission physique des données entre deux équipements réseaux. Elle s'occupe de tout ce qui a trait au bas-niveau, au matériel …

* la **transmission des bits** ;
* leur **encodage** ;
* la **synchronisation entre deux interfaces réseau** ;
* ...

Elle définit les standards des câbles réseaux, des fils de cuivre, du WIFI, de la fibre optique, ou de tout autre support électronique de transmission.

Il existe trois catégories principales de lien de couche physique, selon le type de distance physique qu'ils permettent de faire circuler ...

* les **supports filaires** permettent de faire circuler une grandeur électrique sur un câble généralement métallique ;
* les **supports aériens** désignent l'air ou le vide, ils permettent la circulation d'ondes électromagnétiques ou radioélectriques diverses ;
* les **supports optiques** permettent d'acheminer des informations sous forme lumineuse.

Son travail est généralement limité à l'émission et la réception d'un bit ou d'un train de bits continu   
(notamment pour les supports synchrones comme la fibre optique).

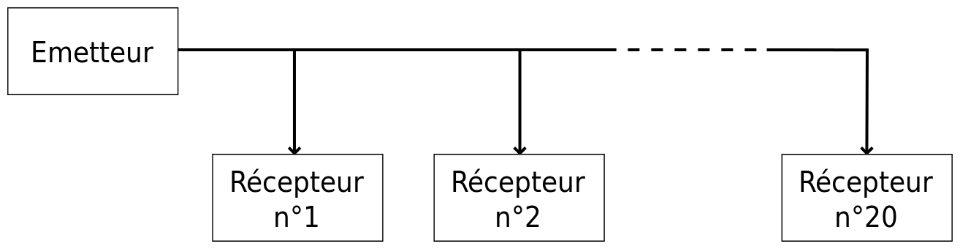
La couche physique est chargée de la conversion entre bits et signaux électriques ou optiques.  
Elle est en pratique toujours réalisée par un circuit électronique spécifique.

**Remarque** …  
Tous les supports de transmission réseau ne transfèrent qu'un bit à la fois.   
En termes techniques, on dit que ce sont des liaisons séries.   
Il existe des liaisons qui échangent plusieurs bits en même temps, les liaisons parallèles, mais ils ne sont pas beaucoup utilisés dans le domaine du réseau. Ils sont plus utilisés à l'intérieur des circuits électroniques, dans les ordinateurs ou pour relier un PC à un périphérique.   
Pour résumer, les transmissions réseaux sont des flux de bits, transmis un par un, à la file indienne.

### Bus de communication

Sur un bus, on peut connecter un nombre assez important de composants, qui dépasse largement les deux nœuds d'une liaison point à point. Avec un bus, un émetteur va envoyer ses données à tous les autres récepteurs.

Sur tous ces récepteurs, il se peut que seul l'un d'entre eux soit le destinataire du paquet : les autres vont alors ignorer le paquet, seul le destinataire traitant le paquet.   
Toutefois, il se peut qu'il y ait plusieurs récepteurs comme destinataires : dans ce cas, les destinataires vont tous recevoir la donnée et la traiter.   
Ces bus permettent donc de faire des envois de données à plusieurs composants en une seule fois.



### Affectation d’un bit lors de sa transmission

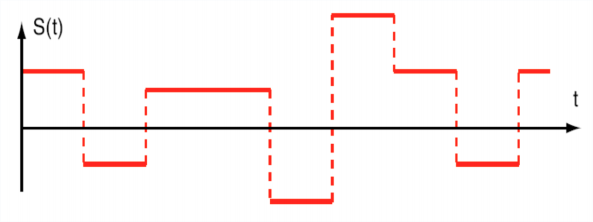
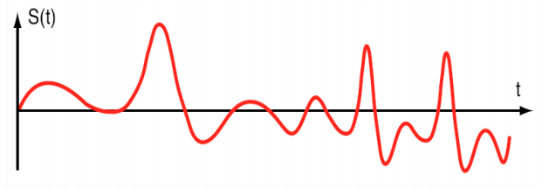
Il existe plusieurs phénomènes qui influence la qualité des transmissions.   
Ces éléments perturbateurs peuvent, soit diminuer l’efficacité du support de transport, soit le rendre totalement inopérationnel.

Parmi les plus importants, on note …

* **Propagation**  
  Temps mis par un bit pour se déplacer dans le média ;
* **Atténuation**  
  Perte de la force (amplitude) du signal ;
* **Bruit**  
  Ajout indésirable d’énergie à un signal causé par des sources d’énergie se trouvant à proximité ;
* **Dispersion**  
  Étalement des impulsions dans le temps ;
* **Gigue**  
  Variation du délai de transfert de l’information ;
* **Latence**  
  Retard de transmission causé par le temps de déplacement d’un bit dans le média et la présence de circuits électroniques dans le cheminement ;
* **Collisions**  
  Phénomène qui existe lorsque deux ordinateurs utilisant le même segment de réseau émettent en même temps.

### Techniques de transmission

Pour transmettre des données, on peut soit utiliser un …

* Signal numérique variant de manière discontinue dans le temps ...  
  
* Signal analogique variant de manière continue dans le temps ...  
  

# Couche Liaison de données

La couche de liaison de données est la couche de protocole qui transfère des données entre les nœuds adjacents d'un réseau étendu (WAN) ou entre des nœuds sur le même segment d'un réseau local (LAN).

La couche de liaison de données fournit les moyens fonctionnels et procéduraux pour le transfert de données entre des entités d'un réseau et, dans certains cas, les moyens de détecter et potentiellement corriger les erreurs qui peuvent survenir au niveau de la couche physique.

## Principe

D'après la sémantique de l'architecture réseau OSI, les protocoles de liaison de données répondent aux demandes de services provenant de la couche de réseau et réalisent leur fonction par l'envoi de demandes de services à la couche physique.

La couche de liaison de données s'occupe de la livraison locale de trames entre dispositifs présents sur un même LAN. Les trames de liaison de données, comme sont nommées ces unités de données de protocole (PDU), ne franchissent pas les limites du réseau local. Le routage inter-réseau et l'adressage global sont des fonctions de couches supérieures, permettant aux protocoles de liaison de données de se focaliser au niveau local, de la livraison, de l'adressage et de l'arbitrage du support. Ainsi, cette couche est analogue à un policier qui ferait la circulation ; il tente d'arbitrer entre les différentes parties s'affrontant pour l'accès au moyen de communication.

Quand des appareils essaient d'utiliser simultanément un support, des collisions de trame surviennent. Les protocoles de liaison de données spécifient comment les appareils détectent et se remettent de telles collisions, mais ils ne les empêchent pas d'arriver.

La livraison de trames par des appareils de couche 2 est établie par l'utilisation d'adresses non-ambiguës de matériel. Un entête de trame contient l'adresse source et destination indiquant de quel appareil provient la trame et quel appareil est censé la recevoir et la traiter. À la différence des adresses routables et hiérarchiques de la couche réseau, les adresses de la couche 2 sont plates, c'est-à-dire qu'il n'y a pas de parties de cette adresse qui pourraient être utilisées pour identifier le groupe physique ou logique auquel l'adresse appartient.

La liaison de données fournit ainsi un transfert de données à travers la liaison physique. Ce transfert peut être fiable ou non ; plusieurs protocoles de liaison de données n'ont pas de mécanismes d'accusé réception et acceptation des trames, et certains protocoles de liaison de données peuvent même n'avoir aucune sorte de somme de contrôle pour tester s'il y a eu des erreurs de transmission. Dans ces cas-là, des protocoles de plus haut niveau doivent fournir du contrôle de flux, de la détection d'erreur, de l'accusé réception et de la retransmission.

Dans certains réseaux, comme les réseaux locaux IEEE 802, la couche de liaison de données est décrite plus en détail avec des sous-couches de contrôle d'accès au support (Media Access Control - MAC) et de contrôle de la liaison logique (Logical Link Control - LLC) ; cela signifie que le protocole LLC IEEE 802.2 peut être utilisé avec toutes les couches MAC IEEE 802, comme Ethernet, Token ring, IEEE 802.11, …, autant qu'avec certaines couches MAC non-802 comme FDDI.

Ethernet pour les réseaux locaux (multi-nœuds), le protocole point à point (PPP), HDLC et ADCCP pour des connexions point à points (double nœud) sont des exemples de protocoles de liaison de données.

## Liste des services de la Couche de liaison de données

Les services offerts pas la couche Liaison de données sont …

* **Encapsulation des paquets de données** **de la couche réseau** dans des trames ;
* **Synchronisation des trames** ;
* La sous-couche de **Contrôle de la liaison logique** (Logical link control - LLC) …  
  La détection d'erreur et l'annulation de paquets fournis à toutes les couches, notamment la couche réseau ;
* La sous-couche de **Contrôle d'accès au support** (Media access control - MAC) …
  + **Protocoles d'accès multiple pour le contrôle d'accès au canal**, par exemple les protocoles CSMA/CD pour la détection de collision et la retransmission dans les réseaux à bus Ethernet et les réseaux à concentrateurs (hubs), ou le protocole CSMA/CA pour l'évitement des collisions dans les réseaux sans-fils ;
  + **Adressage physique** (adressage MAC) ;
  + La **commutation LAN** (Commutation de paquets) incluant le filtrage MAC et le spanning tree protocol ;
  + La **gestion de files d'attente des paquets de données ou leur ordonnancement** ;
  + La **commutation Store-and-forward ou cut-through** ;
  + Le **contrôle de la Qualité de service** (QoS) ;
  + Les **réseaux LAN virtuels** (VLAN)

### Découpage des données en trames

Le découpage des données en trame donne une signification aux bits qui sont transmis sur le réseau. Elle doit acheminer sans erreur des blocs d'information utilisateur sur la liaison physique ...

* **Contrôle d’intégrité**Détection et de correction d’erreurs élémentaires dues au support physique imparfait et signale à la couche réseau les erreurs irrécupérables ;
* **Reconnaissance des débuts et fin de trames réceptionnées** ;
* **Spécifications des tailles et moyens d’adressage** des trames.

La couche Liaison de données s’assure que deux ou plusieurs nœuds n’essaient pas de transmettre des données sur le canal (partagé) de transmission en même temps. Enfin, elle permet l’dentification des trames par adresse MAC.

Le protocole Ethernet (IEEE 802.3) est le protocole le plus utilisé dans les réseaux locaux.

## Réseau Local (*Local Area Network*)

Un LAN, pour *Local Area Network* est, comme son nom anglais l'indique, un réseau d'aire locale, ou encore un Réseau Local d'Entreprise (RLE). En fait un LAN est un réseau qui possède une couverture géographique très limitée, généralement celle d’un site ou d’un bâtiment.

**Caractéristiques**  
Voici en quelques lignes les caractéristiques essentielles d’un LAN ...

* **Couverture géographique limités**;
* **Hauts débits de transmission** (plusieurs mégabits par seconde -- Mbps).   
  Leur faible couverture géographique permet d'installer des supports physiques d'excellente qualité (affaiblissement faible, peu de diaphonie et paradiaphonie, bande passante élevée) pour la transmission de données.   
  Ces supports sont de la paire torsadée (L120, catégorie 5, etc.), de la fibre optique ou du câble coaxial (de plus en plus rare) ;
* **Réseau de diffusion** (avant l'arrivée des commutateurs (*switchs*).  
  Toute donnée émise sur le réseau est vue par tous les connectés.   
  Lorsqu'une station envoie une trame de niveau 2, cette trame est diffusée à toutes les stations présentes sur le réseau local. Cette opération est facilitée par la topologie logique (et éventuellement physique) du réseau.  
  Les stations sont connectées à un bus ou à un anneau (pour 99% des cas !). On dit que les réseaux locaux fonctionnent sur des supports partagés. Cette caractéristique est extrêmement importante car c'est elle qui oblige à la mise en place d'un adressage au niveau 2.
* **Méthode d'accès au support**  
  Chaque type de LAN met en œuvre une procédure particulière, et normalisée, d'accès au support afin d'émettre les trames. Cette procédure est communément appelée le MAC: Medium Access Control.   
  D'une manière générale étant donné que le support est partagé entre toutes les stations d'un réseau local, seule une trame à la fois est véhiculée par le support. Le MAC veille donc à donner équitablement la parole à chaque station, et bien sûr détecte les éventuelles cacophonies si certaines stations ne respectent pas le jeu.

Pour la suite de ce document, il est important de retenir que ...

* un **LAN est un réseau de diffusion qui partage son suppor**t entre toutes les stations présentes   
  (technologie logique du bus) ;
* un L**AN implémente une procédure de niveau 2 nommée MAC** qui définit un format de trame et gère la méthode d'accès au support ;
* en raison de la caractéristique de diffusion, le **MAC défini un adressage de niveau 2 pour les trames**   
  (adresse de destination et adresse source).

**Principaux LAN**  
La star incontestée des LAN est **Ethernet**.  
Ce type de réseau local représente aujourd'hui plus de 80% du parc.   
Il se décline en plusieurs débits, 10, 100, 1000 et 10 000 Mbps.   
Les appellations courantes sont respectivement Ethernet, Fast Ethernet, Giga Ethernet et 10 Giga Ethernet.

Un autre réseau local assez connu, mais en sérieuse perte de vitesse est le **Token Ring** (Anneau à jeton).  
 C'est une production d'IBM. Belle technique, bonne qualité de service mais pas assez ouvert, trop complexe.   
N'existe que dans les environnements IBM d'ancienne génération (SNA).   
Il se décline en deux versions ... 4 et 16 Mbps.

### Adressage MAC

Puisque sur un LAN, les stations voient passer toutes les trames émises, il faut donc qu'elles puissent différencier celles qui les concernent des autres. Pour cela, elles vont se baser sur l'adresse destination de la trame. Une station ne lira le contenu que des trames qui lui sont adressées.

Quand une trame est adressée à une station unique du réseau local, elle comporte un format d'adresse dit *unicast*. Mais une trame peut également être adressée à un groupe de stations ou à toutes les stations du réseau local. L'adressage est alors respectivement nommé *multicast* et *broadcast*.

La station qui reçoit une trame doit pouvoir y répondre. Chaque trame possède donc une adresse MAC source qui permet au destinataire d'identifier l'émetteur.

L'adressage MAC garanti l'unicité mondiale d'une adresse. L'adresse d'une station est d'ailleurs inscrite « en dur » sur l’interface réseau installée dans la machine.

Cette adresse est définie sur 6 octets.   
Les trois premiers octets indiquent le numéro du constructeur et les trois derniers indiquent le numéro de la carte dans la série du constructeur. Lorsqu’un fabricant veut se mettre à produire des cartes Ethernet, il lui faudra d'abord demander au groupe IEEE une adresse de constructeur.   
Après quoi, il s’engagera à ne pas produire deux interfaces réseau ayant le même numéro sur les trois derniers octets. Ainsi l'unicité de l'adresse est garantie.

Sur 6 octets on a donc ...

* 224 (plus de 16 millions) constructeurs possibles;
* 224 (plus de 16 millions) interfaces réseau par constructeur !

En vérité, pour assurer les adressages *multicast* et *broadcast* certains bits sont bloqués notamment sur la partie du constructeur. Toutefois, cela donne une bonne marge.

Lorsqu'on énonce une adresse MAC, on donne sa valeur en hexadécimal pour chaque octet.   
Par exemple: 00:00:0C:1A:BE:34.

Il faut donc retenir simplement:

* qu'une adresse MAC se définie sur 6 octets ;
* qu'il existe des adresses MAC destinations unicast, multicast et broadcast.  
  L'adresse destination broadcast a pour valeur héxadécimale:  
  FF:FF:FF:FF:FF:FF (tous les bits à 1 !);
* que l'adresse est, généralement, codée en dur sur l’interface réseau de la machine.

**Adresse MAC (Media Access Control)**

Chaque équipement Ethernet possède donc une adresse MAC unique, qui est codée en dur (brûlée) sur le matériel -- BIA (**Burn-In Address**).  
Une adresse MAC identifie un équipement (un nœud) à l’intérieur d’un réseau local.  
Une adresse MAC est constituée de six octets généralement affichés en valeur hexadécimale e.g., 00:0A:CC:32:FO:FD.

**Remarque** …  
L'adresse MAC est écrite sur chaque interface réseau d’un hôte, commutateur (switch), routeur, …

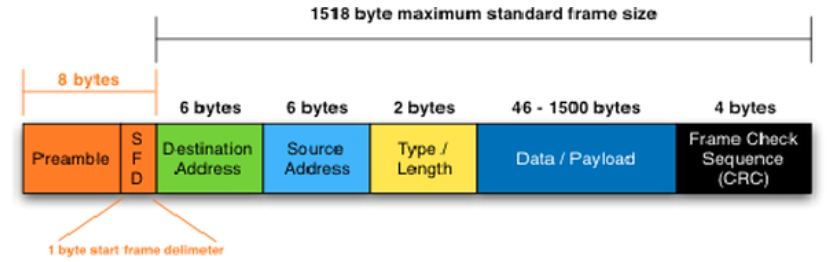
## Système d’adressage Ethernet

### Adresses MAC (Medium Access Control)

* L’adresse de la couche Liaison de données d’un élément de réseau ;
* Format -- 6 octets en hexadécimal séparés par : ou – ;
* L’identifiant OUI (Organization Unique Id) sont les trois premiers octets  
  et  
  L’adresse matérielle spécifique (Product ID) sont les trois derniers octets.  
  

### Trame Ethernet

* **Adresse source**  
  Adresse MAC de la station source (émetteur) ;
* **Adresse de destination**  
  Adresse MAC de la station destinataire (récepteur) ;
* **Type Ethernet**Type de données encapsulées  
  (géralement paquets IP) ;
* **Charge utile** (paylosd)  
  Données transmises.
* **Somme de contrôle (*checksum*)**  
  Parfois appelée empreinte, est un nombre qu'on ajoute à une trame à transmettre pour permettre au récepteur de vérifier que le message reçu est bien celui qui a été envoyé.



### Mode de transmission du protocole Ethernet

* **Diffusion dirigée (*Unicast*)**   
  Adresse MAC -- 0A:00:81:2F:42:51  
  Adresse désignant une seule interface ;
* **Multi diffusion (Multicast)**   
  Adresse MAC -- 01:XX:XX:XX:XX:XX (premier octet impair)  
  Adresse désignant plusieurs stations (groupe de diffusion) ;
* **Adresse de diffusion globale (Broadcast address)**  
  Adresse MAC -- FF:FF:FF:FF:FF:FF  
  Adresse de diffusion qui désigne l’ensemble des stations du domaine de collision.concerné

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Monodiffusion (**unicast**) | Multidiffusion (*multicast*) | Diffusion générale (*broadcast*) |

### Taille des trames Ethernet/IEEE 802.3

* La taille maximale d’une trame Ethernet est de ...  
  1 518 octets ;
* Cette petite taille empêche un nœud de monopoliser le canal pendant trop longtemps ;
* Ceci est une valeur arbitraire. La taille minimale d’une trame est de   
  64 octets ;
* Si la quantité de données transportées ne permet pas de remplir une trame, il faut ajouter des octets de bourrage (*padding*).

### Commutateur Ethernet

Un commutateur réseau (*switch*), est un équipement qui relie plusieurs segments (câbles ou fibres) dans un réseau informatique et de télécommunication et qui permet de créer des circuits virtuels.

La commutation est un des deux modes de transport de trame au sein des réseaux informatiques et de communication, l'autre étant le routage.

Dans les réseaux locaux (LAN), il s'agit le plus souvent d'un boîtier disposant de plusieurs ports RJ45 (entre 4 et plusieurs centaines), il a donc la même apparence qu'un concentrateur (*hub*).

Le commutateur établit et met à jour une table, dans le cas du commutateur pour un réseau Ethernet il s'agit de la table d'adresses MAC, qui lui indique sur quels ports diriger les trames destinées à une adresse MAC donnée, en fonction des adresses MAC source des trames reçues sur chaque port.   
Le commutateur construit donc dynamiquement une table qui associe numéro de port et adresses MAC.

Lorsqu'il reçoit une trame destinée à une adresse présente dans cette table, le commutateur renvoie la trame sur le port correspondant. Si le port de destination est le même que celui de l'émetteur, la trame n'est pas transmise. Si l'adresse du destinataire est inconnue dans la table, alors la trame est traitée comme un *broadcast*, c'est-à-dire qu'elle est transmise à tous les ports du commutateur à l'exception du port d'émission.

Un commutateur de niveau 2 est similaire à un concentrateur (*hub*) dans le sens où il fournit un seul domaine de diffusion. En revanche, chaque port a son propre domaine de collision. Le commutateur utilise la micro-segmentation pour diviser les domaines de collision, un par segment connecté. Ainsi, seules les interfaces réseau directement connectées par un lien point à point sollicitent le medium.   
Si le commutateur auquel il est connecté prend en charge le full-duplex, le domaine de collision est éliminé.

### Méthodes de transmission

La transmission des paquets peut s'opérer selon quatre méthodes ...

* **Mode direct** (*cut through*)  
  Le commutateur lit juste l'adresse du matériel et la transmet telle quelle.   
  Aucune détection d'erreur n'est réalisée avec cette méthode ;
* **Mode différé** (*store and forward*)  
  Le commutateur met en tampon, et le plus souvent, réalise une opération en somme de contrôle sur chaque trame avant de l'envoyer ;
* **Mode sans fragment** (*fragment free*)  
  Les paquets sont passés à un débit fixé, permettant de réaliser une détection d'erreur simplifiée. C'est un compromis entre les précédentes méthodes ;
* **Commutation automatique** (*adaptive switching*)  
  En fonction des erreurs constatées, le commutateur choisit automatiquement un des trois modes précédents.

### VLAN (*Virtual Local Area Network*)

Un réseau local virtuel, communément appelé VLAN est un réseau informatique logique indépendant.   
De nombreux VLAN peuvent coexister sur un même commutateur réseau.

Un VLAN est un ensemble d’unités regroupées quel que soit l’emplacement de leur segment physique. Le VLAN permet de définir un nouveau réseau au-dessus du réseau physique et à ce titre offre les avantages suivants …

* **Plus de souplesse pour l'administration et les modifications du réseau** (e.g. mobilité).  
  Toute l'architecture peut être modifiée par simple paramétrage des commutateurs ;
* **Optimisation de la bande passante** …  
  En séparant les flux un VLAN permet de segmenter un réseau physique afin de réduite le domaine de diffusion générale (*broadcast*) ;
* **Gain en sécurité car les informations sont encapsulées** dans un niveau supplémentaire et éventuellement analysées.  
  Un VLAN permet de créer un ensemble logique isolé pour améliorer la sécurité.   
  Le seul moyen pour communiquer entre des machines appartenant à des VLAN différents est alors de passer par un routeur;
* **Réduction de la diffusion du trafic sur le réseau**.

### Assignation des VLAN

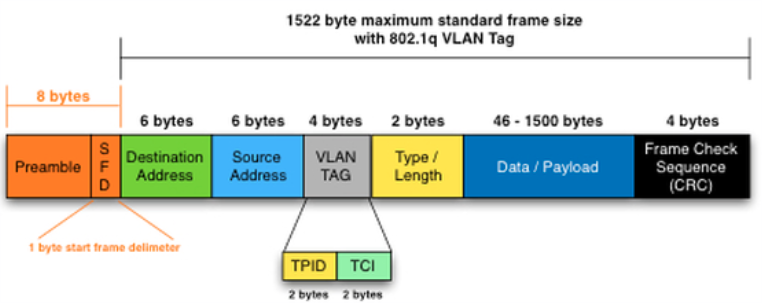
La manière la plus courante de configurer les VLAN est de le faire de manière statique en spécifiant pour chacun des ports du commutateur à quel VLAN il appartient.   
On nomme ce type de configuration VLAN statique ou VLAN de niveau 1.

Il est également possible de modifier dynamiquement la configuration d'un port.   
Cela peut se faire de manière sûre avec un mécanisme tel que 802.1X ou sur base de l'adresse MAC voire IP de l'équipement.   
Cette dernière solution est toutefois très peu sécurisée, les adresses MAC ou IP étant très facilement modifiables.  
On nomme ce type de configuration VLAN dynamique ou VLAN niveau 2.

### Trane Ethernet 802.1Q (dot1q)

La trame IEEE 802.1Q, est une trame Ethernet à laquelle on a ajouté 4 octets de façon à ce que la trame soit identifiée comme appartenant à un VLAN particulier.

C'est le rôle des commutateurs (*switchies*) Ethernet de traiter ces 4 octets afin de déterminer où les trames doivent être délivrées.   
Dans ces 4 octets d'entête, trois bits sont utilisés pour la priorisation des trames.



# Couche réseau

## Définition

La couche réseau construit une voie de communication de bout à bout à partir de voies de communication avec ses voisins directs. Ses apports fonctionnels principaux sont donc …

* le **routage**  
  Détermination d'un chemin permettant de relier les 2 hôtes distants ;
* le **relayage**  
  Retransmission d'un PDU (*Protocol Data Unit* ou Unité de données de protocole) dont la destination n'est pas locale pour le rapprocher de sa destination finale.

Cette couche est donc la seule à être directement concernée par la topologie du réseau.   
C'est aussi la dernière couche supportée par tous les hôtes du réseau pour le transport des données utilisateur : les couches supérieures sont réalisées uniquement dans les machines d'extrémité.

Le PDU de cette couche est souvent appelé paquet. La fonction de relayage (terme OSI) est parfois appelée acheminement.

**Remarques** …  
Cette couche est la plus caractéristique d'une architecture réseau.   
C'est pourquoi l'architecture prend souvent le nom du protocole principal de niveau réseau.

Déterminer un chemin est une tâche complexe normalement réalisée dans les grands réseaux par des protocoles dédiés dont le rôle est de découvrir la topologie du réseau et d'en tirer la meilleure route. Les protocoles de routage se différencient par les critères de choix des routes et la précision de la topologie découverte. En dehors du cas des petits réseaux, le routage est hiérarchique : la précision de la connaissance de l'environnement d'un routeur décroît avec la distance.

Si les routeurs n'ont pas de couche supérieure à la couche réseau du point de vue des hôtes utilisateurs du réseau, ils peuvent supporter des protocoles de niveau transport et au-dessus pour la gestion du réseau (supervision et exécution des protocoles de routage par exemple).   
Bien que les données calculées par les protocoles de routage soient utilisées par la couche réseau, ce ne sont pas des protocoles de niveau réseau car ils ne servent pas à transporter les données des hôtes utilisateurs du réseau. D'ailleurs, dans le monde IP, les protocoles de routage non local (i.e. hors RIP) sont transportés par TCP.

Voici les principaux protocoles de la couche Réseau …

* **Protocoles IP** (v4 et v6) ;
* **Protocole ARP** ;
* **Protocole ICMP** ;
* **Protocole IGMP**.

## Le protocole IP (v4 et v6)

Les protocoles TCP/IP se situent dans un modèle souvent nommé famille de protocoles TCP/IP.  
Les protocoles TCP et IP ne sont que deux des membres de la suite de protocoles IP.

Le protovole IP (*Internet Protocol*) est un protocole qui se charge de l'acheminement des paquets pour tous les autres protocoles de la famille TCP/IP. Il fournit un système de remise de données optimisé sans connexion. Le terme optimisé souligne le fait qu'il ne garantit pas que les paquets transportés parviennent à leur destination, ni qu'ils soient reçus dans leur ordre d'envoi. La fonctionnalité de somme de contrôle du protocole ne confirme que l'intégrité de l'entête IP. Ainsi, seuls les protocoles de niveau supérieur sont responsables des données contenues dans les paquets IP (et de leur ordre de réception).

Le protocole IP travaille en mode non connecté, c'est-à-dire que les paquets émis par le niveau 3 sont acheminés de manière autonome (datagrammes), sans garantie de livraison.

**Principes guidant IP**

Dès l’origine deux principes de base ont guidés les concepteurs d’Internet pour la création du protocole de transport (en occurrence IP). Ces deux principes sont les suivants …

* La **communication de bout en bout**  
  Une communication se fait entre deux partenaires et ce sont ces derniers qui en gèrent les modalités. On sait que la commutation des paquets implique la présence de plusieurs intervenants intermédiaires (souvent désignés comme aiguilleurs dans le langage Internet) et que ces intermédiaires sont transparents et ne participent pas dans la communication (en fait ils ne sont que des relais).  
  Les nœuds d’extrémités sont considérés comme intelligents, permettant ainsi le déploiement d’applications client/serveur à l'aide du réseau.
* La **livraison au meilleur effort**Ce principe implique que tous les éléments intermédiaires (aiguilleurs) de la connexion n’offre aucune garantie quant au transport des paquets entre les nœuds d’extrémités. Ils « font de leur mieux » pour rendre les paquets à destination. IP n’est donc pas selon ce principe un protocole fiable. Toutefois, on le considère comme étant robuste. Agissant ainsi le protocole n’utilise pas de ressources supplémentaires aux intermédiaires pour fournir une qualité de service à priori.   
  Seulement les nœuds d’extrémités vont utiliser un protocole de la couche Transport du modèle OSI pour garantir cette livraison (ce qui n’est pas le cas des nœuds intermédiaires).  
  C’est pourquoi le protocole TCP est implicitement lié au protocole IP.

**Configuration IP**Toute configuration IP doit comporter les données suivantes …

* l’**adresse IP**   
  (unique à l’hôte) ;
* le **masque de sous-réseau** ;
* l’**adresse** **de la passerelle par défaut**   
  (un aiguilleur est traditionnellement appelé ainsi pour le réseau Internet).  
  La passerelle est la porte d’entrée sur le réseau Internet ;
* l’**adresse** **d’un serveur DNS**  
  Le serveur DNS est l’intermédiaire qui résout un nom convivial comme www.profsavard.info en une adresse valide de destination comme 206.167.36.22.

Toute configuration IP valide doit obligatoirement comporter les deux premiers éléments.   
Sans ces informations il sera impossible pour un hôte de communiquer avec un hôte distant (qu’il soit dans son propre réseau ou sur le réseau Internet).   
Quant au troisième il est essentiel si un hôte veut communiquer avec un hôte distant hors de son propre réseau local.   
Enfin, sans le quatrième élément, l’utilisateur devra toujours connaître l’adresse exacte pour rejoindre un hôte en dehors de son réseau local.

**Adresses IPv4**

Pour pouvoir établir une communication avec un autre hôte branché au réseau Internet une machine (ou nœud) doit obligatoirement posséder une adresse IP unique.

Cette adresse peut être caractérisée comme étant statique ou dynamique.

* Une **adresse statique est généralement attribuée manuellement** par un administrateur réseau.   
  Certains nœuds du réseau Internet nécessitent une adresse qui ne sera pas modifiée (ce qui est le cas de la plupart des serveurs).
* Une **adresse dynamique est une adresse qui sera attribuée de manière automatique** lors de l’ouverture d’un hôte.   
  Cette adresse est généralement attribuée par un serveur DHCP[[2]](#footnote-3).

Plusieurs avantages liés à la présence d’un serveur DHCP …

* **Attribution automatique de la configuration IP** à un hôte;
* **Résolution du problème d’un réseau** qui possède plus d’hôtes que d’adresses disponibles.

**Anatomie d’une adresse IPv4**Une adresse IP se divise en deux parties …

* La **première représente la partie réseau**.   
  Il est important de préciser ici qu’un réseau est un regroupement logique de plusieurs hôtes.   
  Cette partie réseau est attribuée par un responsable fournisseur et est unique pour le réseau Internet
* La **seconde partie d’une adresse IPv4 est l’identifiant unique** pour un hôte de ce réseau.   
  Cette partie est gérée localement par les responsables du réseau interne. C’est cette partie qui est attribuée dynamiquement ou statiquement[[3]](#footnote-4).  
  Cette partie identifiant doit être obligatoirement unique au réseau interne.

Une adresse IP peut être publique ou privée …

* Une **adresse publique est une adresse que doit posséder tout hôte qui veut établir une communication sur le réseau Internet**.   
  Ceci permet d’établir une session sans intermédiaire interne (comme un serveur proxy).
* Une **adresse privée est une adresse qui permet aux hôtes d’un même réseau de communiquer entre eux sans toutefois pouvoir ouvrir une session avec un hôte sur le réseau Internet**.   
  Par défaut, les aiguilleurs (la porte de sortie sur le réseau Internet) sont configurés pour ne pas laisser passer les paquets provenant d’un hôte possédant une adresse privée.  
  Le protocole NAT[[4]](#footnote-5) permet de faire le lien entre un hôte possédant une adresse privée et un serveur proxy possédant une adresse publique. Ce serveur proxy devient à la fois traducteur entre adresses privées et adresses publiques et aiguilleur.

**Méthode d’attribution des identificateurs IPv4**

* **Prévoir l’avenir**Tenir compte de la croissance future du réseau   
  (surtout dans la planification des sous-réseaux)
* **Assurer l’unicité**Bien identifier les sous-réseaux
* **Éviter les adresses à accès limité**  
  Adresses de réseau, diffusion, bouclage, …

|  |  |
| --- | --- |
| Dispositifs de réseau | Étendues d’adresses |
| Aiguilleurs | 192.168.0.1 à 192.168.0.5 |
| Stations de travail | 192.168.0.6 à 192.168.0.245 |
| Stations serveurs | 192.168.0.246 à 192.168.0.254 |

Une adresse IPv4 est composée de 32 bits dont une partie sert à identifier le réseau et une partie pour identifier les hôtes composant ce réseau.

|  |  |
| --- | --- |
| Identifiant réseau | Identifiant hôte |
| ç 32 bits è | |

La partie de ces 32 bits attribuée à chacune des parties varie selon la classe.

**Adressage IPv4**

|  |
| --- |
| **Classe A 0RRR RRRR HHHH HHHH HHHH HHHH HHHH HHHH\*** |
| * 7 bits pour l'identification réseau ou 27 è128 réseaux |
| * **Réseaux possibles**: **1**.0.0.0 à **126**.0.0.0 donc 126 réseaux **Exclusions**:  les réseaux 0.0.0.0 (utilisé dans les tables d’aiguillage) et 127.0.0.0 (réservé pour l’adresse de bouclage) **Adresses privées**:  le réseau 10.0.0.0 |
| * 24 bits disponibles pour l'hôte ou 224 è16 777 216 - 2 (numéro de réseau et adresse de diffusion générale) |

|  |
| --- |
| **Classe B** **10RR RRRR RRRR RRRR HHHH HHHH HHHH HHHH\*** |
| * 14 bits pour identifier les réseaux ou 214 è16 384 réseaux |
| * **Réseaux possibles**: 128.0.0.0 à 191.255.0.0 **Adresses privées**:  les réseaux 172.16.0.0 à 172.31.0.0 |
| * 16 bits disponibles pour l'hôte ou 216 è 65 536 - 2 (numéro de réseau et adresse de diffusion générale) |

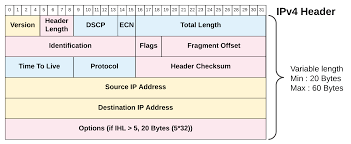
|  |
| --- |
| **Classe C** **110R RRRR RRRR RRRR RRRR RRRR HHHH HHHH\*** |
| * 21 bits pour identifier les réseaux ou 221 è 2 097 152 réseaux possibles |
| * **Réseaux possibles**: 192.0.0.0 à 223.255.255.0 **Adresses privées**:  les réseaux 192.168.0.0 à 192.168.255.0 |
| * 8 bits disponibles pour l'hôte ou 28 è 256 - 2 (numéro de réseau et adresse de diffusion générale) |
| **Classe D 1110 RRRR RRRR RRRR RRRR RRRR HHHH HHHH\*** |
| * **Adresses possibles:** 224.0.0.0 à 239.255.255.255 |

|  |
| --- |
| **Classe E**  **1111 0RRR RRRR RRRR RRRR RRRR HHHH HHHH\*** |
| * Adresses possibles: 240.0.0.0 et plus |

**\*** **R** = bits pour l’identification du réseau et **H** = bits pour l’identification de l’hôte.

**En-tête IPv4**

Le datagramme correspond au format de paquet défini par le protocole Internet. Les cinq ou six (sixième facultatif) premiermot de 32 bits représentent les informations de contrôle appelées en-tête.



* La longueur théorique maximale d'un datagramme IP est de 65535 octets.
* En pratique la taille maximale du datagramme est limitée par la longueur maximale des trames transportées sur le réseau physique.
* La fragmentation du datagramme (définie dans le 2ème mot de 32 bits) devient alors nécessaire dès que sa taille ne lui permet plus d'être directement transporté dans une seule trame physique.
* Les modules internet des équipements prennent en charge le découpage et le réassemblage des datagrammes.

Le protocole Internet transmet le datagramme en utilisant l'adresse de destination contenue dans le cinquième mot de l'en-tête. L'adresse de destination est une adresse IP standard de 32 bits permettant d'identifier le réseau de destination et la machine hôte connectée à ce réseau.

Dans un réseau TCP/IP, on assigne généralement une adresse IP à chaque hôte. Le terme d'hôte est pris dans son sens large, c'est à dire un noeud de réseau. Une imprimante, un routeur, un serveur, un poste de travail sont des nœuds qui peuvent avoir également un nom d'hôte, s'ils ont une adresse IP.

Voici en quoi consistent les différents champs d’une entete IP

* **Version -- 4 bits**   
  Version d'IP utilisée… IPv4 ou IPv6.
* **Longueur de l'en-tête ou IHL (*Internet Header Length*) -- 4 bits**Nombre de mots de 32 bits, soit 4 octets (ou nombre de lignes du schéma).   
  La valeur est comprise entre 5 et 15, car il y a 20 octets minimum et on ne peut dépasser 40 octets d'option (soit en tout, 60 octets).
* **Type de service ou ToS (pour Type of Service) -- 8 bits**  
  Ce champ permet de distinguer différentes qualités de service différenciant la manière dont les paquets sont traités. Composé de 3 bits de priorité (donc 8 niveaux) et trois indicateurs permettant de différencier le débit, le délai ou la fiabilité.
* **Longueur totale en octets ou Total Length -- 16 bits**Nombre total d'octets du datagramme, en-tête IP comprise.   
  Donc, la valeur maximale est (216)-1 octets.
* **Identification -- 16 bits**  
  Numéro permettant d'identifier les fragments d'un même paquet.
* **Indicateurs ou drapeaux (*flags*) -- 3 bits**
  + Premier bit  
    actuellement inutilisé.
  + Deuxième bit DF (*Don't Fragment*)  
    Lorsque ce bit est positionné à 1, il indique que le paquet ne peut pas être fragmenté.   
    Si le routeur ne peut acheminer ce paquet (taille du paquet supérieure à la MTU), il est alors rejeté.
  + Troisième bit MF (*More Fragments*)  
    Quand ce bit est positionné à 1, on sait que ce paquet est un fragment de données et que d'autres doivent suivre.   
    Quand il est à 0, soit le fragment est le dernier, soit le paquet n'a pas été fragmenté.
* **Décalage du segment (*Fragment offset*) -- 13 bits**Position du fragment par rapport au paquet de départ, en nombre de mots de 8 octets.
* **Durée de vie ou TTL (*Time To Live*) -- 8 bits**  
  Initialisé par l'émetteur, ce champ est décrémenté d'une unité généralement à chaque saut de routeur. Quand TTL = 0, le paquet est abandonné et un message ICMP est envoyé à l'émetteur pour information.
* **Protocole (*Protocol*) -- 8 bits**  
  Numéro du protocole au-dessus de la couche réseau : TCP = 6, UDP = 17, ICMP = 1.  
  Ce champ permet d'identifier le protocole utilisé par le niveau supérieur …
  + **Internet Control Message Protocol** ou ICMP est repéré par les bits 00000001,   
    qu'on écrit souvent en hexadécimal avec 01 ;
  + **Transmission Control Protocol** ou TCP par les bits 00000110, soit 06 ;
  + **User Datagram Protocol** ou UDP par les bits 00010001, soit 17 en décimal.
* **Somme de contrôle de l'en-tête (*Header* *Checksum*) -- 16 bits**Complément à un de la somme complémentée à un de tout le contenu de l'en-tête afin de détecter les erreurs de transfert. Si la somme de contrôle est invalide, le paquet est abandonné sans message d'erreur.
* **Adresse source (*Source address*) -- 32 bits**Adresse IP de l'émetteur sur 32 bits.
* **Adresse de destination (*destination address*) -- 32 bits**Adresse IP du récepteur sur 32 bits.
* **Options -- 0 à 40 octets par mots de 4 octets**Options facultatives ...
* **Remplissage (*padding*)**  
  Champ de taille variable comprise entre 0 et 7 bits.   
  Il permet de combler le champ option afin d'obtenir un en-tête IP multiple de 32 bits. La valeur des bits de bourrage est 0.

### **Fragmentation et assemblage IP**

**Fragmentation**Lorsqu’un un datagramme est fragmenté, il n'est réassemblé que par la couche IP destinatrice finale.   
Cela implique trois remarques …

* La **taille des datagrammes reçus** par le destinataire final est directement **dépendante du plus petit MTU rencontré** sur le réseau
* Les **fragments deviennent des datagrammes** à part entière.
* Rien ne s'oppose à ce qu'un **fragment soit à nouveau fragmenté**.

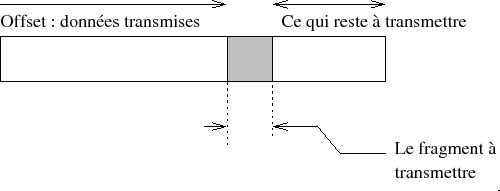
Cette opération est absolument transparente pour les couches de transport qui utilisent IP.

Quand un datagramme est fragmenté, chaque fragment comporte la même valeur de champ identification que le datagramme initial.  
S'il y a encore des fragments, un des bits du champ Drapeaux (*flags*) est positionné à 1 pour indiquer More fragment ».  
Ce champ a une longueur de 3 bits.

Le champ Décalage des données (*fragment* *offset*) contient le décalage du fragment, relativement au datagramme initial.  
Cet offset est codé sur 13 bits.

**Fonctionnement de la fragmentation**

* Quand un datagramme est fragmenté, il n'est réassemblé que par la couche IP destinatrice finale. Cela implique trois remarques :
  1. La taille des datagrammes reçus par le destinataire final est directement dépendante du plus petit MTU rencontré.
  2. Les fragments deviennent des datagrammes à part entière.
  3. Rien ne s'oppose à ce qu'un fragment soit à nouveau fragmenté.
* Cette opération est absolument transparente pour les couches de transport qui utilisent IP.
* S'il y a encore des fragments, un des bits du champ **FLAGS** est positionné à 1 pour indiquer **More fragment**. Ce champ a une longueur de 3 bits.
* Le champ **FRAGMENT OFFSET** contient l'offset du fragment, relativement au datagramme initial.   
  Cet offset est codé sur 13 bits.
* Quand un datagramme est fragmenté, chaque fragment comporte la même valeur de champ **IDENTIFICATION** que le datagramme initial.



Pour tous les fragments …

* Les données doivent faire un multiple de 8 octets, sauf pour le dernier fragment, évidement.
* Le champ Longueur totale change.
* Chaque fragment est un datagramme indépendant, susceptible d'être à son tour fragmenté.

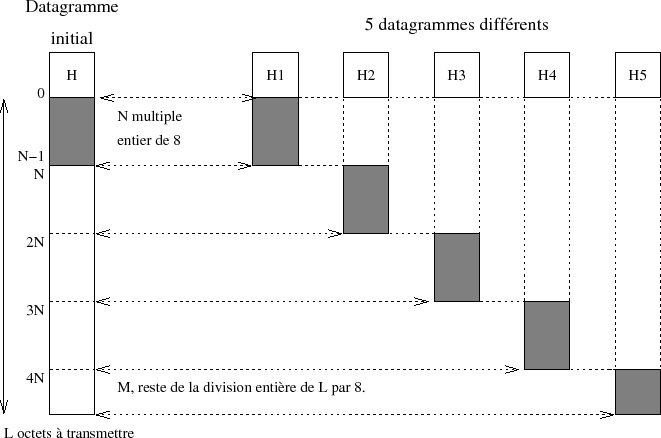
Pour le dernier fragment …

* Le drapeau est remis à zéro.
* Les données ont une taille quelconque.

**Réassemblage**

Tous les datagrammes issus d'une fragmentation deviennent des datagrammes IP comme (presque) les autres.  
Ils arrivent à destination, possiblement dans le désordre, dupliqués. IP doit faire le tri.  
il y a suffisamment d'information dans l'en-tête pour réassembler les fragments épars  
Mais si un fragment manque, la totalité du datagramme est perdu car aucun mécanisme de contrôle n'est implémenté pour cela dans IP. C'est la raison principale pour laquelle il faut absolument éviter de fragmenter un datagramme IP.

La figure suivante résume l'opération de fragmentation d'un datagramme IP.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | H1 | H2 | H3 | H4 | H5 |
| Identification | I | I | I | I | I |
| flag | MF | MF | MF | MF | 0 |
| offset | 0 | N | 2 x *N* | 3 x *N* | 4 x *N* |
| total length | H+N | H+N | H+N | H+N | H+M |
| header checksum | *C*1 | *C*2 | *C*3 | *C*4 | *C*5 |

On note les variations de certains champs de l'en-tête …

* L’Identification est le même pour tous.
* Le drapeau est 0 pour le dernier datagramme.
* Le décalage croît de la taille du fragment, ici N.
* La longueur totale est généralement différente pour le dernier fragment, sauf cas particulier.
* La somme de contrôle de l’entête est modifiée à chaque fois car le décalalge change (rappel : il ne tient pas compte des données).

## Masque de sous-réseau (IPv4)

Le masque de sous réseau est un élément essentiel à toute configuration IP. Il complète et caractérise l’adresse IP.

### Fonctions

Les deux fonctions du masque de sous-réseau sont les suivantes ...

**Première fonction**  
Déterminer si un hôte distant est un hôtes attaché à son réseau local ou encore est un hôte extérieur, connecté sur Internet.  
C’est donc un élément essentiel.  
Pour déterminer si l’hôte à rejoindre est interne ou externe à son réseau, il y a comparaison binaire (AND) entre son adresse et son masque de sous-réseau et l’adresse distante et son propre masque de sous-réseau.

* Le résultat de cette comparaison peut-être de deux ordres:  
  **Identique**  
  Cela signifie que l’hôte à rejoindre est sur son propre réseau, et peut alors directement communiquer avec ce dernier.  
  **Différent**  
  Il doit alors acheminer le ou les paquets à sa passerelle par défaut, qui alors, à l’aide de sa table d’aiguillage, fera suivre les données jusqu’au destinataire final.

**Seconde fonction**  
Permettre à l’administrateur local de séparer un réseau en sous-entités. Cette option permet de gérer plus facilement des grands réseaux.  
Il permet également aux distributeurs d’adresses Internet de récupérer (à l’aide de CIDR) des réseaux entiers déjà attribué afin de les attribuer de nouveau à d’autre réseaux.

**Avantages**  
Les avantages d’utiliser des sous-réseaux sont les suivants ...

* Utilisation optimisée de l’espace d’adressage.
* Administration simplifiée:  
  Permet d’administrer les sous-réseaux de manière indépendante et efficace.   
  Permet de gérer plus efficacement des parties du réseau qui ont des exigences contradictoires.
* Possibilité de structurer un réseau interne sans affecter les réseaux externes.  
  Seul l’aiguilleur du réseau destination doit connaître la présence des sous-réseaux.
* Sécurité améliorée.  
  La structure du sous-réseau est invisible à l’externe donc la sécurité est améliorée indirectement.

Exemples de création de sous réseau

2 fonctions déterminant …

* La séparation d’un réseau en plus petites parties.   
  Il est alors possible de déterminer le nombre de sous-réseaux souhaités et le nombre de nœuds (hôtes).
* Le masque de sous-réseau permet de déterminer si l’adresse de destination appartient au réseau local (réseau de l’émetteur) ou distant.

**Méthode 1  
Détermination du nombre de bits**

1. Écrire en binaire le nombre de sous-réseaux souhaités
2. Compter le nombre de bits nécessaire  
   (ne pas oublier de soustraire 2)
3. Définition du masque de sous-réseau

Pour établir le masque de sous-réseau, il faut emprunter des bits à la portion identificateur d’hôte.  
Additionner au masque de sous-réseau par défaut le nombre de bits nécessaire.

Ex. Classe B

1111 1111 1111 1111 0000 0000 0000 0000

0000 0000 0000 0000 **1111** 0000 0000 0000

1111 1111 1111 1111 1111 0000 0000 0000

Si l’on désire obtenir 12 sous-réseaux :

1. Il faut 4 bits ou 24
2. Un masque de sous-réseau ne doit pas contenir que des 0 ou des 1 -- donc 16 – 2 = 14
3. Il reste 12 bits pour les hôtes -- 212 ou 4 094 – 2 hôtes

**Remarque** …  
Il n’est pas possible d’utiliser qu’un seul bit pour définir un masque de sous-réseau   
(**Quoiqu’une récente RFC le permet**)

Valeurs possibles pour un masque de sous réseau :  
**128** – 192 – 224 – 240 – 248 – 252 – 254 – 255

ID des masques de sous-réseau

Ex. Classe B (160.16.0.0)  
 4 bits utilisés pour le masque  
 16 sous-réseaux – 2 = 14

|  |  |  |
| --- | --- | --- |
| Octet binaire | Valeur décimale | ID du réseau |
| **0000** 0000 | 0 | Invalide |
| **0001** 0000 | 16 | 160.16.16.0 |
| **0010** 0000 | 32 | 160.16.32.0 |
| **0011** 0000 | 48 | 160.16.48.0 |
| **0100** 0000 | 64 | 160.16.64.0 |
| **0101** 0000 | 80 | 160.16.80.0 |
| **0110** 0000 | 96 | 160.16.92.0 |
| **0111** 0000 | 112 | 160.16.112.0 |
| **1000** 0000 | 128 | 160.16.128.0 |
| **1001** 0000 | 144 | 160.16.144.0 |
| **1010** 0000 | 160 | 160.16.160.0 |
| **1011** 0000 | 176 | 160.16.176.0 |
| **1100** 0000 | 192 | 160.16.192.0 |
| **1101** 0000 | 208 | 160.16.208.0 |
| **1110** 0000 | 224 | 160.16.224.0 |
| **1111** 0000 | 240 | invalide |

Ex. Classe B  
 3 bits utilisés pour le masque  
 8 sous-réseaux – 2 = 6

|  |  |  |
| --- | --- | --- |
| Octet binaire | Valeur décimale | ID du réseau |
| **000**0 0000 | 0 | Invalide |
| **001**0 0000 | 32 | 160.16.32.0 |
| **010**0 0000 | 64 | 160.16.64.0 |
| **011**0 0000 | 96 | 160.16.96.0 |
| **100**0 0000 | 128 | 160.16.128.0 |
| **101**0 0000 | 160 | 160.16.160.0 |
| **110**0 0000 | 192 | 160.16.192.0 |
| **111**0 0000 | 224 | Invalide |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tableau synthèse Masque de sous-réseaux, ID de sous-réseaux et nombre de sous-réseaux pour une classe B | | | | | | | |
| Poids binaire | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| Bits supplémentaires | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| SR possibles | (22 – 2)  2 | (23 – 2)  6 | (24 – 2)  14 | (25 – 2)  30 | (26 – 2)  62 | (27 – 2)  126 | (28 – 2)  254 |
| Masque de SR | 128 + 64 192 | 192 + 32 224 | 224 + 16 240 | 240 + 8 248 | 248 + 4 252 | 252 + 2 254 | 254 + 1 255 |
| Bits ID hôtes | 6 + 8 14 | 5 + 8 13 | 4 + 8 12 | 3 + 8 11 | 2 + 8 10 | 1 + 8 9 | 0 + 8 8 |

**Méthode 2**  
**ID des hôtes**

Les adresses pour chaque sous-réseau correspondent à toutes les combinaisons binaires possibles comprises entre l’adresse du sous-réseau et son adresse de diffusion.

Ex. Classe B  
 160.16.32.0 (numéro de réseau)  
 255.255.240.0 (masque de SR)  
 **160.16.32.1** à **160.16.47.254**

Étapes …

1. Trouver l’ID du sous-réseau suivant :  
 ex. 255.255.240.0 (4 bits sont utilisés donc 16 comme valeur incrémentée)  
 160.16.32.0 + 16 = 160.16.48.0

2. Déterminer l’adresse du premier hôte :  
 160.16.32.1

3. Déterminer l’adresse de diffusion :  
 160.16.47.255

3. Déterminer l’adresse du dernier hôte :  
 160.16.47.254

**Étude de cas**

On dispose d’une classe B.  
On doit établir un schéma d’implantation pour 28 sous-réseaux avec un maximum de 300 hôtes chacun.

**Solution 1**Prendre 5 bits pour le sous-réseau   
donc 30 sous réseaux possibles (25 – 2)  
Reste 11 bits pour l’hôte  
donc 2 046 hôtes (211 – 2)

**Solution 2**Prendre 9 bits pour les hôtes   
donc 510 hôtes possibles (29 – 2)  
Reste 7 bits pour le sous-réseau  
donc 126 sous-réseaux (27 – 2)

**Affectation optimisée des numéros de sous-réseaux  
Comptage en miroir**

Afin de permettre une souplesse pour la modification des sous-réseaux (dimensionnement) une technique d’attribution des adresses pour les hôtes est suggérée.   
La frontière du masque de sous-réseau délimite la partie appartenant au réseau et celle de l’hôte. À gauche de cette frontière le masque est composé uniquement de 1 et à droite de 0.   
Déplacer la frontière (vers la gauche ou vers la droite) affectera le masque de sous-réseau et par le fait même la numérotation des sous-réseaux et des hôtes.

Afin de minimiser le processus de renumérotation lors d’une telle modification il est suggéré ...

* d’affecter les numéros de sous-réseaux contenant des 1 le plus à gauche;
* d’affecter les numéros d’hôtes contenant des 0 le plus à droite.



|  |  |
| --- | --- |
| Numéro de sous-réseau | Numéro d’hôte |
| 0 (réservé – ce sous-réseau) | 0 (réservé – cet hôte) |
| 10 | 10 |
| 01 | 01 |
| 11 | 11 |
| 100 | 100 |
| 101 | 101 |
| : | : |
| : | : |
| 11…1110 | 11…1110 |
| 11…1111 (réservé – tous les SR) | 11…1111 (réservé – tous les hôtes) |

Cette technique est présentée dans la RFC 1219.

### Internet Protocol version 6 (IPv6)

Au début des années 90, l’évolution du réseau Internet semblait compromise à court terme car le protocole IP (Internet Protocol) limitait le nombre d’équipements qui pouvaient s’y connecter.

À sa création, ce réseau ne devait servir qu’à relier une centaine de machines mais de nombreuses catégories d’utilisateur sont très vite venues s’y joindre. On y trouvait les scientifiques, les universitaires, puis en 1992, le réseau fut ouvert aux activités commerciales ainsi qu’aux particuliers. Depuis, le nombre d’équipements connectés ne cesse d’augmenter et on approche de la saturation du réseau.

Les chercheurs ont repoussé cette saturation afin de développer une nouvelle version du protocole IP qui sera nommé IPv6. IPv6 prend en compte les avancées issues des recherches sur les réseaux menées depuis 25 ans.

Elle porte notamment sur …

* l’**auto configuration** ;
* la **mobilité** ;
* la **diffusion multipoints**  
  et
* la **sécurité**.

IPv6 est le protocole de la prochaine génération d'Internet et c'est pour cela qu'on l'appelle aussi IPng, initiales de ***Internet Protocol Next Generation***.  
On s'attend à ce que IPv6 remplace graduellement IPv4, avec une période de transition de plusieurs années de coexistence.

**IPv6 et non IPv5**

L'information qui circule par un réseau IP comme Internet est divisée en paquets. Chaque paquet comprend non seulement les données à transmettre mais aussi un emballage qui, entre autres, possède le numéro de version du protocole IP qui est en train de s'utiliser.

IANA a décidé d'assigner le numéro de version 5 pour un protocole expérimental qui, dans la pratique, ne s'est jamais utilisé, le ST-II ***Stream Protocol version 2***.

Au départ, l'idée était d'identifier les paquets de ST en regardant le numéro de version du protocole IP …

* si c'était le numéro 4, il s'agissait alors d'un paquet normal,
* si c'était le 5, il s'agissait d'un paquet *Stream Protocol*.

C'est pour cette raison que le numéro 5 ne peut s'utiliser pour désigner la version du protocole IP qui suit la version 4. Il ne s'agit donc pas de brûler une étape en ne désignant pas la version 5 mais simplement, la version 6 est celle qui suit la 4 puisque le numéro 5 s'est réservé pour un autre protocole.

**Avantages d’IPv6**

* **Échelonnement**  
  IPv6 possède des adresses de 128 bits face aux adresses de 32 bits de Ipv4. Par conséquent, le nombre d'adresses IP disponibles est multiplié par   
  7,9 \* 1028
* **Sécurité**  
  IPv6 assure la sécurité dans ses spécifications en utilisant le cryptage de l'information et l'authentification de l'expéditeur de ladite information.
* **Applications en temps réel**Afin d'offrir un meilleur support pour le trafic en temps réel (par exemple, vidéoconférence), IPv6 comprend l'étiquetage de flux dans ses spécifications. Avec ce système, les aiguilleurs peuvent reconnaître à quel flux d'un extrême à l'autre appartiennent les paquets qui se transmettent.
* **Plug and Play**  
  IPv6 comprend dans sa forme standard un mécanisme plug and play, ce qui facilite aux utilisateurs la connexion de leurs équipements au réseau.   
  La configuration se réalisera automatiquement.
* **Spécifications plus claires et optimisées**IPv6 suivra les bonnes pratiques de IPv4 et éliminera les caractéristiques non utilisées ou obsolètes de IPv4, ce qui permettra d'optimiser le protocole d'Internet.   
  L'idée est de conserver l'aspect positif et d'éliminer les inconvénients du protocole actuel.

Les travaux de recherche sont à ce jour terminés et de nombreuses implantations sont disponibles aussi bien pour les équipements d’interconnections que les ordinateurs.

**Nombre d'adresses IPv6 disponibles**

Le nombre d'adresses IP disponibles en IPv6 est égal au plus grand numéro qui puisse être représenté avec 128 bits, c'est-à-dire …  
2128 = 3.4 \* 1038

Le numéro qui représente cette quantité est si incroyablement grand qu'il est difficile de l'imaginer. Nous pourrions le comparer au nombre d'étoiles de le Voie Lactée, qui s'élève à cent mille millions.   
Il faudrait 100 mille billions de galaxies pareilles à notre Voie Lactée pour réunir autant d'étoiles que d'adresses IP il y ura dans le nouvel univers d'Internet.   
Bien que l'on sache qu'il existe des milliers de millions de galaxies, il n'est pas imprudent d'affirmer que le nouvel Internet aura plus d'adresses IP disponibles que d'étoiles dans l'univers.

Nous pourrions également comparer le nombre d'adresses IPv6 avec la quantité d'atomes qu'il y a dans la matière. Il faudrait 67 000 tonnes de fer pour réunir autant d'atomes dudit métal qu'il existe d'adresses IPv6. Si au lieu de fer nous prenons l'exemple de l'atmosphère, il faudrait un cube d'au moins 23 Km de côté pour que l'air compris à l'intérieur ait autant de molécules qu'il y a d'adresses IP dans l'Internet de la nouvelle génération.

**Assignation des adresses Ipv6**

Les fournisseurs et ISP qui contribuent au processus d'implantation de la nouvelle version du protocole IP suivent les instructions du RIPE relatives à la distribution de l'énorme espace d'adressage IP version 6 entre les clients.

Il existe une grande différence entre les recommandations quant à l'assignation des adresses IP version 4 qui cherche avant tout à économiser les adresses puisqu'il reste peu de temps avant qu'elles ne s'épuisent, et la version 6 qui cherche surtout à être flexible.

Le RIPE recommande aux ISP et aux opérateurs d'assigner à chaque client de IPv6 un sous réseau du type /48 pour que le client puisse gérer ses propres sous réseaux sans avoir à utiliser NAT (avec IPv6, le NAT disparaît).

Avec IPv6, le client recevra une sous classe comme la suivante …  
**2001:0ba0:1c01::/48**

Ce même client pourra également concevoir dans ses installations 65 535 sous réseaux différents, résultant des combinaisons créées en utilisant w,x,y,z dans le groupe ...  
**2001:0ba0:01b0:wxyz::/64**

Chacun de ces 65 535 sous réseaux que notre client pourra créer, aura à son tour plus de 18 trillions d'adresses IP distinctes qui peuvent être d'assignation automatique (plug and play) ou manuelle de la part du client.   
À l’heure actuelle, IPv6 n’est pas utilisé dans la majorité des réseaux informatiques. Les entreprises boudent IPv6 car le déploiement de ce protocole engendre des frais de mise en place non justifiés à leurs yeux.

**Adressage IPv6**

La taille d'une adresse IPv6 est de 16 octets (128 bits). Cette taille, suffisamment importante, permet d'établir un plan d'adressage hiérarchisé en trois niveaux.

* **Topologie publique utilisant 48 bits** ;
* **Topologie de site sur 16 bits** ;
* **Topologie d'interface sur 64 bits**.

Une adresse IPv6 est donc longue de 128 bits, soit 16 octets, contre 32 bits pour IPv4.   
La notation décimale pointée employée pour les adresses IPv4 (par exemple 172.31.128.1) est abandonnée au profit d'une **écriture hexadécimale**, où les **8 groupes de 2 octets (16 bits par groupe)** sont séparés par un signe deux-points …  
**2001:0db8:0000:85a3:0000:0000:ac1f:8001  
2A01:E35:2421:4BE0:CDBC:C04E:A7AB:ECF3**

Il est permis d'**omettre de 1 à 3 chiffres zéros non significatifs dans chaque groupe** de 4 chiffres hexadécimaux. Ainsi, l'adresse IPv6 ci-dessus est équivalente à:  
2001:db8:0:85a3:0:0:ac1f:8001

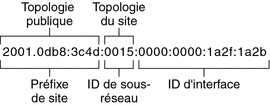
De plus, une unique suite de un ou plusieurs groupes consécutifs de 16 bits tous nuls peut être omise, en conservant toutefois les signes deux-points de chaque côté de la suite de chiffres omise, c'est-à-dire une paire de deux-points (::)19. Ainsi, l'adresse IPv6 ci-dessus peut être abrégée en:  
2001:db8:0:85a3::ac1f:8001

|  |  |
| --- | --- |
| Préfixe | Description |
| ::/8 | Adresses réservées |
| 2000::/3 | Adresses unicast routables sur Internet |
| fc00::/7 | Adresses locales uniques |
| fe80::/10 | Adresses locales lien (auto-configuration) |
| ff00::/8 | Adresses multicast |
| ::1/128 | **A**dresse de bouclage (semblable à 127.0.0.1 en IPv4) |
| 0:0:0:0:0:0:0:0 ou encore notée :: | Adresse utilisée pendant l'initialisation de l'adresse IPv6 d'une machine. Cela est une phase transitoire |

**Attribution des adresses IP**

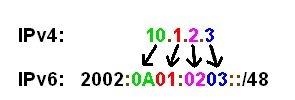
L'utilisation de préfixes séparés pour les adresses affectées à un fournisseur et les adresses affectées à une zone géographique constitue un compromis entre deux différentes visions du futur réseau Internet. Chacun de ces fournisseurs dispose d'une fraction réservée de l'espace d'adressage (adresses unicast = 1/8 de cet espace). Les 5 premiers bits qui suivent le préfixe 0010 (2000::/3) sont utilisés pour indiquer dans quel " registre " se trouve le fournisseur d'accès. Actuellement, trois registres sont opérationnels, pour l'Amérique du nord, l'Europe et l'Asie. Jusqu'à 29 nouveaux registres pourront être ajoutés ultérieurement.

Chaque registre est libre de diviser les 15 octets restants comme il l'entend. Une autre possibilité est d'utiliser un octet pour indiquer la nationalité du fournisseur et de laisser toute liberté aux octets suivant pour définir une structure d'adresses spécifique.

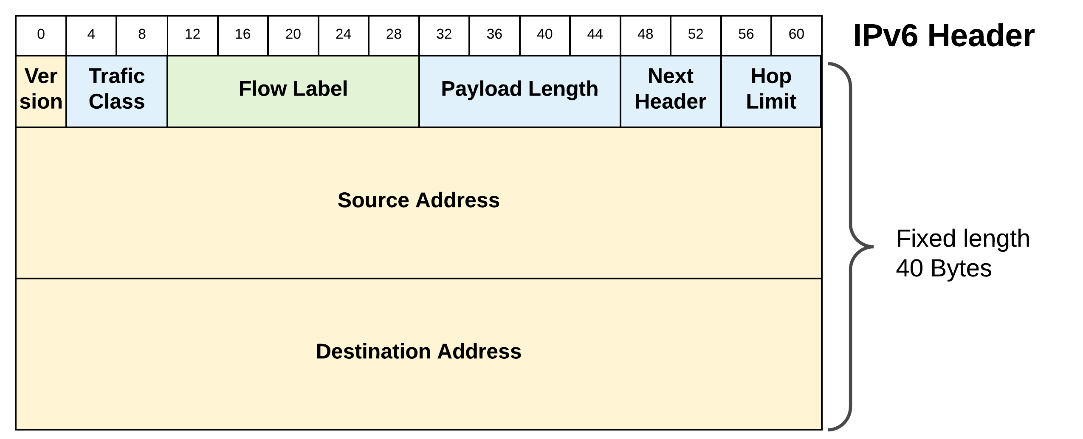


Insertion d’une Adresse IPv4 dans une adresse IPv6

Il est possible d’encapsuler une adresse IPv4 dans un réseau IPv6.  
Elle prend la forme suivante …



**Entête IPv6**



L'en-tête du paquet IPv6 est de **taille fixe à 40 octets**, tandis qu'en IPv4 la taille minimale est de 20 octets, des **options pouvant la porter jusqu'à 60 octets**, ces **options demeurant rares** en pratique.

La signification des champs est la suivante :

* **Version *(Version)*** -- 4 bits  
  Ce champ représente le numéro de version du Protocole Internet et est fixé à la valeur du numéro de protocole internet (04 pour IPv4 et 06 pour IPv6) ;
* **Classe de trafic (*Traffic Class*)** -- 8 bits  
  Ce champ est utilisé pour la qualité de service elle définit la priorité du datagramme afin que des nœuds origines et des routeurs transmetteurs puissent identifier et distinguer la classe ou la priorité du paquets IPv6 en question ;
* **Étiquette de flux (*Flow Label*)** -- 20 bits  
  Ce champ est utilisé par une source pour nommer des séquences de paquets pour lesquels un traitement spécial de la part des routeurs IPv6 est demandé et permet le marquage d'un flux pour un traitement différencié dans le réseau ;
* **Longueur de la charge utilse (*Payload length*)** -- 16 bits  
  Ce champ indique la taille de la charge utile en octets ;
* **Prochain entête (Next Header)** -- 8 bits  
  Ce champ identifie le type de d’entête qui suit immédiatement selon la même convention qu'IPv4 ;
* **Nombre de sauts (Hop Limit)** -- 8 bits  
  Ce champ est décrémenté de 1 par chaque aiguilleur (*router*), le paquet est détruit si ce champ atteint 0 en transit ;
* **Adresse source (*Source Address*)** -- 128 bits  
  Champ indiquant l’adresse source ;
* **Adresse de destination (*Destination Address*)** -- 128 bits  
  Champ indiquant l’adresse de destination.

Il est possible qu'un ou plusieurs en-têtes d'extension suivent l'en-tête IPv6.   
L'en-tête de routage permet par exemple à la source de spécifier un chemin déterminé à suivre.

**Comparaison avec IPv4**

* La **taille de l'en-tête est fixe**, le champ IHL (*IP Header Length*) est donc inutile.
* Le **champ *Time to Live*** (TTL) est **renommé en *Hop Limit***, reflétant la pratique (la RFC 791 prévoyait en effet que le champ TTL reflétait le temps en secondes).
* Il n'y a **pas de somme de contrôle** **sur l'en-tête**.   
  Avec IPv4, cette somme de contrôle inclut le champ TTL et oblige les routeurs à le recalculer dans la mesure où le TTL est décrémenté.   
  Ceci simplifie le traitement des paquets par les routeurs.
* Le **champ *Payload length* n'inclut pas la taille de l'en-tête** standard (ni des en-têtes optionnels qui suivent), contrairement au champ *Total length* d'IPv4.
* Les éventuelles informations relatives à la fragmentation sont repoussées dans un en-tête qui suit.

### CIDR (Classless Inter-Domain Routing)

Le routage inter-domaine sans classe est une méthode d'allocation d'adresses IP et de routage IP. L'Internet Engineering Task Force (IEEE) a introduit le CIDR en 1993 afin de remplacer l'ancienne architecture d'adressage réseau de classe sur Internet. Son objectif était de ralentir la croissance des tables de routage sur les routeurs sur Internet et d'aider à ralentir l'épuisement rapide des adresses IPv4.

Les adresses IP sont décrites comme consistant en deux groupes de bits dans l'adresse: les bits les plus significatifs sont le préfixe de réseau, qui identifie un réseau ou un sous-réseau entier, et l'ensemble le moins significatif constitue l'identifiant d'hôte, qui spécifie une interface particulière d'un hôte sur ce réseau. Cette division est utilisée comme base du routage du trafic entre les réseaux IP et pour les politiques d'allocation d'adresses.

Alors que la conception de réseau classe pour IPv4 a dimensionné le préfixe réseau comme un ou plusieurs groupes 8 bits, ce qui entraîne les blocs d'adresses de classe A, B ou C, sous l'espace d'adressage CIDR est alloué aux fournisseurs de services Internet et aux utilisateurs finaux sur n'importe quelle adresse - limite de bits. Dans IPv6, cependant, l'identificateur d'interface a une taille fixe de 64 bits par convention, et les sous-réseaux plus petits ne sont jamais alloués aux utilisateurs finaux.

Le CIDR englobe plusieurs concepts. Il est basé sur le masquage de sous-réseau de longueur variable (Variable Length Subnet Mask) qui permet de spécifier des préfixes de longueur arbitraire. Le CIDR a introduit une nouvelle méthode de représentation des adresses IP, désormais connue sous le nom de notation CIDR, dans laquelle une adresse ou un préfixe de routage est écrit avec un suffixe indiquant le nombre de bits du préfixe, comme 192.0.2.0/24 pour IPv4, et 2001:db8::/32 pour IPv6.   
Le CIDR a introduit un processus administratif d'allocation de blocs d'adresses aux organisations en fonction de leurs besoins réels et projetés à court terme. L'agrégation de plusieurs préfixes contigus a abouti à des super-réseaux dans le plus grand Internet, qui, dans la mesure du possible, sont annoncés comme agrégats, réduisant ainsi le nombre d'entrées dans la table de routage globale.

Le CIDR a été mis au point afin (principalement) de diminuer la taille de la table de routage contenue dans les routeurs, Il permet d’agréger plusieurs entrées de cette table en une seule. Cette technique a permis d'agréger des réseaux par région géographique et fournisseurs d'accès. Elle permet ainsi une agrégation maximum des sous-réseaux qui sont routés ensembles avec la même politique

L’adresse IP est suivie d'une barre oblique (*slash /*) indiquant le nombre de bits à 1 dans la notation binaire du masque de sous-réseau.

**Notation CIDR**

La notation CIDR est une représentation compacte d'une adresse IP et de son préfixe de routage associé. La notation est construite à partir d'une adresse IP, d'une barre oblique (/) et d'un nombre décimal. Le nombre final est le nombre de bits 1 dans le masque de routage, traditionnellement appelé masque de sous-réseau.   
L'adresse IP dans la notation est toujours représentée selon les normes IPv4 ou IPv6.

L'adresse peut désigner une adresse d'interface unique et distincte, ou elle peut être l'adresse de début d'un réseau entier. Lors de l'expression d'un réseau, sa taille est donnée par le nombre d'adresses possibles avec le nombre de bits restants les moins significatifs en dessous du préfixe, c'est-à-dire l'identifiant d'hôte.

Par exemple …

* 192.168.100.14/24 représente l'adresse IPv4 192.168.100.14 et son préfixe de routage associé 192.168.100.0, ou de manière équivalente, son masque de sous-réseau 255.255.255.0, qui a 24 bits de début 1 ;
* le bloc IPv4 192.168.100.0/22 représente les 1024 adresses IPv4 de 192.168.100.0 à 192.168.103.255 ;
* le bloc IPv6 2001:db8::/48 représente le bloc d'adresses IPv6 de 2001: db8:0:0:0:0 0:0 à 2001:db8:0:ffff:ffff:ffff:ffff:ffff ;
* :1/128 représente l'adresse de bouclage IPv6. S  
  a longueur de préfixe est 128, qui est le nombre de bits dans l'adresse.

Pour IPv4, la notation CIDR est une alternative à l'ancien système de représentation des réseaux par leur adresse de départ et le masque de sous-réseau, tous deux écrits en notation décimale par points.   
192.168.100.0/24 est équivalent à 192.168.100.0/255.255.255.0.

Le nombre d'adresses d'un sous-réseau peut être calculé comme **2** **longueurs d'adresse** - longueur de préfixe, où la longueur de l'adresse est de 128 pour IPv6 et de 32 pour IPv4.

**Masques de sous-réseau**

Un masque de sous-réseau est un masque de bits qui code la longueur de préfixe associée à une adresse IPv4 ou à un réseau en notation à quatre points: 32 bits, commençant par un nombre de 1 bits égal à la longueur de préfixe, se terminant par 0 bit et codé en quatre points (.) -- format décimal en pointillé : 255.255.255.0. Un masque de sous-réseau code les mêmes informations qu'une longueur de préfixe mais est antérieur à l'avènement du CIDR.   
En notation CIDR, les bits de préfixe sont toujours contigus. La RFC 950 [5] autorisait les masques de sous-réseau à spécifier des bits non contigus jusqu'à la RFC 4632 [4]: la section 5.1 stipulait que le masque devait rester contigu. Compte tenu de cette contrainte, un masque de sous-réseau et une notation CIDR remplissent exactement la même fonction.

## Protocole ARP

### Rôle d'ARP

Une station ne lit le contenu d'une trame qu'à la condition qu'elle lui soit adressée.   
Or le paquet IP est justement placé dans la trame. Lorsqu'une couche haute transmet des données à une autre hôte, elle communique à la couche IP, l'adresse IP de destination.

IP formate ensuite un paquet avec une entête IP. Celle-ci comportent l'adresse de destination, transmise par la couche supérieure, et l'adresse IP source de l'émetteur.   
Cette dernière est connue par IP car il a été initialisé lors de l'installation de l’hôte.

IP communique ce paquet à la couche 2 (ici la sous-couche MAC) pour l'encapsuler dans une trame MAC avant émission sur le support.   
Mais il doit également communiquer l'adresse MAC de destination de la trame, pour que la couche 2 puisse formater correctement la trame.

ARP (*Address Resolution Protocol*) prend en charge, comme son nom l'indique, la résolution d'adresse entre le niveau 2 et le niveau 3. Il a pour rôle de trouver l'adresse MAC correspondant à une adresse IP donnée.

Lorsqu'il a découvert cette adresse, il met à jour une table ARP.

ARP a été défini dans la RFC 8261 -- *An Ethernet Address Resolution Protocol*.

Le protocole ARP est nécessaire au fonctionnement d’IPv4 utilisé au-dessus d’un réseau de type Ethernet.   
En IPv6, les fonctions de ARP sont reprises par le *Neighbor Discovery Protocol* (NDP).

### Fonctionnement d'ARP

IP doit donc transmettre à la couche MAC l'adresse de destination MAC de la trame qui véhiculera le paquet IP transmis.

La procédure est la suivante ...

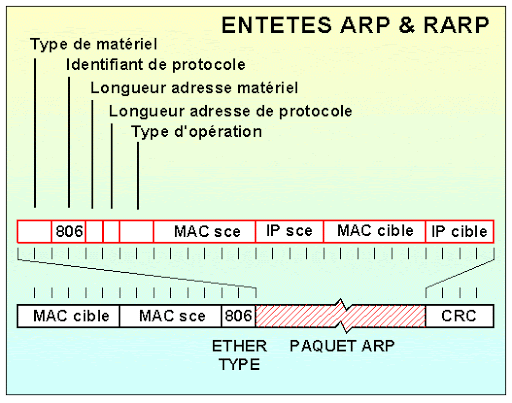
* IP scrute sa table ARP pour trouver la correspondance MAC de l'adresse IP transmise par la couche supérieure :
* S'il n'a jamais communiqué avec la station ayant cette adresse depuis son démarrage (la table ARP est en mémoire vive, donc volatile), il ne connaît pas la correspondance et passe à l'étape suivante ;
* S'il trouve la correspondance, il transmet l'adresse MAC à la couche MAC en même temps que son paquet IP et c'est fini pour lui ;
* S'il ne trouve pas de correspondance, il transmet l'adresse IP destination au protocole ARP, en lui demandant de trouver la correspondance d'adresse MAC ;
* ARP formate un paquet *ARP request* (son format sera détaillé plus tard), qu'il place directement dans une trame MAC de diffusion (*broadcast*).   
  Celle-ci, parce qu'elle est émise en diffusion générale (*broadcast*), est lue par toutes les stations du LAN.   
  Le paquet ARP Request prend la forme ...  
  Avis à tous … Je suis 192.168.1.100 et mon adresse MAC est … 00:00:0C:1A:34:EC.   
  Je cherche 192.168.1.200.  
  Quelle est l’adresse MAC qui correspond à cette adresse IP ?  
  Toutes les stations actives du réseau reçoivent la trame et lisent son contenu.   
  Elles détectent un paquet *ARP Request* qu'elles transmettent à leur propre sous-programme ARP ;
* Seul l’hôte ayant l'adresse IP indiquée dans le paquet *ARP Request* va répondre.  
  Elle émettra un paquet *ARP Reply* annonçant ...  
  Je suis l’hôte qui possède l’adresse IP 192.168.1.200 et mon adresse MAC est 00:00:0C:2A.32:55.  
  Le paquet *ARP Reply* est encapsulé dans une trame émise en monodiffusion (*unicast*) à l'hôte qui a recherché l'adresse (adresse MAC 00:00:0C:1A:34.EC dans cet exemple). Il est, en effet, inutile de faire à nouveau une diffusion générale avec la réponse.  
  L'hôte en profite également pour mettre à jour sa table ARP (192.168.1.200 = 00:00:0C:1A:34:EC) ;
* L'hôte ayant initié la séquence ARP reçoit donc une trame MAC qui lui est adressée et contenant un paquet *ARP Response* véhiculant l'adresse MAC en correspondance avec l'adresse IP sur laquelle elle a lancé la recherche.   
  Elle met à jour sa table ARP (192.168.1.100 = 00:00:0C:2A:32:55) ;

La séquence de recherche est terminée.  
Maintenant, IP peut envoyer son paquet à la couche MAC en transmettant l'adresse MAC.

### Formats des paquets ARP

ARP est encapsulé directement dans IP (il n'est pas placé dans UDP ou TCP). Il propose deux paquets …

* ARP Request  
  La requête pour initier la recherche ;
* ARP Reply  
  La réponse à la requête.



Voici le détail de la trame ARP ...

* **Type de matériel (*Hardware type*)**  
  Ce champ indique le type de réseau support sur lequel la séquence a été activée.   
  Pour Ethernet le code est (0001)h ;
* **Indicateur de protocole (*Hardware type*)**  
  Ce champs inique le type 806 qui caractérise ces trames ARP et RARP ;
* **Longueur Adresse matérielle (*Hardware Address Length*)**  
  Ce champ indique le nombre d'octets des champs d'adresse MAC des paquets ARP.   
  La valeur est généralement (0006)h ;
* **Longueur adresse de protocole (*Protocol Address Length*)**  
  Le champ Longueur d'adresse protocolaire indique la longueur en octets des champs d'adresse de niveau 3.   
  Pour IP la valeur est (0004)h ;
* **Type d’opération** **(*Operation* *type*)**  
  Le champ Type d'opération indique le type de paquet ARP.   
  (0001)h = ARP\_Request et (0002)h = ARP\_Response (appelé aussi ARP\_Reply) ;
* **Adresse MAC Source (*Sender Hardware Address*)**  
   **Adresse réseau de l’émetteur (Sender Prococol Address)**  
  Les champs adresse MAC source et adresse protocolaire source supportent les adresses MAC et protocolaire (IP) de l'émetteur du paquet.   
  Dans le cas d'un paquet ARP\_Request ce sont donc les adresses niveau 2 et 3 de l'initiateur de la séquence ARP. Dans le cas du paquet ARP\_Reply, l’hôte répondant à la requête ARP y place ses propres adresses niveau 2 et 3.
* **Adresse MAC du destinataire (*Target Hardware Address*)**  
  **Adresse réseau du destinataire (*Target Protocol Address*)**  
  Les champs adresse MAC destination et adresse protocolaire destination supportent les adresses MAC et protocolaire (IP) de l’hôte à atteindre.   
  Dans le cas d'un paquet ARP\_Request ce sont donc les adresses niveau 2 et 3 de l'hôte pour lequel l'initiateur de la séquence ARP a lancé la recherche. Comme l'émetteur du paquet ne connait pas l'adresse niveau 2 de l’hôte qu'il cherche à atteindre, le champ adresse MAC destination est placé à 0. Dans un paquet ARP\_Request le champ adresse MAC destination est donc toujours à 0. Par contre le champ adresse protocolaire destination contient toujours l'adresse niveau 3 de l’hôte dont on veut faire la résolution d'adresse.   
  Dans le cas du paquet ARP\_Reply, l’hôte répondant à la requête ARP remplace les valeurs de ces champs par les valeurs d'adresse de l'émetteur du paquet ARP\_Request. Elle recopie en fait les champs adresse MAC source et adresse protocolaire source du paquet ARP\_Request reçu dans les champs destination correspondants du paquet ARP\_Reply émis en réponse.

### Cache ARP

Le cache ARP ou table ARP est une table de couples adresse IPv4 -- adresse MAC contenue dans la mémoire d'un hôte qui utilise le protocole ARP, ce qui est le cas des hôtes qui sont connectés à un réseau IP sur un segment Ethernet.

**Utilisation**  
Cette table est utilisée par les hôtes afin de déterminer l'adresse MAC d'un autre hôte sur le même segment.

Les entrées dans cette table ont une durée de vie limitée, quand une entrée vient à expiration, une nouvelle requête ARP devra être initiée si besoin est.

Certains systèmes d'exploitation permettent de fixer une association dans le cache ARP de façon permanente.

**Requêtes ARP gratuites**

Des requêtes ARP gratuites (*gratuitous ARP*) sont envoyées au démarrage de certains systèmes d'exploitation.   
Par exemple, certains modems-routeurs envoient ce type de requêtes au démarrage. Elles permettent à cet équipement, nouvel arrivant sur le réseau, de vérifier que son adresse IP n'existe pas déjà, ce qui évite des conflits par doublon d'adresse IP.

L'interface expéditrice de la requête n'attend aucune réponse. La mise à jour de la mémoire tampon des systèmes connectés au réseau est alors assurée. Les commutateurs sont informés de l'existence de l'adresse MAC de la machine en question.

L'ensemble de ces actions assure une plus grande rapidité ultérieure de connexion au réseau. Une multitude d'émission de ces types de requêtes peut être un indicateur de câble défectueux entraînant des reconnexions fréquentes.   
Lorsqu’une adresse IP change d'interface physique sur le même réseau, il est possible que les caches ARP empêchent la communication avec cette nouvelle interface jusqu'à l'expiration du cache ARP.   
Pour limiter cet inconvénient, les hôtes envoient un *gratuitous* ARP (ARP non sollicité) pour avertir les hôtes de la nouvelle association.

**Exemple de cache ARP**  
Voici un exemple de sortie de la commande arp sur une machine Linux, l'option -n permet de ne pas résoudre les adresses IP en noms d’hôtes.

**-> arp –an**  
**? (192.168.2.109) à 00:23:69:15:28:51 [ether] sur wlan1**  
**? (10.20.30.100) à 00:22:19:dd:0b:65 [ether] sur eth0**

### ARP et la passerelle par défaut

Une station envoie son paquet à une passerelle par défaut lorsque le paquet est à destination d'un autre réseau IP.

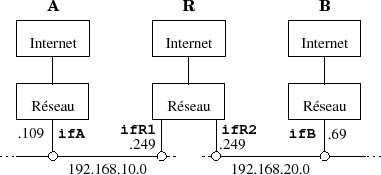
Le protocole ARP jour un rôle important dans ce mécanisme.  
En effet, comment envoyer un paquet IP à destination de 206.167.46.100 en voulant le transférer à 192.168.1.1 (la passerelle par défaut) sans modifier l'adresse IP du paquet ?   
On l'envoie dans une trame ayant pour adresse destination, l'adresse MAC du routeur (passerelle).

Lorsque l’hôte doit envoyer un paquet en dehors de son réseau IP, il scrute sa configuration IP afin de trouver l'adresse de la passerelle par défaut (*Default* *Gateway*) que lui a été attribué. Par la suite, il regarde dans sa table ARP, l'adresse MAC correspondante à l'adresse IP de sa passerelle par défaut …

* Si celle-ci existe dans la table ARP, l’hôte encapsule son paquet IP dans une trame à destination de l'adresse MAC de la passerelle ;
* Si celle-ci n'existe pas, il lance une procédure de résolution d'adresse ARP.

### ARP en pratique

Dans cet exemple, deux réseaux privés de la RFC 1918 ... 192.168.10.0 et 192.168.20.0 et l'hypothèse est que la passerelle fonctionne comme une machine Linux qui ferait du routage entre deux de ses interfaces.



Ce tableau résume l'adressage physique et logique de la situation ...

|  |  |  |
| --- | --- | --- |
| Interface | Adresse MAC | Adresse IP |
| ifA | 08:00:20:20:CF:AF | 192.168.10.109 |
| ifB | 00:01:E6:A1:07:64 | 92.168.20.69 |
| 1ifR1 | 00:06:5B:0F:5A:1F | 192.168.10.249 |
| ifR2 | 00:06:5B:0F:5A:20 | 192.168.20.249 |

On tient compte des hypothèses suivantes ...

* H1 -- Les caches ARP des machines A, B et R sont vides ;
* H2 -- L’hôte A a connaissance d'une route vers le réseau 192.168.20 passant par 192.168.10.249 et réciproquement la machine B voit le réseau 192.168.10.0 via le 192.168.20.249 ;
* H3 -- L’hôte A a connaissance de l'adresse IP de l’hôte B.

L’hôte A envoie un datagramme à l’hôte B ...

* **Étape 1**  
   A applique l'algorithme de routage et comprend que la partie réseau de l'adresse de l’hôte B n'est pas dans le même réseau local (192.168.10/24 et 192.168.20/20 diffèrent).  
  H2 explicite qu'une route existe pour atteindre ce réseau, passant par R.   
  L'adresse IP de R est dans le même réseau, A peut donc atteindre R par un routage direct.   
  La conséquence de H1 implique que pour atteindre R directement, il nous faut d'abord déterminer son adresse physique.   
  Le protocole ARP doit être utilisé.  
  A envoie en conséquence une trame ARP comportant les éléments suivants …  
  **Adresse MAC de l’envoyeur** -- 08:00:20:20:CF:AF  
  **Adresse IP de l’envoyeur** -- 192.168.10.109  
  **Adresse MAC du destinataire** -- FF:FF:FF:FF:FF:FF  
  **Adresse IP du destinataire** -- 192.168.10.249  
  Avec un champ OPERATION qui contient la valeur 1, comme question ARP.  
  Remarque ... L'adresse IP destination est celle de R.
* **Étape 2**  
  R répond à la question ARP par une réponse ARP (OPERATION contient 2) et un champ complété …  
  **Adresse MAC de l’envoyeur** -- 00:06:5B:0f:5A:1F  
  **Adresse IP de l’envoyeur** -- 192.168.10.249  
  **Adresse MAC du destinataire** -- HA 08:00:20:20:cf:af  
  **Adresse IP du destinataire** -- 192.168.10.109
* **Étape 3**  
  A est en mesure d'envoyer son datagramme à B en passant par R. Il s'agit de routage indirect puisque l'adresse de B n'est pas sur le même réseau.  
  Les adresses MAC et IP se répartissent maintenant comme ceci …  
  **Adresse IP source** -- 192.168.10.109  
  **Adresse IP de destination** -- 192.168.20.69  
  **Adresse MAC source** -- 08:00:20:20:CF:AF  
  **Adresse MAC de destination** -- 00:06:5B:0F:5A:1F  
  Remarque ... L'adresse IP destination est celle de B.
* **Étape 4**  
  R a reçu le datagramme depuis A et à destination de B.   
  Celui-ci est sur un réseau dans lequel R se trouve également, un routage direct est donc le moyen de transférer le datagramme. Pour la même raison qu'à l'étape 1 R n'a pas l'adresse MAC de B et doit utiliser ARP pour obtenir cette adresse ...  
  **Adresse MAC de l’envoyeur** -- 00:06:5B:0F:5A:20  
  **Adresse IP de l’envoyeur** -- 192.168.20.249  
  **Adresse MAC du destinataire** -- FF:FF:FF:FF:FF:FF  
  **Adresse IP du destinataire** -- 192.168.20.69
* **Étape 5**  
  Et la réponse ARP …  
  **Adresse MAC de l’envoyeur** -- 00:01:E6:A1:07:64  
  **Adresse IP de l’envoyeur** -- 192.168.20.69  
  **Adresse MAC du destinataire** -- 00:06:5B:0F:5A:20  
  **Adresse IP du destinataire** --192.168.20.249
* **Étape 6**  
  Enfin, dans cette dernière étape, R envoie le datagramme en provenance de A, à B …  
  **Adresse IP source** -- 192.168.10.109  
  **Adresse IP de destination** -- 192.168.20.69  
  **Adresse MAC source** -- 00:06:5b:0f:5a:20  
  **Adresse MAC de destination** -- 00:01:E6:A1:07:64

**Remarques** …   
En comparaison avec le datagramme de l'étape 3, les adresses IP n'ont pas changé mais les adresses MAC diffèrent complètement.  
Si A envoie un deuxième datagramme, les caches ARP ont les adresses MAC utiles et donc les étape 1, 2, 4 et 5 deviennent inutiles.

## Protocole ICMP

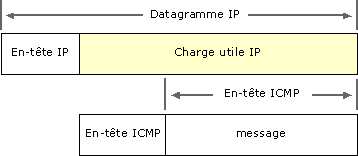
ICMP (*Internet Control Messaging Protocol*) est le protocole de gestion de la suite TCP/IP, qui est requis dans chaque mise en œuvre TCP/IP et autorise deux nœuds d'un réseau IP à partager des informations d'erreur ainsi que l'état IP.   
ICMP est utilisé par l'utilitaire *ping* pour déterminer l'accessibilité d'un système distant

Avec ICMP, les hôtes et les aiguilleurs (*routers*) qui utilisent la communication IP peuvent faire état des erreurs et échanger des informations sur le contrôle limité et sur l'état.

Les messages ICMP sont généralement envoyés automatiquement dans l'un des cas suivants ...

* Un p**aquet (datagramme) IP ne peut pas atteindre sa destination** ;
* Un **aiguilleur IP (passerelle) ne peut pas transmettre les paquet**s et donne des informations sur l'état de la transmission ;
* Un **aiguilleur IP redirige l'hôte d'envoi pour utiliser un meilleur itinéraire** vers la destination.

Les messages ICMP sont encapsulés et envoyés dans les paquets IP ...



### Messages ICMP

Différents types de messages ICMP sont identifiés dans l'en-tête ICMP.   
Les messages ICMP étant traités dans des paquets IP, ils ne sont pas fiables.

|  |  |  |
| --- | --- | --- |
| **Message ICMP** | **Description** | |
| **Requête d'écho** (*Echo request*) | | Détermine si un nœud IP (un hôte ou un aiguilleur) est disponible sur le réseau. |
| **Réponse d'éch**o (*Echo reply*) | | Répond à une requête d'écho ICMP. |
| **Destination inaccessible** (*Destination unreachable*) | | Informe l'hôte qu'un paquet ne peut pas être livré. |
| **Extension de source** (*Source quench*) | | Informe l'hôte pour réduire la vitesse à laquelle il envoie les paquets à cause de la congestion. |
| Redirection (*Redirect*) | | Informe l'hôte d'un itinéraire préféré. |
| **Dépassement de la temporisation** (*Time* *exceeded*) | | Indique que la durée de vie TTL (Time-to-Live) d'un paquet IP a été dépassée. |

Il est possible d’utiliser la commande *ping* pour envoyer des messages de requête d'écho ICMP et enregistrer la réception des messages des réponses d'écho ICMP.   
Avec ces messages, il est possible de détecter les problèmes de communication de réseau ou d'hôte et résoudre les problèmes courants de connectivité TCP/IP.

|  |  |
| --- | --- |
| Type | Message ICMP |
| 0 | *Echo reply* |
| 3 | *Destination unreachable* |
| 4 | *Source quench* |
| 5 | *Redirect* |
| 8 | *Echo request* |
| 11 | *Time exceeded* |
| 12 | *Parameter (IP) unintelligible* |
| 13 | *TimeStamp request* |
| 14 | *TimeStamp reply* |
| 15 | *Information request* |
| 16 | *Information reply* |
| 17 | *Address mask request* |
| 18 | *Address mask reply* |

Le protocole ICMP est une norme TCP/IP obligatoire définie dans la RFC 792.

## Multidiffusion

En informatique, le terme multi-diffusion (*multicast*) définit une connexion réseau multipoint.

### Définition

On entend par multi-diffusion le fait de communiquer simultanément avec un groupe d'ordinateurs identifiés par une adresse spécifique (adresse de groupe).

### Avantages

L'avantage de ce système par rapport à la classique monodiffusion (*unicast*) devient évident quand on veut diffuser de la vidéo. En streaming on envoie une image autant de fois que l'on a de connexions simultanées. Comme résultat ... moins de perte de temps, de ressources du serveur et surtout de bande passante.

En multidiffusion le paquet n'est émis qu'une seule fois et sera aiguillé (routé) vers tous les hôtes du groupe de diffusion.

### Protocoles

En multidiffusion, le protocole IP utilise les adresses de la classe d'adresses D (224.0.0.1 à 239.255.255.254).   
Les adresses IP de multidiffusion 224.0.0.1 à 224.0.0.255 ont un rôle spécifique à utilisation locale.

Les paquets de données sont aiguillés sur le réseau selon l'adresse des destinataires encapsulée dans la trame transmise. Normalement, seuls les destinataires interceptent et décodent les paquets qui leurs sont adressés.

Exemple d'une adresse IP locale pouvant servir à une communication de multidiffusion …  
224.0.0.1 ;   
Exemple d'une adresse IP Internet pouvant servir à une communication de multidiffusion …  
239.254.254.254.

Un groupe de multidiffusion se compose d'un ensemble d’hôtes. Il est entièrement dynamique (un hôte peut rejoindre ou quitter le groupe à tout moment) et ouvert (une station peut émettre un paquet dans un groupe sans en faire partie).

Un groupe de multidiffusion est désigné par une adresse IP (de 224.0.0.1 à 239.255.255.255).   
Lorsqu'un hôte veut envoyer un paquet à un groupe multidiffusion, il envoie ce paquet à l'adresse IP identifiant ce groupe (par ex : 224.1.2.3). La réception est réalisée par un aiguilleur abonné au groupe et le paquet est alors dupliqué et renvoyé grâce à une trame de niveau 2 Multicast.

Le protocole IGMP est utilisé par le protocole IP pour l'adhésion aux groupes de multidiffusion.

### Utilisation

L'usage de la multidiffusion sur Internet est encore limité aux universités ou utilisé en interne par les fournisseurs d'accès Internet (diffusion des chaînes de télévision pour certains).  
Certaines radio web expérimentent un flux de multidiffusion pour la diffusion de leurs programmes.

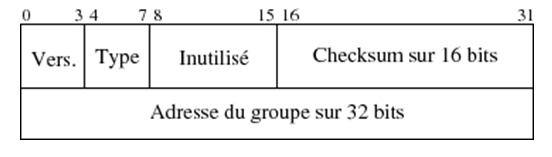
## Protocole IGMP

IGMP (*Internet Group Management Protocol*) est historiquement défini dans l'Annexe I de la RFC 1112.   
Sa raison d'être est que les datagrammes ayant une adresse de multidiffusion sont à destination d'un groupe d'utilisateurs dont l'émetteur ne connait ni le nombre ni l'emplacement. L'usage de la multidiffusion étant par définition dédié aux applications comme la radio ou la vidéo sur le réseau Internet, donc consommatrices de bande passante, il est primordial que les aiguilleurs (*routers*) poss<edent un moyen de savoir s'il y a des utilisateurs de tel ou tel groupe sur les réseaux locaux directement accessibles pour ne pas encombrer la bande passante associée avec des flux d'octets que personne n'utilise plus.

### Description de l'en-tête

IGMP est un protocole de communication entre les aiguilleurs susceptibles de transmettre des datagrammes de multidiffusiont et des hôtes qui veulent s'enregistrer dans tel ou tel groupe.   
IGMP est encapsulé dans IP avec le protocole numéro 2.

LA taille d’un datagramme IGMP est fixe (contrairement à ICMP) … seulement 2 mots de 4 octets.



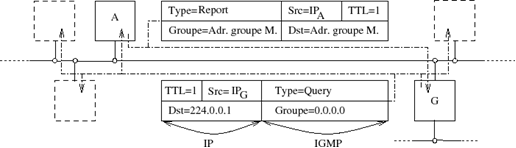
|  |  |  |  |
| --- | --- | --- | --- |
| **Champ** | | Description | |
| **Type** | | Ce champ prend deux valeurs ...  **1** pour indiquer qu'il s'agit d'une question (*query* d'un aiguilleur)  **2** pour indiquer qu'il s'agit de la réponse d'un hôte | |
| Inutilisé | | **Inutilisé** | |
| Somme de contrôle (*Checksum*) | | Somme de contrôle afin de valider l’intégralité du datagramme | |
| Adresse | | Adresse de multidiffusion (classe D) à laquelle appartient l'hôte | |
|  |  | |

### Fonctionnement du protocole

La RFC 1112 précise que les aiguilleurs de multidiffusion envoient des messages de questionnement (Type = Queries) afin de connaître quels sont les éventuels hôtes appartenant à quel groupe. Ces questions sont envoyées à tous les hôtes des réseaux locaux (LANs) directement raccordés à l'aide de l'adresse multicast du groupe 224.0.0.1 encapsulé dans un datagramme IP ayant un champ TTL = 1.   
Tous les hôtes, par hypothèse, susceptibles de joindre un groupe multicast écoutent ce groupe.

Les hôtes, dont les interfaces ont été correctement configurées, répondent à une question par autant de réponses que de groupes auxquels ils appartiennent sur l'interface réseau qui a reçu la question. Afin d'éviter une tempête de réponses chaque hôte met en œuvre la stratégie suivante …

* Un hôte ne répond pas immédiatement à la question reçue.   
  Pour chaque groupe auquel il appartient, il attend un délai compris entre 0 et 10 secondes, calculé aléatoirement à partir de l'adresse IP de monodiffusion (*unicast*) de l'interface qui a reçu la question, avant de renvoyer sa réponse ;
* La réponse envoyée est écoutée par tous les membres du groupe appartenant au même réseau local.   
  Tous ceux qui s'apprêtaient à envoyer une telle réponse au serveur en interrompent le processus pour éviter une redite.   
  L’aiguilleur ne reçoit ainsi qu'une seule réponse pour chaque groupe, et pour chaque réseau local, ce qui lui suffit pour justifier le routage demandé.



Il y a deux exceptions à la stratégie ci-dessus ...

* La première est que si une question est reçue alors que le compte à rebours pour répondre à une réponse est en cours, il n'est pas interrompu.
* La deuxième est qu'il n'y a jamais de délai appliqué pour l'envoi de datagramme portant l'adresse du groupe de base 224.0.0.1.

Pour rafraîchir leur connaissance des besoins d’aiguillage, les aiguilleurs envoient leurs questions avec une fréquence très faible -- de l'ordre de la minute -- afin de préserver au maximum la bande passante du réseau.   
Si aucune réponse ne leur parvient pour tel ou tel groupe demandé précédemment, l’aiguillage s'interrompt.   
Quand un hôte rejoint un groupe, il envoie immédiatement une réponse (type = Report) pour le(s) groupe(s) qui l'intéresse plutôt que d'attendre une question de l’aiguilleur.   
Au cas où cette réponse se perdrait, il est recommandé d'effectuer une réémission dans un court délai.

**Remarques** …

* Sur un réseau local sans aiguilleur pour lae multidiffusion, le seul trafic IGMP est celui des hôtes demandant à rejoindre tel ou tel groupe ;
* Il n'y a pas de report pour quitter un groupe ;
* L’étendue d'adresses multicast entre 224.0.0.0 et 224.0.0.225 est réservée aux applications utilisant une valeur de 1 pour le champ TTL (administration et services au niveau du réseau local) ;
* Les aiguilleurs ne doivent pas transmettre de tels datagrammes ;
* Il n'y a pas de message ICMP sur les datagrammes ayant une adresse de destination du type de multidiffusion.

En conséquence, les applications qui utilisent la multidiffusion (avec une adresse supérieure à 224.0.0.225,) pour découvrir des services, doivent avoir une stratégie afin d’augmenter la valeur du champ TTL en cas d’absence de réponse.

### Adressage multidiffusion

En règle générale l'adressage de multidiffusion est employé pour s'adresser en une seule fois à un groupe d’hôtes.   
Dans le cas d'un serveur vidéo/audio, cette approche induit une économie de moyen et de bande passante évidente quand on la compare à une démarche de monodiffusion -- un seul datagramme est aiguillé vers tous les clients intéressés au lieu d'un envoi massif d'autant de datagrammes qu'il y a de clients.

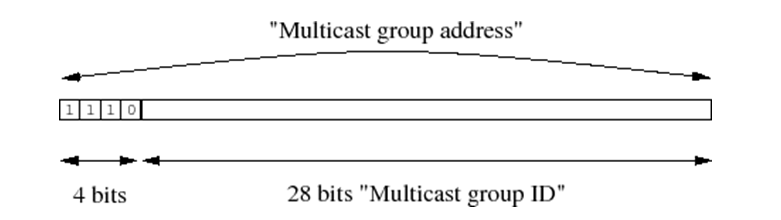
Les adresses de type multicast ont donc la faculté d'identifier un groupe d'hôtes qui partagent un protocole commun par opposition à un groupe d’hôtes qui partagent un réseau commun.   
La plupart des adresses de multidiffusion allouées le sont pour des applications particulières comme par exemple la découverte de aiguilleurs ou encore la radio ou le téléphone/vidéo sur Internet.

Lesadresse de multidiffusion les plus couramment utilisées sur un réseau local sont ...

* **224.0.0.1**  
  Tous les hôtes sur ce sous-réseau ;
* **224.0.0.2**  
  Tous les aiguilleurs sur ce sous-réseau.

### Adresse de groupe de multidiffusion

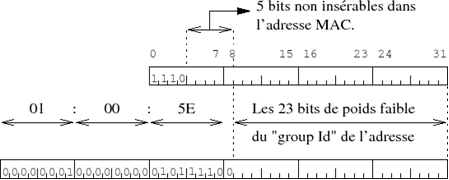
Si une adresse multicast démarre avec les bits 1110, les 28 bits suivants diffèrent de celle des classes A, B et C.



* Les 28 bits n'ont pas de structure particulière.   
  On continue par contre à utiliser la notation décimale pointée ...  
  224.0.0.0 à 239.255.255.255 ;
* Un groupe d'hôtes qui partagent un protocole commun utilisant une adresse de multidiffusion commune peuvent être répartis n'importe où sur le réseau ;
* L'appartenance à un groupe est dynamique, les hôtes qui le désirent joignent et quittent le groupe comme ils veulent ;
* Il n'y a pas de restriction sur le nombre d'hôtes dans un groupe et un hôte n'a pas besoin d'appartenir à un groupe pour lui envoyer un message.

### Adresse multicast et adresse MAC

Un hôte Ethernet quelconque doit être configuré afin d’accepter la multidiffusion, c'est à dire pour accepter les trames contenant un datagramme munis d'une adresse IP de destination qui est une adresse de multidiffusion.   
Cette opération sous-entend que l’interface réseau sait faire le tri entre les trames.   
En effet, les trames de multidffusion ont une adresse MAC particulière -- elles commencent forcément par les trois octets 01:00:5E.   
Ceux-ci ne désignent pas un constructeur en particulier mais sont la propriété de l'ICANN (ex IANA).   
Il reste donc trois octets dont le bit de poids fort est forcément à 0 pour désigner les adresses de multidiffusion (contrainte de la RFC 1700), ce qui conduit au schéma suivant …



### Adresses IP de multidiffusion réservées

L'IANA (*Internet Assigned Numbers Authority*) est l'organisation chargée de gérer les noms de domaine de niveau supérieur et l'adressage IP.  
Cette dernière a assigné pour l'adressage de multidiffusion l'étendue d'adresse de classe D allant de 224.0.0.0 à 239.255.255.255. Les paquets qui ont ce type d'adresse ne doivent jamais être transmis par un aiguilleur, ils doivent restés sur le sous réseau en question. Ils sont toujours transmis avec un temporisateur (*Time To Live* ou TTL) égal à 1.

Le tableau suivant présente une liste non exhaustive d'adresses de Classe D réservées pour la multidiffusion.

|  |  |
| --- | --- |
| Adresse IP  multidestinataire | Description |
| 224.0.0.0 | Adresse de base réservée |
| 224.0.0.1 | Groupe multidestinataire de tous les hôtes  (contient tous les systèmes du sous réseau) |
| 224.0.0.2 | Groupe multidestinataire de tous les aiguilleurs (contient tous les routeurs du sous réseau) |
| 224.0.0.5 | Aiguilleurs OSPF (open shortest path first) |
| 224.0.0.6 | Aiguilleurs OSPF désignés |
| 224.0.0.9 | Groupe RIP v2 |
| 224.0.0.12 | Serveurs ou agent de relais DHCP |
| 224.0.0.24 | Adresse du groupe de serveurs WINS |

* ***Globally scoped Address***  
  Ce terme désigne les adresses de 224.0.1.0 à 238.255.255.255.   
  Ces dernières peuvent être utilisées pour des données de multidiffusion entre des organisations et ce à travers Internet. Par exemple l'adresse 224.0.1.1 est réservée pour le protocole NTP (*Network Time Protocol*) ;
* ***Limited scope Addresses***  
  Ce terme désigne les adresses de 239.0.0.0 à 239.255.255.255.   
  D'après la RFC 2365 ,ces dernières sont réservées à un groupe local ou à une organisation.   
  La plupart du temps, les aiguilleurs sont configurés avec des filtres interdisant le trafic de multidiffusion sur cette étendue d'adresse.   
  Dans un système autonome ou un domaine, cette étendue peut être subdivisée de façon à ce que ces limites locales de multidiffusion puissent être définies ;
* ***Global Addressing***  
  D'après la RFC 2770, l’étendue d'adresse 233.0.0.0 /8 est réservée aux adresses IP statiques des organisations qui possèdent un numéro d'AS réservé (*Autonomous System*).  
   Ainsi le numéro d'AS du domaine est incorporé au second et au troisième octet de la plage 233.0.0.0 /8.   
  Comme exemple … L'AS 62010 s'écrit en hexadécimal F23A, en séparant nos deux octets F2 et 3A on obtient en décimal 242 et 58 ce qui nous conduit à un sous réseau 233.242.58.0 qui sera réservé pour l'utilisation de l'AS 62010.

# Couche Transport

La couche transport remplit également d’autres fonctions …

* Elle permet à de **nombreuses applications de communiquer sur le réseau au même moment**, sur un même périphérique ;
* elle **vérifie**, si cela est nécessaire, **que toutes les données sont reçues de façon fiable et dans l’ordre** par l’application voulue ;
* elle **utilise des mécanismes de gestion des erreurs**.

## Rôle et fonction de la couche transport

### Rôles de la couche transport

La couche transport segmente les données et se charge du contrôle nécessaire au réassemblage de ces blocs de données ans les divers flux de comunication. Pour ce faire, il doit …

* effect**uer un suivi des communications individuelles entre les applications** résidant sur les hôtes source et de destination ;
* **segmenter les données et gérer chaque bloc individuel** ;
* **réassembler les segments en flux de données d’application** ;
* **identifier les différentes applications**.

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

**Segmentation des données**   
Chaque application crée un flux de données à envoyer vers une application distante. Ces données doivent donc être préparées pour être expédiées sur le support sous forme de blocs faciles à gérer.   
Les protocoles de la couche transport décrivent les services qui segmentent les données provenant de la couche application. Il s’agit notamment de l’encapsulation devant s’appliquer à chaque bloc de données.   
Des en-têtes doivent être ajoutés à chaque bloc de données d’application au niveau de la couche Transport pour indiquer à quelle communication il est associé.

**Reconstitution des segments**   
L’hôte recevant les blocs de données peut les diriger vers l’application appropriée.   
Il faut en outre que ces blocs de données individuels puissent être réassemblés dans un flux de données complet utile à la couche Application.   
Les protocoles intervenant au niveau de la couche Transport gèrent la façon dont les informations d’entête de la couche Transport servent à réassembler les blocs de données en flux qui seront transmis à la couche Application.

**Identification des applications**   
Pour que les flux de données atteignent les applications auxquelles ils sont destinés, la couche Transport doit identifier l’application cible. Pour cela, la couche Transport affecte un identificateur à chaque application.   
Les protocoles TCP/IP appellent cet identificateur un numéro de port. Chaque processus logiciel ayant besoin d’accéder au réseau se voit affecter un numéro de port unique sur son hôte. Ce numéro de port est inclus dans l’entête de la couche Transport afin de préciser à quelle application ce bloc de données est associé.

La couche Transport fait le lien entre la couche Application et la couche inférieure responsable de la transmission réseau. Cette couche accepte les données provenant de plusieurs conversations et les fait descendre vers les couches inférieures sous forme de blocs faciles à gérer pouvant au final faire l’objet d’un multiplexage sur le support.   
Les applications n’ont pas besoin de connaître les détails du fonctionnement du réseau utilisé. Les applications génèrent des données qui sont envoyées d’une application à une autre sans se soucier du type de l’hôte de destination, du type de support que les données doivent emprunter, du chemin suivi par ces données, de l'encombrement d’une liaison ni de la taille du réseau.

En outre, les couches inférieures ignorent ce que les applications envoient comme données sur le réseau. Leur responsabilité se limite à livrer les données au périphérique approprié.   
La couche transport trie ensuite ces blocs avant de les acheminer vers l’application voulue.

**Variabilité des besoins en données**  
Parce que des applications différentes ont des besoins différents, il existe plusieurs protocoles pour la couche Transport. Dans le cas de certaines applications, les segments doivent arriver dans un ordre bien précis pour être traités correctement. Pour d’autres applications, il faut que toutes les données soient arrivées pour qu’il soit possible de traiter n’importe laquelle d’entre elles. D’autres applications, enfin, tolèrent la perte d’une certaine quantité de données lors de la transmission sur le réseau.

**Séparation de communications multiples**  
Un hôte connecté à un réseau peut envoyer et recevoir simultanément des courriels et messages instantanés, afficher des sites Web et passer un appel téléphonique par voix sur IP. Chacune de ces applications envoie des données sur le réseau et en reçoit simultanément.   
Pourtant, les données de l’appel téléphonique ne sont pas orientées vers le fureteur Web et le texte des messages instantanés ne finit pas dans un courriel.  
De plus, les informations contenues dans un courriel ou une page Web doivent avoir été intégralement reçues et affichées pour présenter un intérêt pour l’utilisateur.   
On considère certains retards comme acceptables pour veiller à ce que l’ensemble des informations soit reçu et présenté.

### Fonctions de la couche Transport

Tous les protocoles de la couche Transport ont des fonctions essentielles communes …

**Segmentation et reconstitution**  
La plupart des réseaux limitent la quantité de données pouvant être incluses dans une même unité de données de protocole. La couche Transport divise les données d’application en blocs de données d’une taille adéquate.   
Une fois ces blocs parvenus à destination, la couche Transport réassemble les données avant de les envoyer vers l’application ou le service de destination.

**Multiplexage de conversations**De nombreux services ou applications peuvent s’exécuter sur chaque hôte sur le réseau. Une adresse, appelée port, est affectée à chacun de ces services ou applications afin que la couche transport puisse déterminer à quel service ou application les données se rapportent.

**Établissement d’une session**   
La couche Transport est en mesure d’orienter la connexion en créant des sessions entre les applications. Ces connexions préparent les applications à communiquer entre elles avant le transfert des données. Dans ces sessions, il est possible de gérer avec précision les données d’une communication entre deux applications.

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

**Livraison dans un ordre défini**  
Étant donné que les réseaux fournissent une multitude de routes dont les délais de transmission varient, il se peut que les données arrivent dans le désordre.   
En numérotant et en ordonnant les segments, la couche transport s’assure que ces segments sont réassemblés dans le bon ordre.

**Contrôle du flux**  
Les hôtes du réseau disposent de ressources limitées, par exemple en ce qui concerne la mémoire ou la bande passante. Quand la couche Transport détermine que ces ressources sont surexploitées, certains protocoles peuvent demander à l’application qui envoie les données d’en réduire le flux.   
Ceci s’effectue au niveau de la couche Transport en régulant la quantité de données que la source transmet sous forme de groupe. Le contrôle du flux contribue à prévenir la perte de segments sur le réseau et à rendre inutiles les retransmissions.

## Protocole TCP

TCP (qui signifie *Transmission Control Protocol*, soit en français: Protocole de Contrôle de Transmission) est un des principaux protocoles de la couche Transport du modèle TCP-IP. Il permet, au niveau des applications, de gérer les données en provenance (ou à destination) de la couche inférieure du modèle (c'est-à-dire le protocole IP).

Lorsque les données sont fournies au protocole IP, celui-ci les encapsule dans des datagrammes IP, en fixant le champ protocole à 6 (pour savoir que le protocole en amont est TCP).

TCP est un protocole orienté connexion, c'est-à-dire qu'il permet à deux hôtes qui communiquent de contrôler l'état de la transmission.

Les caractéristiques principales du protocole TCP sont les suivantes :

* TCP permet de **remettre en ordre les datagrammes en provenance du protocole IP**;
* TCP permet de **vérifier le flot de données afin d'éviter une saturation du réseau** ;
* TCP permet de **formater les données en segments de longueur variable** afin de les remettre au protocole IP;
* TCP permet de **multiplexer les données**, c'est-à-dire de faire circuler simultanément des informations provenant de sources (applications par exemple) distinctes sur une même ligne;
* TCP permet enfin l'**initialisation et la fin d'une communication de manière courtoise**.

### Objectif de TCP

Grâce au protocole TCP, les applications peuvent communiquer de façon sûre (grâce au système d'accusés de réception du protocole TCP), indépendamment des couches inférieures. Cela signifie que les aiguilleurs (qui travaillent sur la couche Réseau) ont pour seul rôle l'acheminement des données sous forme de datagrammes, sans se préoccuper du contrôle des données, car celui-ci est réalisé par la couche Transport (plus particulièrement par le protocole TCP).

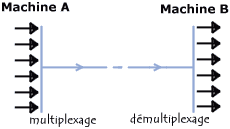
Lors d'une communication à travers le protocole TCP, les deux hôtes doivent établir une connexion.   
L’hôte émetteur (celui qui demande la connexion) est appelé client, tandis que l’hôte récepteur est appelé serveur.   
On dit qu'on est alors dans un environnement client-serveur.

Les hôtes dans un tel environnement communiquent en mode connecté, c'est-à-dire que la communication se fait dans les deux sens.   
Pour permettre le bon déroulement de la communication et de tous les contrôles qui l'accompagnent, les données sont encapsulées, c'est-à-dire qu'on ajoute aux paquets de données un en-tête qui va permettre de synchroniser les transmissions et d'assurer leur réception.

Une autre particularité de TCP est de pouvoir réguler le débit des données grâce à sa capacité à émettre des messages de taille variable, ces messages sont appelés segments.

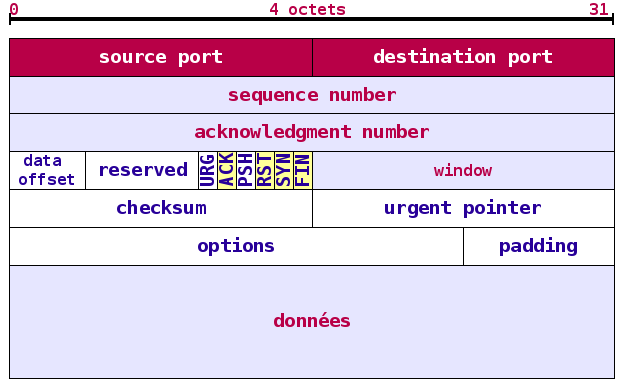
**Fonction de multiplexage**  
TCP permet d'effectuer une tâche importante: le multiplexage/démultiplexage, c'est-à-dire faire transiter sur une même ligne des données provenant d'applications diverses ou en d'autres mots mettre en série des informations arrivant en parallèle.

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.



### Format des données

Un segment TCP est constitué comme suit …



Signification des différents champs …

* **Port Source (*Source port*)** -- 16 bits)  
  Port relatif à l'application en cours sur l’hôte source ;
* **Port Destination (Destination port)** -- 16 bits  
  Port relatif à l'application en cours sur l'hôte de destination ;
* **Numéro d'ordre (*Sequence number*)** -- 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 (*Acknowlegment* *number*)** -- 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 (*Data* *offset*)** -- 4 bits  
  Le contenu de ce champ 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 (*Reserved*)** -- 6 bits  
  Ce champ (inutilisé actuellement) est prévu pour faire éventuellement évoluer le protocole ;
* **Drapeaux (*flags*)** (6x1 bit)  
  Les drapeaux représentent des informations supplémentaires …
  + **URG**    
    Cet indicateur est utilisé pour informer une station de réception que certaines données d'un segment sont urgentes et doivent être classées par ordre de priorité ;
  + **ACK**  
    Si cet indicateur est positionné à 1, le paquet est un accusé de réception ;
  + **PSH (*PUSH*)**  
    Cet indicateur informe l'hôte de réception que les données doivent être transmises immédiatement à l'application de réception ;
  + **RST (*RESET*)**  
    La réception d’un paquet dont l’indicateur est positionné à 1 demande la réinitialisation de la connexion ;
  + **SYN**  
    Cet indicateur signifie une demande d'établissement de connexion ;
  + **FIN**  
    Si cet indicateur est positionné à 1, la connexion s'interrompt immédiatement ;
* **Fenêtre (*Window*)** -- 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)** -- 16 bits  
  Ce champ vise la validation de l’intégrité d’un segment.   
  La somme de contrôle est calculée en effectuant la somme des champs de données de l'entête ;
* **Pointeur d'urgence (*Urgent* *pointer*)** -- 16 bits  
  Ce champ a comme fonction d’indiquer le numéro d'ordre à partir duquel l'information devient urgente
* **Options** -- Taille variable  
  Des options diverses
  + **Bourrage (*Paddind*)**  
    On remplit l'espace restant après les options avec des zéros pour avoir une longueur multiple de 32 bits
  + **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. À la réception d'un segment de donnée, l’hôte récepteur va retourner un segment de donnée dont l’indicateur 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.

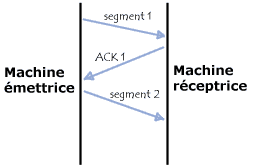
De plus, grâce à une minuterie déclenchée dès réception d'un segment au niveau de l’hôte émetteur, le segment est réexpédié dès que le temps imparti est écoulé, car dans ce cas l’hôte émettreur considère que le segment est perdu.

Toutefois, si le segment n'est pas perdu et qu'il arrive tout de même à destination, l’hôte récepteur saura, grâce au numéro d'ordre, qu'il s'agit d'un doublon et ne conservera que le dernier segment arrivé à destination...

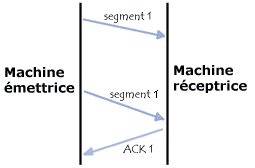
### Fiabilité des transferts

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.   
À la réception d'un segment de donnée, l’hôte récepteur va retourner un segment de donnée dont l’indicateur 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 l’hôte émetteur, le segment est réexpédié dès que le temps imparti est écoulé, car dans ce cas l'hôte émetteur 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.

### Établissement d'une connexion

Étant donné que ce processus de communication, qui se fait grâce à une émission de données et d'un accusé de réception, est basé sur un numéro d'ordre, il faut que les hôtes émetteurs et récepteurs (client et serveur) connaissent le numéro d'ordre initial de l'autre hôte.

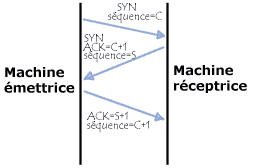
L'établissement de la connexion entre deux applications se fait souvent selon le schéma suivant ...

* Les ports TCP doivent être ouverts ;
* L'application sur le serveur est passive, c'est-à-dire que l'application est à l'écoute, en attente d'une connexion ;
* L'application sur le client fait une requête de connexion sur le serveur dont l'application est en ouverture passive.   
  L'application du client est dite en ouverture active.

Les deux hôtes doivent donc synchroniser leurs séquences grâce à un mécanisme communément appelé **poignée de main en trois temps** (*three ways handshake*) que l'on retrouve aussi lors de la clôture de session.

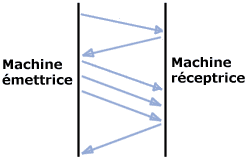
Ce dialogue permet d'initier la communication et se déroule en trois temps, comme sa dénomination l'indique :

* Dans un premier temps l’hôte émetteur (le client) transmet un segment dont l’indicateur SYN est à 1 (pour signaler qu'il s'agit d'un segment de synchronisation), avec un numéro d'ordre *N*, que l'on appelle numéro d'ordre initial du client ;
* Dans un second temps l’hôte récepteur (le serveur) reçoit le segment initial provenant du client, puis lui envoie un accusé de réception, c'est-à-dire un segment dont l’indicateur ACK est à 1 et l’indicateur SYN est à 1 (car il s'agit là encore d'une synchronisation).  
  Ce segment contient le numéro d'ordre de cet hôte (du serveur) qui est le numéro d'ordre initial du client.   
  Le champ le plus important de ce segment est le champ accusé de réception qui contient le numéro d'ordre initial du client, incrémenté de 1.
* Enfin, le client transmet au serveur un accusé de réception, c'est-à-dire un segment dont l’indicateur ACK est à 1, dont l’indicateur SYN est à 0 (il ne s'agit plus d'un segment de synchronisation).   
  Son numéro d'ordre est incrémenté et le numéro d'accusé de réception représente le numéro d'ordre initial du serveur incrémenté de 1.



Suite à cette séquence comportant trois échanges les deux hôtes sont synchronisés et la communication peut commencer.

**Méthode de la fenêtre glissante**  
Dans de nombreux cas, il est possible de limiter le nombre d'accusés de réception, afin de désengorger le réseau, en fixant un nombre de séquence au bout duquel un accusé de réception est nécessaire. Ce nombre est en fait stocké dans le champ fenêtre de l'en-tête TCP/IP.   
On appelle effectivement cette méthode **méthode de la fenêtre glissante** car on définit en quelque sorte une fourchette de séquences n'ayant pas besoin d'accusé de réception, et celle-ci se déplace au fur et à mesure que les accusés de réception sont reçus.



De plus, la taille de cette fenêtre n'est pas fixe. En effet, le serveur peut inclure dans ses accusés de réception en stockant dans le champ fenêtre la taille de la fenêtre qui lui semble la plus adaptée.   
Ainsi, lorsque l'accusé de réception indique une demande d'augmentation de la fenêtre, le client va déplacer le bord droit de la fenêtre.

http://www.commentcamarche.net/internet/images/plus.gif

Par contre, dans le cas d'une diminution, le client ne va pas déplacer le bord droit de la fenêtre vers la gauche mais attendre que le bord gauche avance (avec l'arrivée des accusés de réception).

**Fin d'une connexion**  
Le client peut demander à mettre fin à une connexion au même titre que le serveur.   
La fin de la connexion se fait de la manière suivante ...

* Une des hôtes envoie un segment avec l’indicateur FIN à 1 et l'application se met en état d'attente de fermeture de session, c'est-à-dire qu'elle finit de recevoir le segment en cours et ignore les suivants
* Après réception de ce segment, l'autre hôte envoie un accusé de réception avec l’indicateur FIN à 1 et continue d'expédier les segments en cours.  
   Suite à cela la machine informe l'application qu'un segment FIN a été reçu, puis envoie un segment FIN à l'autre machine, ce qui clôture la connexion...

### Protocole UDP

Le protocole UDP ou *User Datagram Protocol* est un des principaux protocoles de télécommunication utilisé par Internet. Il fait partie de la couche Transport de la pile de protocole TCP/IP: dans l'adaptation approximative de cette dernière au modèle OSI, il appartiendrait à la couche 4, comme TCP.   
Il est détaillé dans la RFC 768.

Le rôle de ce protocole est de permettre la transmission de paquets de manière très simple entre deux entités, chacune étant définie par une adresse IP et un numéro de port (pour différencier différents utilisateurs sur le même hôte). Contrairement au protocole TCP, il travaille en mode non connecté: il n'y a pas de moyen de vérifier si tous les paquets envoyés sont bien arrivés à destination et dans quel ordre (le séquencement peut cependant être assuré par un protocole réseau de couche inférieure).   
Il n'est prévu aucun contrôle de flux ni contrôle de congestion. C'est pour cela qu'il est souvent décrit comme étant un protocole non fiable.   
En revanche, pour un paquet UDP donné, l'exactitude du contenu des données est assurée grâce à une somme de contrôle (*checksum*).

**Structure d'un segment UDP**

Le paquet UDP est encapsulé dans un datagramme IP.   
Il comporte un en-tête suivi des données proprement dites à transporter.

|  |  |  |
| --- | --- | --- |
| En-tête IP | En-tête UDP | Données |

L'en-tête (*header* en anglais) d'un segment UDP est bien plus simple que celui de TCP …

|  |  |
| --- | --- |
| Port Source (16 bits) | Port Destination (16 bits) |
| Longueur (16 bits) | Somme de contrôle (16 bits) |
| Données (longueur variable) | |

Il contient les 4 champs suivants …

* **Port Source (*Source* *port*)** -- 16 bits  
  Ce champ indique depuis quel port le segment a été envoyé ;
* **Port de Destination (*Destination* *port*)** -- 16 bits  
  Ce champ indique à quel port le paquet doit être envoyé ;
* **Longueur (Length**) -- 16 bits  
  il indique la longueur totale du segment UDP (en-tête et données).   
  La longueur minimale est donc de 8 octets (taille de l'en-tête)
* **Somme de contrôle (*Checksum*)** -- 16 bits  
  La csomme de contrôle (CRC, *Cyclic Redundancy Check*) permet de s'assurer de l'intégrité du paquet reçu.   
  Elle est calculée sur l'ensemble de l'en-tête UDP et des données mais aussi sur un pseudo en-tête (extrait de l'en-tête IP)

**Remarque** …  
La présence de ce pseudo en-tête, interaction entre les deux couches IP et UDP, est une des raisons qui font que le modèle TCP/IP ne s'applique pas parfaitement au modèle OSI.

### Utilisation

Le protocole UDP est utilisé ...

* soit de t**ransmettre des données très rapidement** et où la **perte d'une partie de ces données n'a pas grande** **importance**,
* soit de **transmettre des petites quantités de données**, là où la connexion en trois phase de TCP serait trop lourde.

Par exemple, dans le cas de la transmission de la voix sur IP, ce n'est pas grave si l'un ou l'autre paquet se perd (il existe des mécanismes de substitution des données manquantes), par contre la rapidité de transmission est un critère primordial pour la qualité d'écoute.

Exemples d'utilisation …

* le programme *traceroute* ;
* les protocoles DNS, TFTP, … ;
* les jeux en réseau ;
* La diffusion (*streaming*) :  
  Il est indispensable pour les applications multimédias de par sa faible latence.

# Protocoles de la couche Applications

## Fonctionnement général

Pour désigner les informations transmises et leur enveloppe, selon le niveau concerné, on parle de message (ou de flux) entre applications, de datagramme (ou segment) au niveau TCP, de paquet au niveau IP, et enfin, de trames au niveau de l'interface réseau (Ethernet ou Token Ring).

Les protocoles du niveau Application les plus connus sont ...

* **HTTP** (*HyperText Transfer Protocol*) permet l'accès aux documents HTML et le transfert de fichiers depuis un site WWW ;
* **FTP** (*File Transfer Protocol*) pour le transfert de fichiers s'appuie sur TCP et établit une connexion sur un serveur FTP ;
* **DNS** (*Domain Naming Service*) et **DHCP** (*Dynamic Hosts Configuration Protocol*) pour la configuration IP des hôtes ;
* **SSH**(*Secure shell*)pour les connexions à distance en émulation terminal à un hôte Linux ;
* **SMTP** (*Simple Mail Transfer Protocol*) pour la messagerie électronique (UDP et TCP).  
  Il travaille conjointement avec **POP** et **IMAP** ;
* **SNMP** (*Simple Network Management Protocol*) et NTP (*Network Time Protocol*) pour l'administration du réseau ;
* **NFS** (*Network File System*) pour le partage des fichiers Linux.

## Fonctions

La couche Application du modèle TCP/IP …

* i**dentifie et détermine la disponibilité des partenaires de communication** voulus ;
* **synchronise les applications coopératives** ;
* **établit une entente sur les procédures de reprise sur incident et de contrôle de l'intégrité** des données.

La couche Application propose un ensemble complet de programmes d'interface à Internet.   
À chaque type d'application correspond un protocole d'application qui lui est propre.

Exemple de service de la couche Application **…**

* Les programmes de **messagerie électronique** supportent les protocoles de couche application SMTP, POP3 et IMAP4 pour le courrier électronique ;
* Les programmes d'**accès distant** utilisent le protocole SSH afin de se connecter directement aux ressources distantes ;
* Les utilitaires de **transfert de fichiers** utilisent le protocole FTP pour copier et déplacer les fichiers entre emplacements distants.  
  Le protocole TFTP (*Trivial File Transfert Protocol*) est également un protocole de transferts de fichiers ;
* Les outils de **saisie et de contrôle des données réseau** utilisent le protocole SNMP et NTP ;
* Le protocole d’**accès Internet** comme HTTP ;
* Les protocoles de **gestion des hôtes** comme DHCP et DNS.

## Accès Internet

**HTTP (*Hypertext Transfect Protocol*)**  
HTTP est un protocole de communication client-serveur développé pour le World Wide Web. HTTPS (avec S pour *secured*) est la variante du HTTP sécurisée par l'usage des protocoles SSL ou TLS.

HTTP peut fonctionner sur n'importe quelle connexion fiable. Toutefois, dans les faits on utilise le protocole TCP comme couche de transport. Un serveur HTTP utilise alors par défaut le port 80 (443 pour HTTPS).

Les clients HTTP les plus connus sont les fureteurs Web permettant à un utilisateur d'accéder à un serveur contenant les données. Il existe aussi des systèmes pour récupérer automatiquement le contenu d'un site tel que les aspirateurs de site Web ou les robots d'indexation.

HTTP utilise le langage HTML  
HTML (Hypertext Markup Language) est le langage standard pour les documents Internet.

**Méthodes**

Pour le protocole HTTP, une méthode est une commande spécifiant un type de requête, c'est-à-dire qu'elle demande au serveur d'effectuer une action. En général l'action concerne une ressource identifiée par l'URL qui suit le nom de la méthode.

**Exemple d’une méthode** …  
**GET / HTTP/1.1**  
**Host: www.profsavard.info**

Il existe de nombreuses méthodes, les plus courantes étant GET, HEAD et POST ...

* **GET**  
  C'est la méthode la plus courante pour demander une ressource.   
  Une requête GET est sans effet sur la ressource, il doit être possible de répéter la requête sans effet ;
* **HEAD**  
  Cette méthode ne demande que des informations sur la ressource, sans demander la ressource elle-même ;
* **POST**  
  Cette méthode est utilisée pour transmettre des données en vue d'un traitement à une ressource (le plus souvent depuis un formulaire HTML).   
  L'URI fourni est l'URI d'une ressource à laquelle s'appliqueront les données envoyées.   
  Le résultat peut être la création de nouvelles ressources ou la modification de ressources existantes.   
  À cause de la mauvaise implémentation des méthodes HTTP par certains fureteurs (et la norme HTML qui ne supporte que les méthodes GET et POST pour les formulaires), cette méthode est souvent utilisée en remplacement de la requête PUT, qui devrait être utilisée pour la mise à jour de ressources.

**Autres méthodes**Il existe aussi d’autres méthodes qui dsont néanmoins moins utilisées ...

* **OPTIONS**  
  Cette méthode permet d'obtenir les options de communication d'une ressource ou du serveur en général ;
* **CONNECT**  
  Cette méthode permet d'utiliser un proxy comme un tunnel de communication ;
* **TRACE**  
  Cette méthode demande au serveur de retourner ce qu'il a reçu, dans le but de tester et effectuer un diagnostic sur la connexion ;
* **PUT**  
  Cette méthode permet de remplacer ou d'ajouter une ressource sur le serveur.   
  L'URI fourni est celui de la ressource en question ;
* **PATCH**  
  Cette méthode permet, contrairement à PUT, de faire une modification partielle d'une ressource ;
* **DELETE**  
  Cette méthode permet de supprimer une ressource du serveur.

Ces 3 dernières méthodes nécessitent généralement un accès privilégié.

Certains serveurs autorisent d'autres méthodes de gestion de leurs ressources (par exemple WebDAV).

**Du client au serveur**

La liaison entre le client et le serveur n'est pas toujours directe car il peut exister des hôtes intermédiaires servant de relais ...

* Un **proxy** (ou serveur mandataire) peut modifier les réponses et requêtes qu'il reçoit et peut gérer un cache des ressources demandées ;
* Une **passerelle** (ou gateway) est un intermédiaire modifiant le protocole utilisé ;
* Un **tunnel** transmet les requêtes et les réponses sans aucune modification, ni mise en cache.

**Identification**

HTTP permet l'identification du visiteur par transmission d'un nom et d'un mot de passe.

Il existe 2 modes d'identification ... Basic et Digest (RFC 26176) …

* Le premier mode transmet le mot de passe en clair, et ne doit donc être utilisé qu'avec le protocole HTTPS ;
* Le deuxième mode permet une identification sans transmettre le mot de passe en clair.   
  L'identification est cependant souvent effectuée par une couche applicative supérieure à HTTP.



## Messagerie électronique

**SMTP, POP3 et IMAP4**  
Le courrier électronique permet l'envoi de messages entre des ordinateurs sur un réseau

**Présentation**  
Le courrier électronique, aussi simple soit-il à utiliser, repose sur un fonctionnement plus compliqué que celui du web.   
Il est un des plus important service réseau. Pour la plupart des utilisateurs son fonctionnement est transparent, ce qui signifie qu'il n'est pas nécessaire de comprendre comment le courrier électronique fonctionne pour pouvoir l'utiliser.  
Néanmoins, il est important de comprendre le principe.

Linux propose un gestionnaire de courriels même dans un environnement totalement dépourvu de réseau: certains sous-systèmes Linux, tels que *cron*, peuvent utiliser le courrier électronique pour informer de ses activités.   
Pour cette raison, la plupart des distributions Linux sont livrées avec le logiciel de serveur de messagerie installé et configuré pour les activités de base.

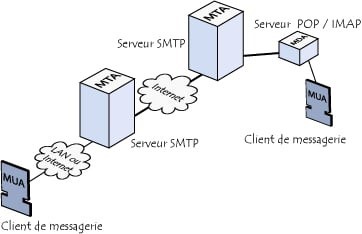
**Fonctionnement du courrier électronique**  
Le fonctionnement du courrier électronique est basé sur l'utilisation d'une boîte à lettres électronique.   
Lors de l'envoi d'un email, le message est acheminé de serveur en serveur jusqu'au serveur de messagerie du destinataire.   
Plus exactement, le message est envoyé au serveur de courrier électronique chargé du transport (nommé MTA pour *Mail Transport Agent*), jusqu'au MTA du destinataire. Sur internet, les MTA communiquent entre-eux grâce au protocole SMTP et sont logiquement appelés serveurs SMTP (parfois serveur de courrier sortant).

Le serveur MTA du destinataire délivre alors le courrier au serveur de courrier électronique entrant (nommé MDA pour *Mail Delivery Agent*), qui stocke alors le courrier en attendant que l'utilisateur vienne le relever.

Il existe deux principaux protocoles permettant de relever le courrier sur un MDA :

* le protocole POP3 (*Post Office Protocol*), le plus ancien, permettant de relever son courrier et éventuellement d'en laisser une copie sur le serveur ;
* le protocole IMAP (*Internet Message Access Protocol*), permettant une synchronisation de l'état des courriers (lu, supprimé, déplacé) entre plusieurs clients de messagerie.   
  Avec le protocole IMAP une copie de tous les messages est conservée sur le serveur afin de pouvoir assurer la synchronisation.

Ainsi, les serveurs de courrier entrant sont appelés serveurs POP ou serveurs IMAP, selon le protocole utilisé.



Par analogie avec le monde réel, les MTA font office de bureau de poste (centre de tri et facteur assurant le transport), tandis que les MDA font office de boîte à lettres, afin de stocker les messages (dans la limite de leur capacité en volume), jusqu'à ce que les destinataires relèvent leur boîte.   
Ceci signifie notamment qu'il n'est pas nécessaire que le destinataire soit connecté pour pouvoir lui envoyer du courrier.

Pour éviter que chacun puisse consulter le courrier des autres utilisateurs, l'accès au MDA est protégé par un nom d'utilisateur appelé identifiant (*login*) et par un mot de passe.

La relève du courrier se fait grâce à un logiciel appelé MUA (*Mail User Agent*).

Lorsque le MUA est un logiciel installé sur le système de l'utilisateur, on parle de client de messagerie.

**Terminoogie**  
Les termes suivants sont souvent utilisés lorsque l’on parle de services de messagerie …

* **Agent de transport de message – MTA**  
  **MTA (abréviation de** *Mail* ou *Message Transport Agent*), également appelé relais de messagerie, est un logiciel chargé de transférer les messages électroniques d'un serveur vers un client (et inversement).   
  On retrouve dans cette catégorie *Postfix*, *sendmail* ou *Exim*.
* **Agent d'utilisateur de messagerie – MUA**  
  **MUA** (abréviation de *Mail User Agent*) est un programme informatique utilisé pour accéder aux boîtes de réception de l’utilisateur et les gérer.   
  L'agent de messagerie est le terme technique utilisé pour désigner un client de messagerie.  
  Les exemples de MUA comprennent, entre autres, *Thunderbird*, *Outlook* et les interfaces de messagerie Web telles que *Gmail*, *Outlook.com*, ….
* **Agent de distribution du courrier – MDA**   
  **MDA (abréviation de** *Message* ou *Mail Delivery Agent*) est la partie logicielle qui envoie les courriers électroniques aux boîtes de réception de l’utilisateur.   
  Le MDA se charge également de l'authentification de l'utilisateur.  
  Le MDA, en tant que MTA, a également la possibilité d'effectuer des alias et des transferts. L'*aliasing* se produit lorsqu'un utilisateur souhaite recevoir un courrier électronique avec un nom différent.   
  Par exemple, au lieu de créer un compte pour [ventes@profsavard.info](mailto:ventes@profsavard.info), on peut créer un alias pointant ventes vers un ou plusieurs comptes existants. Ainsi, on peut avoir une adresse ventes publique, mais qui envoie tout le courrier électronique vers un compte d’un utilisateur particulier. Cela vous permet de changer de destination sans avoir à rendre le changement public et signifie également que les utilisateurs n'ont besoin de vérifier qu'une seule boîte aux lettres.  
  Le transfert est similaire à l'aliasing, sauf que la redirection concerne un système de messagerie différent. [tux@profsavard.info](mailto:tux@profsavard.info) pourrait vouloir envoyer tous ses courriels sur son téléphone intelligent doté de sa propre adresse électronique.   
  Certaines organisations interdisent cela pour garder le contrôle sur le courrier électronique, ou peut avoir des politiques sur l'endroit où les courriels peuvent être transférés.   
  Certains MDA permettent également aux utilisateurs de créer leurs propres règles de transfert.  
  On retrouve dans cette catégorie *Dovecot*.
* **Protocole de transfert de courrier simple – SMTP**  
  **Pour que ces composants puissent** communiquer les uns aux autres, ils doivent utiliser le même langage (ou protocole), à savoir SMTP (*Simple Mail Transfer Protocol*), tel que défini dans le RFC 2821.
* Les autres protocoles à prendre en compte sont **IMAP4 (*Internet Message Access Protocol*),** qui permet de gérer les courriers électroniques directement sur le serveur sans les télécharger sur le disque dur du client, et **POP3 (*Post Office Protocol*)**, qui permet de télécharger le courrier électronique, messages et dossiers sur l'ordinateur de l'utilisateur.

## Protocoles de gestion des courriels

### SMTP (*Simple Mail Transfer Protocol*)

SMTP permet d’envoyer des messages texte vers des hôtes disposant de services de courriel.  
L’usager communique avec le serveur à partir d’un agent utilisateur (User agent).  
Le serveur dispose d’un d’agent d’envoi et de réception (MTA ou Message Transfert Agent).  
SMTP est un protocole de transfert simple basé sur le protocole TCP.   
Il n’offre aucune interface utilisateur.

**Fonctionnement**

De l’agent utilisateur (client) à l’agent d’envoi et de réception (serveur) ou MTA (Message Transfert Agent)  
ou  
de l’agent d’envoi et de réception (serveur) à l’agent utilisateur (client).

L’utilitaire **UUENCODE** permet d’encoder n’importe quel fichier en ASCII.   
Il est disponible sur la plupart des plates-formes.

Cependant les **MTA** utilisent de plus en plus le protocole MIME (*Multipurpose Internet Mail Extensions*) car il prend aussi en charge, en plus des images et des sons, d’autres fonctionnalités comme le texte enrichi (format RTF).

Le corps d’un message MIME permet d’imbriquer des contenus.  
Un terminal texte qui ne peut afficher des images pourra afficher un texte alternatif ou encore un message peut contenir un pointeur vers un fichier stocké sur un serveur FTP permettant ainsi d’envoyer des messages plus légers.



**POP3 (*Post Office Protocol* version 3)**  
SMTP s’attend à ce que l’hôte destinataire qui reçoit le courriel soit disponible et connecté sur le réseau Internet (sinon il est impossible d’ouvrir une session TCP). Les hôtes d’extrémités doivent être ouverts en permanence.  
Pour la majorité des environnements réseau, les courriels sont reçus par des serveurs SMTP.   
Les clients communiquent avec ces serveurs de courriels avec le protocole POP3 afin de récupérer leurs messsages.

**Fonctionnement**  
De l’agent utilisateur (client POP3) à l’agent d’envoi et de réception (serveur POP3)  
ou  
De l’agent d’envoi et de réception (serveur POP3) à l’agent utilisateur (client POP3).



**IMAP4 (*Internet Message Access Protocol* rev. 4)**  
Protocole qui succède à POP3 qui possède la limite que le courriel doit être téléchargé sur la station de travail.   
En effet, POP3 ne permet pas de manipuler les courriels directement sur les serveurs.  
IMAP4 permet en plus de …

* **accéder et de manipuler des portions de courriels sans avoir à les télécharger** ;
* **voir des messages et des pièces jointes sans les télécharger** ;
* t**élécharger les messages pour les consulter hors connexion** ;
* **manipuler (créer, supprimer et renommer) des répertoires appelés boîte aux lettres** ;
* **synchroniser le client sur le serveur**.

Tout comme POP3, IMAP4 ne spécifie pas de méthode d’envoi de courrier.   
Cette fonctionnalité est offerte pour SMTP.



## Accès distant

Secure SHell (SSH) est à la fois un programme informatique et un protocole de communication sécurisé.

Le protocole de connexion impose un échange de clés de chiffrement en début de connexion.   
Par la suite, tous les segments TCP sont authentifiés et chiffrés.   
Il devient alors impossible (ou presque) d'utiliser un renifleur (*sniffer*) pour voir ce que fait un utilisateur.

Le protocole SSH a été conçu avec l'objectif de remplacer les différents protocoles non chiffrés comme rlogin, telnet, rcp et rsh.

Le protocole SSH existe en deux versions majeures : la version 1.0 et la version 2.0.

* La **première version** permet de se connecter à distance à un ordinateur afin d'obtenir un shell ou ligne de commande.   
  Cette version souffrait néanmoins de problèmes de sécurité dans la vérification de l'intégrité des données envoyées ou reçues, la rendant vulnérable à des attaques actives.   
  En outre, cette version implémentait un système sommaire de transmission de fichiers, et du port tunneling.
* La **version 2** est beaucoup plus sûre au niveau cryptographique, et possède en plus un protocole de transfert de fichiers complet, le **SSH File Transfer Protocol** (SFTP).

Habituellement le protocole SSH utilise le port TCP 22.   
Il est particulièrement utilisé pour ouvrir un shell sur un hôte distant.

SSH peut également être utilisé pour transférer des ports TCP d'un hôte vers un autre, créant ainsi un tunnel.   
Cette méthode est couramment utilisée afin de sécuriser une connexion qui ne l'est pas (par exemple le protocole de récupérations de courrier électronique POP3) en la faisant transférer par le biais du tunnel chiffré SSH.

Il est également possible de faire plusieurs sauts entre consoles SSH, c'est-à-dire ouvrir une console sur un serveur, puis, de là, en ouvrir une autre sur un autre serveur.

Les usages de base de SSH sont ...

* L'**accès à distance à la console en ligne commande** (shell), ce qui permet, entre autres, d'effectuer la totalité des opérations courantes et/ou d'administration sur un hôte distant ;
* Le **déport de l'affichage graphique** de l’hôte distant ;
* Le **transfert des fichiers en ligne de commande** ;
* Le **montage ponctuel et/ou automatique d’un répertoire distant**, soit en ligne de commande, soit à l’aide d’une interface graphique.

### Tunnel SSH

Un tunnel, dans le contexte de réseaux informatiques, est une encapsulation de données d'un protocole réseau dans un autre, situé dans la même couche du modèle en couches, ou dans une couche de niveau supérieur.

Par exemple, pour faire passer le protocole IPv6 par le réseau Internet actuel (qui est en grande partie en IPv4) on va créer un tunnel entre deux machines IPv4 ; ce tunnel, pour le protocole IPv6, semblera un simple lien point-à-point   
(un logiciel comme traceroute ne verra donc pas le tunnel).

En sécurité, on crée souvent des tunnels chiffrés, par exemple comme le fait SSH. Les données peuvent alors y circuler sans craindre d'être écoutées. Les tunnels peuvent être utilisés pour créer des réseaux privés virtuels (VPN).

Le tunnel HTTP est un cas particulier qui consiste à faire passer une connexion explicitement interdite par un pare-feu (le protocole SSH par exemple) encapsulé dans un protocole HTTP presque toujours autorisé. C'est particulièrement utile quand le pare-feu ne bloque pas simplement les ports associés (22 pour SSH par exemple), mais analyse aussi les protocoles utilisés.

## Transfert de fichiers

### FTP

En règle générale, FTP est utilisé lorsque le montage d’un partage permanent ou temporaire n’est pas souhaité ou réalisable et la vitesse de transfert est un problème. Idéal pour les situations où les fichiers sont partagés pour téléchargement uniquement, par exemple lorsqu'un instructeur sur un réseau local de laboratoire local partage des fichiers binaires avec les débutants étudiants Linux.

Le protocole de transfert de fichier (FTP) existe depuis longtemps. C'est un protocole réseau simple pour le partage de fichiers entre systèmes. Il est principalement utilisé pour partager des documents publics sur un réseau.

Lors d’une connexion à un service FTP, son processus d'authentification peut obliger à lui attribuer un nom d'utilisateur et un mot de passe valides. Une fois authentifié, en fonction de la configuration du service FTP, on pourra essentiellement rechercher et télécharger (ou téléverser) des fichiers.

Attention …  
En utilisant des informations de compte et des mots de passe pour se connecter à un service FTP, il est important de savoir qu’elles ne sont généralement pas chiffrées. Cela signifie que toute personne utilisant une application de détection de réseau, telle que Wireshark, verra le nom d'utilisateur et le mot de passe vers serveur FTP en clair. Il est préférable d’utiliser Secure FTP (SFTP), qui chiffre FTP via openSSH.

Il est plus courant de configurer un service FTP pour un accès anonyme et d’autoriser uniquement les téléchargements de fichiers. Avec un accès FTP anonyme, au lieu d’un nom d’utilisateur attribué individuellement, un nom d’utilisateur général, tel que **anonymous** ou **ftp**, est utilisé. Il se peut qu'aucun mot de passe ne soit requis ou qu'une demande d'adresse de courriel ne soit demandée (bien que l'adresse de courriel ne soit pas vérifiée).

### Connexions passives et actives

FTP utilise généralement deux ports TCP: le port de données et le port de commande.

* Le **port de commande** est utilisé pour envoyer des commandes et gérer les réponses aux commandes ;
* Le **port de données** est utilisé pour le transport des données de fichier.

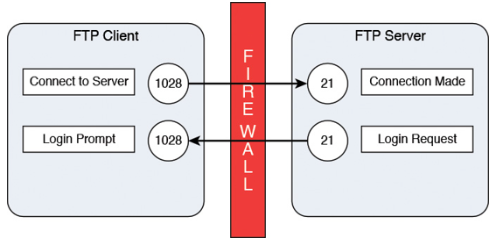
Il est important de savoir que FTP utilise deux modes de fonctionnement … passif et actif.   
Le mode utilisé détermine comment une connexion est établie.   
Le mode passif cause moins de problèmes (comme décrit sous peu), il est donc le plus populaire des deux.

Pour comprendre la différence entre les modes actif et passif, on doit d'abord comprendre une fonctionnalité commune utilisée dans les pares-feux. Les administrateurs veulent souvent autoriser les hôtes internes à établir des connexions avec des hôtes externes, mais empêchent le contraire, sauf dans de rares cas.

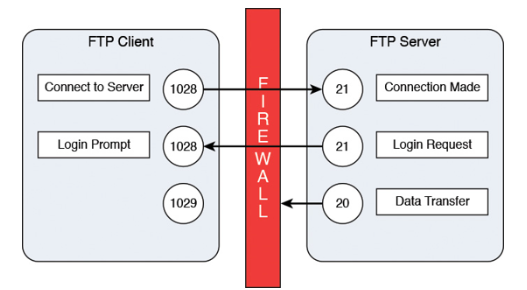
Par exemple, le serveur Web de l’entreprise peut se trouver derrière le pare-feu de l'entreprise. Le pare-feu peut donc être configuré pour autoriser les connexions entrantes pour les ports 80 et 443. Les autres communications initiées sur tout autre port situé à l'extérieur du pare-feu sont bloquées.

Normalement, toute communication initiée depuis le pare-feu, quel que soit le port, est autorisée. Ainsi, si on se situe sur un hôte du pare-feu et on tente de se connecter à un serveur SSH situé en dehors du pare-feu, cette connexion devrait être autorisée par le pare-feu.

Cela fonctionne bien pour presque tous les serveurs, FTP étant une exception. Par défaut, les connexions FTP utilisent une fonctionnalité appelée active. Dans une connexion FTP active, le client FTP utilise un port aléatoire non privilégié (numéro de port supérieur à 1023) pour établir la connexion au port 21 (appelé port de commande) sur le serveur FTP. Le serveur FTP répond initialement au port fourni par le client, en gérant le processus de connexion de l'utilisateur.



Une fois cette connexion initiale établie, des actions supplémentaires sont effectuées. L'utilisateur est invité à entrer un nom d'utilisateur et un mot de passe pour se connecter au serveur FTP. Une fois que l'utilisateur est connecté avec succès, tout transfert de données (téléchargement de fichiers, par exemple) a lieu sur des ports différents. Sur le serveur, le port 20 est utilisé. Sur le client, le numéro de port est supérieur au port de connexion. Ainsi, pour l'exemple de la figure suivante, il s'agit du port 1029.



Toutes les commandes émises par le client (et les réponses du serveur) continuent d'être envoyées via les ports d'origine. Mais le transfert de données utilise les nouveaux ports et est lancé par le serveur. Il s’agit d’un problème, car les pares-feux sont généralement conçus pour bloquer ce type de connexion. On pourrait penser que l’on pourrait simplement autoriser la connexion entrante pour ce port, mais ce port est une cible en mouvement. Le client sélectionne aléatoirement les ports côté client lorsque la connexion est établie pour la première fois.

Le résultat est que le serveur FTP semble se bloquer lorsque le client tente de télécharger ou de téléverser un fichier. Tout le reste à ce point semble bien fonctionner.

Lorsque le mode passif est utilisé, le serveur FTP est invité à ne pas initier la communication de transfert de données et à attendre que le client établisse la connexion. Cela répond aux règles du pare-feu car les données définies par le serveur FTP sont considérées comme une réponse à la demande de connexion du client FTP.

### En résumé … par étapes

En mode actif, le serveur FTP et le client sont actifs pour établir les connexions.  
De plus, le serveur FTP utilise le port 20 comme port de données et le port 21 comme port de commande.

Les étapes de connexion de base en mode actif sont les suivantes ...

* Le client FTP sélectionne et ouvre un port aléatoire non privilégié C (où C est > 1024) pour lui servir de port de commande afin d’envoyer et de recevoir des commandes ;
* Le client FTP sélectionne et ouvre un autre port aléatoire non privilégié D (où D est C + 1) pour lui servir de port de données et commence à écouter les données du serveur FTP ;
* Le client FTP utilise son port de commande (port C) pour informer le serveur FTP via le port de commande du serveur (port 21) qu’il écoute des données sur son port de données (port D) ;
* Le serveur FTP utilise son port de commande (port 21) pour accuser réception de la demande, qui est envoyée au port de commande du client FTP (port C) ;
* Le serveur FTP utilise son port de données (port 20) pour se connecter au port de données du client FTP (port D).

Ainsi, en mode actif, le client FTP établit la connexion de commande, mais le serveur FTP établit la connexion de données.

En mode passif, le serveur FTP est passif et seul le client FTP est actif pour l'établissement des connexions. De plus, le serveur FTP utilise le port 20 comme port de données mais un port aléatoire non privilégié comme port de commande.

Les étapes de connexion de base en mode passif sont les suivantes …

1. Le client FTP sélectionne et ouvre deux ports aléatoires non privilégiés … le port C servant de port de commande et le port D servant de port de données ;
2. Le client FTP utilise son port de commande (port C) pour informer le serveur FTP à partir le port de commande du serveur (port 21) qu’il établit une connexion passive ;
3. Le serveur FTP sélectionne et ouvre un port aléatoire non privilégié SD (où SD est> 1024) en tant que port de données pour envoyer/recevoir les données du client FTP ;
4. Le serveur FTP utilise son port de commande (port 21) pour informer le client FTP via le port de commande du client (port C) qu’il écoute des données sur son port de données (port SD) ;
5. Le client FTP utilise son port de données (port D) pour se connecter au port de données du serveur FTP (port SD).

Les modes FTP passif et actif présentent des avantages et des inconvénients. Généralement, le mode passif est utilisé par les serveurs FTP qui desservent de nombreux systèmes clients FTP WAN. Toutefois, si on utilise ce mode sur un serveur FTP, il est important de configurer correctement le pare-feu du serveur et d’en faire un serveur exclusivement FTP afin d’empêcher les attaques malveillantes.

Pour établir le mode passif, on doit se connecter au serveur FTP, puis émettre la commande passive …

# **ftp localhost**  
**Connected to localhost.**  
**220 (vsFTPd 3.0.3)**  
**Name (localhost:lsavard): anonymous**  
**331 Please specify the password.**  
**Password:**  
**230 Login successful.**  
**Remote system type is UNIX.**  
**Using binary mode to transfer files.**

**ftp> passive**  
**Passive mode on.  
ftp>**

### Commandes FTP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| ! | **cr** | **macdef** | **proxy** | send |
| $ | **delete** | **mdelete** | **sendport** | status |
| account | **debug** | **mdir** | **put** | struct |
| append | **dir** | **mget** | **pwd** | sunique |
| ascii | **disconnect** | **mkdir** | **quit** | tenex |
| bell | **form** | **mls** | **quote** | trace |
| binary | **get** | **mode** | **rcv** | type |
| bye | **glob** | **mput** | **remotehelp** | user |
| case | **hash** | **nmap** | **rename** | verbose |
| cd | **help** | **ntrans** | **reset** | ? |
| cdup | **lcd** | **open** | **rmdir** |  |
| close | **ls** | **prompt** | **runique** |  |
|  |  |  |  |  |

**Attention** …  
Avec FTP, toutes les données (dont les mots de passe) sont transmises en clair.



### TFTP (*Trivial File Transfert Protocol*)

Le protocole TFPT permet de télécharger des fichiers plus rapidement. Cependant il ne garantit pas la qualité du transfert car il utilise avec UDP comme protocole de la couche Transport.  
Puisque le protocole TFTP utilise le protocole UDP et possède donc sa propre correction d’erreur. TFTP propose ses propres messages d’acquittement. Si un message d’acquittement n’est pas parvenu après un certain temps, TFTP renvoie le paquet.

TFTP utilise une simple technique PAR (*Positive Acknowledgement Retransmission*).

**Format des messages TFTP**

|  |  |
| --- | --- |
| Code d’opération | Description |
| 1 | Requête de lecture (RRQ) |
| 2 | Requête d’écriture (WRQ) |
| 3 | Données (DATA) |
| 4 | Acquittement (ACK) |
| 5 | Erreur (ERROR) |
|  |  |

TFTP transmet des blocs de 512 octets.  
Si un bloc n’a pas cette valeur l’hôte distant comprend que c’est le dernier paquet (si le dernier paquet a 512 octets, l’hôte émetteur envoie un paquet vide).



## Accès distant

### SSH

Secure SHell (SSH) est à la fois un programme informatique et un protocole de communication sécurisé.

Le protocole de connexion impose un échange de clés de chiffrement en début de connexion.   
Par la suite, tous les segments TCP sont authentifiés et chiffrés.   
Il devient alors impossible (ou presque) d'utiliser un renifleur (*sniffer*) pour voir ce que fait un utilisateur.

Le protocole SSH a été conçu avec l'objectif de remplacer les différents protocoles non chiffrés comme rlogin, telnet, rcp et rsh.

Habituellement le protocole SSH utilise le port TCP 22.   
Il est particulièrement utilisé pour ouvrir un shell sur un hôte distant.

SSH peut également être utilisé pour transférer des ports TCP d'un hôte vers un autre, créant ainsi un tunnel.   
Cette méthode est couramment utilisée afin de sécuriser une connexion qui ne l'est pas (par exemple le protocole de récupérations de courrier électronique POP3) en la faisant transférer par le biais du tunnel chiffré SSH.

Il est également possible de faire plusieurs sauts entre consoles SSH, c'est-à-dire ouvrir une console sur un serveur, puis, de là, en ouvrir une autre sur un autre serveur.

Les usages de base de SSH sont ...

* L'**accès à distance à la console en ligne commande** (shell), ce qui permet, entre autres, d'effectuer la totalité des opérations courantes et/ou d'administration sur un hôte distant ;
* Le **déport de l'affichage graphique** de l’hôte distant ;
* Le **transfert des fichiers en ligne de commande** ;
* Le **montage ponctuel et/ou automatique d’un répertoire distant**, soit en ligne de commande, soit à l’aide d’une interface graphique.

### Tunnel SSH

Un tunnel, dans le contexte de réseaux informatiques, est une encapsulation de données d'un protocole réseau dans un autre, situé dans la même couche du modèle en couches, ou dans une couche de niveau supérieur.

Par exemple, pour faire passer le protocole IPv6 par le réseau Internet actuel (qui est en grande partie en IPv4) on va créer un tunnel entre deux machines IPv4 ; ce tunnel, pour le protocole IPv6, semblera un simple lien point-à-point   
(un logiciel comme traceroute ne verra donc pas le tunnel).

En sécurité, on crée souvent des tunnels chiffrés, par exemple comme le fait SSH. Les données peuvent alors y circuler sans craindre d'être écoutées. Les tunnels peuvent être utilisés pour créer des réseaux privés virtuels (VPN).

Le tunnel HTTP est un cas particulier qui consiste à faire passer une connexion explicitement interdite par un pare-feu (le protocole SSH par exemple) encapsulé dans un protocole HTTP presque toujours autorisé. C'est particulièrement utile quand le pare-feu ne bloque pas simplement les ports associés (22 pour SSH par exemple), mais analyse aussi les protocoles utilisés.

### Telnet

L'émulation de terminal (Telnet) permet de communiquer à distance avec un autre ordinateur. Cela permet d'ouvrir une session sur un hôte Internet et d'exécuter des commandes.  
Un client Telnet est désigné comme un hôte local et un serveur Telnet, qui exécute un logiciel spécial appelé démon, est désigné comme un hôte distant. Le traitement est effectué par l’hôte serveur et les résultats sont livrés au client.   
Le protocole Telnet permet d’émuler une connexion de terminal à un hôte distant (telnetd ou telnet daemon).

L’hôte client se comporte alors comme un terminal. Le protocole TCP est utilisé afin de transmettre les données saisies au clavier du client et d’afficher les informations provenant du serveur.  
Les caractères saisis sont reçus directement par le serveur qui les passe instantanément à son système d’exploitation. Le serveur envoie par la suite les informations qui seront affichées sur l’écran du client. Pour l’usager, le tout lui apparaît comme s’il travaillait localement.   
Telnet nécessite un nom d’usager et un mot de passe.   
La session TCP est fermée seulement à la fin de session telnet.



# Protocoles de configuration des hôtes

## Protocole DHCP

Dynamic Host Configuration Protocol (DHCP) désigne le protocole réseau dont le rôle est d'assurer la configuration automatique des paramètres TCP-IP d'un hôte, notamment en lui assignant automatiquement une adresse IP et un masque de sous-réseau. DHCP peut aussi configurer l'adresse de la passerelle par défaut, des serveurs de noms DNS et de nombreux autres paramètres.

La conception initiale d'IP supposait la configuration manuelle de chaque hôte connecté au réseau avec les paramètres TCP-IP adéquats : c'est l'adressage statique. Sur des réseaux de grandes dimensions ou étendues, où des modifications interviennent souvent, l'adressage statique engendre une lourde charge de maintenance et des risques d'erreurs. En outre, les adresses assignées ne peuvent être utilisées même si l’hôte qui la détient n'est pas en service : un cas typique où ceci pose problème est celui des fournisseurs d'accès à Internet (FAI ou ISP en anglais), qui ont en général plus de clients que d'adresses IP à leur disposition, mais dont tous les clients ne sont jamais connectés en même temps.

DHCP apporte une solution à ces deux inconvénients …

* Seuls les hôtes en service utilisent une adresse de l'espace d'adressage ;
* Toute modification des paramètres (adresse de la passerelle, des serveurs de noms) est répercutée sur les stations lors du redémarrage.

La modification de ces paramètres est centralisée sur les serveurs DHCP.

### Possibilités offertes

On compte parmi les avantages de l’utilisation de DHCPv4 les suivants …

* **Vérification rapide des adresses IP et autres paramètres** de configuration TCP/IP ;
* **Configuration de l’étendue d’adresses IP se fait de façon centralisée** ;
* **Non nécessité de modifier manuellement chaque client** ;
* **Non attribution de la même adresse IP à deux clients** ;
* **Attribution d’adresses fixes tout en permettant l’attribution dynamique des autres paramètres IP** ;
* **Risques d’erreurs d’écriture et de saisie sont diminués** ;
* **Reconfiguration d’un hôte plus facile si ce dernier est déplacé** d’un sous-réseau à un autre ;
* **Meilleure répartition des adresse IP** (lorsqu’il y a plus d’hôtes que d’adresses IP disponibles).

**Limitations DHCPv4**

Toutefois, le protocole possède certaines limites …

* Un serveur DHCP ne dispose d’**aucun moyen de communiquer avec un autre serveur DHCP** ;
* Il faut que les **étendues d’adresses soient bien définies et ne se recoupent pas** ;
* Les routeurs ne laissent pas passer les **messages de diffusion** (sauf s’ils sont configurés pour laisser passer les messages BOOTP) ;  
  Il faut donc installer l’**agent de relais DHCP** ;
* Une **erreur de configuration DHCP peut entraîner des résultats catastrophiques**.

### Fonctionnement

DHCP fonctionne sur le modèle client-serveur...

* un serveur, qui détient la politique d'attribution des configurations IP, envoie une configuration donnée pour une durée donnée à un client donné (typiquement, un hôte qui vient de démarrer ou d’tablir une connexion avec le réseau local).

Le serveur va servir de base pour toutes les requêtes DHCP (il les reçoit et y répond), aussi doit-il avoir une configuration IP fixe. Dans un réseau, on peut donc n'avoir qu'une seule machine avec adresse IP fixe: le serveur DHCP.

Le protocole DHCP s'appuie entièrement sur BOOTP: il en reprend le mécanisme de base (ordre des requêtes, mais aussi le format des messages). DHCP est une extension de BOOTP.

Quand un hôte vient de démarrer ou d’établir une connexion au réseau, il ne possède pas de configuration réseau (même pas de configuration par défaut), et pourtant, il doit arriver à émettre un message sur le réseau pour qu'on lui donne une vraie configuration.

La technique utilisée est le diffusion (*broadcast*) afin de trouver et dialoguer avec un serveur DHCP. L’hôte va simplement émettre un paquet spécial, dit de diffusion, sur l'adresse IP 255.255.255.255 et sur le réseau local. Ce paquet particulier va être reçu par toutes les hôtes connectées au réseau (particularité de diffusion).

Lorsqu'un serveur DHCP reçoit ce paquet, il répond par un autre paquet de diffusion contenant toutes les informations requises pour la configuration. Si le client accepte la configuration, il renvoie un paquet pour informer le serveur qu'il garde les paramètres, sinon, il fait une nouvelle demande.

Les choses se passent de la même façon si le client possède déjà une adresse IP (négociation et validation de la configuration), sauf que le dialogue ne s'établit plus avec de diffusion.

**Bail**

Pour des raisons d'optimisation des ressources réseau, les adresses IP sont délivrées pour une durée limitée. C'est ce qu'on appelle un bail (*lease*). Un client qui voit son bail arriver à terme peut demander au serveur un renouvellement du bail.

De même, lorsque le serveur verra un bail arrivé à terme, il émettra un paquet pour demander au client s'il veut prolonger son bail. Si le serveur ne reçoit pas de réponse valide, il rend disponible l'adresse IP.

C'est toute la subtilité du DHCP: on peut optimiser l'attribution des adresses IP en jouant sur la durée des baux.  
Le problème est là : si toutes les adresses sont allouées et si aucune n'est libérée au bout d'un certain temps, plus aucune requête ne pourra être satisfaite.

Sur un réseau où beaucoup d'hôtes se connectent et se déconnectent souvent (réseau d'école ou de locaux commerciaux par exemple), il est intéressant de proposer des baux de courte durée. A l'inverse, sur un réseau constitué en majorité d’hôtes fixes, très peu souvent redémarrées, des baux de longues durées suffisent.

**Remarque** …  
Le protocole NAT a permis de régler une bonne partie du manque d’adresses pour un réseau local.

**Dynamique ou non ?**

Un serveur DHCP est censé fournir des adresses dynamiques (un même hôte peut recevoir successivement 2 adresses différentes), mais il peut fournir une adresse IP fixe à un client bien particulier. Ceci ne doit être utilisé que de manière modérée, sinon, le serveur DHCP ne sert à peu près plus à rien, mais cela peut se révéler utile pour fournir l'adresse IP, par exemple, à un serveur TFTP qui va servir pour le démarrage à distance des hôtes.

### Requêtes et les messages DHCP

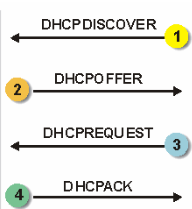
On pourrait croire qu'un seul aller-retour peut suffire à la bonne marche du protocole. En fait, il existe plusieurs messages DHCP qui permettent de compléter une configuration ou de la renouveler...

Ces messages sont susceptibles d'être émis soit par le client pour le ou les serveurs, soit par le serveur vers un client.  
La valeur entre parenthèses est utilisée pour identifier ces requêtes dans les messages DHCP.

* **DHCPDISCOVER** (1)  
  Pour localiser les serveurs DHCP disponibles ;
* **DHCPOFFER** (2)  
  Réponse du serveur à un paquet DHCPDISCOVER, qui contient les premiers paramètres ;
* **DHCPREQUEST** (3)  
  Requête diverse du client pour par exemple prolonger son bail ;
* **DHCPDECLINE** (4)  
  Annonce du client au serveur que l'adresse est déjà utilisée ;
* **DHCPACK** (5)  
  Réponse du serveur qui contient des paramètres et l'adresse IP du client ;
* **DHCPNAK** (6)  
  Réponse du serveur pour signaler au client que son bail est échu ou si le client annonce une mauvaise configuration réseau ;
* **DHCPRELEASE** (7)  
  Libération de l’adresse IP par le client ;
* **DHCPINFORM** (8)  
  Demande du client pour des paramètres locaux (il a déjà son adresse IP).

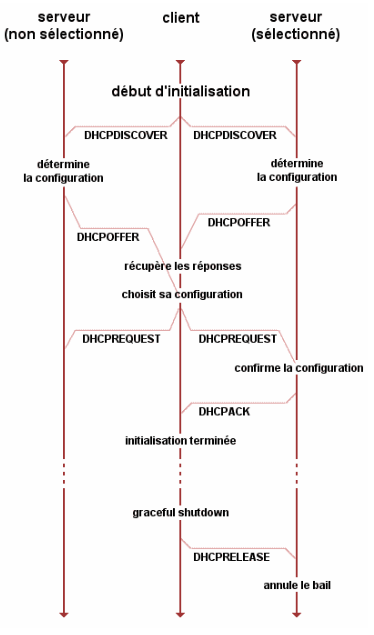
La valeur entre parenthèses est utilisée afin d’identifier ces requêtes dans les messages DHCP.

* La première requête émise par le client est un message **DHCPDISCOVER** ;
* Le serveur répond par un **DHCPOFFER**, en particulier pour soumettre une adresse IP au client ;
* Le client établit sa configuration, demande éventuellement d'autres paramètres, puis fait un **DHCPREQUEST** pour valider son adresse IP ;
* Le serveur répond simplement par un **DHCPACK** avec l'adresse IPv4 pour confirmation de l'attribution.

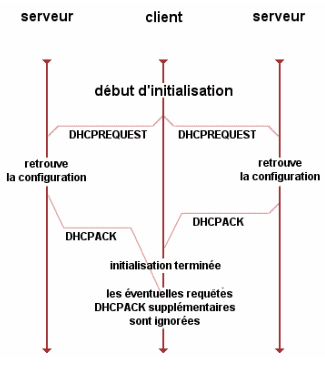


Normalement, cela est suffisant pour qu'un client obtienne une configuration réseau efficace, mais cela peut être plus ou moins long selon que le client accepte ou non l'adresse IP ou demande des infos complémentaires.

* Pour demander une nouvelle adresse, le chronogramme-type est le suivant …



* Pour renouveler une adresse, le fonctionnement est le suivant (les 2 serveurs connaissent le client):Les messages DHCP



Les messages DHCP sont transmis à l’aide du protocole UDP.  
Bien que peu fiable, ce protocole suffit au transport des paquets simples sur réseau local, et surtout il est très léger, donc intéressant pour les petits systèmes (du genre le micro-bout de programme qui fait la requête DHCP lorsque le PC se met en route ou se connecte).

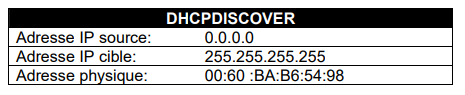
De facto, DHCP fonctionne aussi en mode non connecté. Le client n'utilise que le port 68 afin d’envoyer et de recevoir ses messages de la même façon, le serveur envoie et reçoit ses messages sur un seul port, le port 67.

### Port UDP

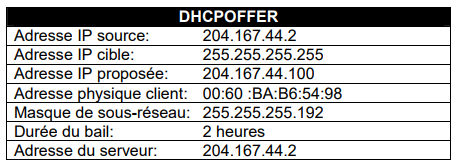
Le service DHCP utilise UDP comme protocole de transport.   
Les ports 67 et 68 (qui sont aussi les ports BOOTP) sont réservés à cet effet.

**Exemple de paquet DHCPv4**

Lors du démarrage, un hôte configuré pour obtenir automatiquement sa configuration IPv4 envoie un DHCPDISCOVER. Plusieurs essais peuvent être fait (1, 2, 4, 8, 16 secondes).



Un ou plusieurs serveurs DHCP peuvent répondre en offrant une configuration IP.



### Renouvellement du bail

Les adresses IPv4 dynamiques sont octroyées pour une durée limitée, qui est transmise au client dans l'accusé de réception qui clôture la transaction DHCP.

La valeur T1 qui l'accompagne détermine la durée après laquelle le client commence à demander périodiquement le renouvellement de son bail auprès du serveur qui lui a accordé son adresse (couramment la moitié de la durée du bail). Cette fois la transaction est effectuée par transmission IP classique, d'adresse à adresse.

Si lorsque le délai fixé par la deuxième valeur, T2, est écoulé, le bail n'a pas pu être renouvelé (par exemple si le serveur DHCP d'origine est hors service), le client demande une nouvelle allocation d'adresse par diffusion.

Si au terme du bail le client n'a pu ni en obtenir le renouvellement, ni obtenir une nouvelle allocation, l'adresse est désactivée et il perd la faculté d'utiliser le réseau de façon normale.

**Processus de renouvellement de bail**

* À **50% de la durée du bail** le client cherche à le renouveler ;
* À **87,5% de la durés du bail** le client fait un nouveau DHCPDISCOVER.
* À la **fin du bail** l’hôte cesse d’utiliser l’adresse IP qui lui a été attribuée.

Il faut prévoir une durée de bail réaliste …

* Le renouvellement occasionne du trafic supplémentaire ;
* Un réseau informatique est composé de plus en plus d’hôtes mobiles.

Cependant le trafic engendré par le renouvellement des baux n’est pas si important et si l’on dispose d’un nombre limité d’adresses IP il est préférable de choisir des durées plus courtes.

|  |  |
| --- | --- |
| Temps | Étapes |
| 50% ou 4/8 | Le client demande de renouvellement auprès de son serveur DHCPv4 |
| 82,5% ou 7/8 | Le client demande un nouveau bail par diffusion générale |
| 100% ou 8/8 | Le client abandonne sa configuration actuelle |

* Alors …  
  Le client va s’autoconfigurer dans le réseau 169.254.0.0/16 (zeroconf ou APIPA)
* Par la suite …  
  Le client DHCP va envoyer périodiquement des requêtes DHCPDISCOVER (1)

### Adresse privée automatique (APIPA ou zeroconf)

Si les tentatives pour rejoindre un serveur DHCPv4 ont échouées, un client DHCP va s’attribuer automatiquement une adresse IP dans l’**étendue privée 169.254.0.1 à 169.254.255.254** et le **masque de sous-réseau 255.255.0.0**.  
Le client DHCP s’assure à l’aide d’une requête ICMP que l’adresse choisie n’est pas déjà utilisée.

Ce client va par la suite faire périodiquement un **DHCPDISCOVER** afin de trouver un serveur pouvant lui fournir une configuration IP

Les clients qui n’utilisent pas cette fonctionnalité auront comme adresse IPv4 0.0.0.0/32.

### Réservation d’adresse IPv4

Il est possible d’effectuer des réservations d’adresses en limitant la possibilité d’octroi de cette adresse au client possédant une adresse physique ou un identifiant client donné.

Ceci peut s’avérer utile pour des hôtes dont l’adresse doit rester fixe mais dont on veut contrôler de manière centrale et automatique les autres paramètres IP.

Ce mécanisme est assuré par l’option 61 (voir RFC 21313).

Le serveur DHCP choisi élabore un datagramme d'accusé de réception qui assigne au client l'adresse IP et son masque de sous-réseau, la durée du bail de cette adresse, deux valeurs T1 et T2 qui déterminent le comportement du client en fin de bail, et éventuellement d'autres paramètres ...

* **adresse IP de la passerelle par défaut** ;
* **adresses IP des serveurs DNS** ;
* **…**

La liste des options que le serveur DHCP peut accepter est consultable dans la RFC 2132 : Options DHCP et Extensions fournisseur BOOTP.

### Client et serveur sur des segments différents

Lorsque le serveur DHCP et le client ne figurent pas sur le même segment IP, les diffusions émises par ce dernier ne parviennent pas au serveur parce que les routeurs, ne transmettent pas les diffusions générales (la RFC 1452 décrit toutefois la possibilité pour un aiguilleur de laisser passer les diffusions DHCP).

Dans ce cas on utilise un agent de relais DHCP.

Cet hôte particulier est configuré avec une adresse IPv4 statique, et connaît l'adresse d'un serveur DHCP auquel il transmet les requêtes DHCP qui lui parviennent sur le port 68 (écouté par le programme agent de relais).   
Il diffuse sur son segment (qui est aussi celui du client) les réponses qu'il reçoit du serveur DHCP.



## Protocole DNS

## Concepts de base

DNS (Domain Name Service) est un protocole conçu pour fournir une résolution de nom en adresse IP.   
Il fait partie de la suite de protocoles TCP/IP standard et de l'un des protocoles pouvant fournir cette fonctionnalité ; les autres étant NIS et LDAP.

Ce qui distingue le DNS des autres protocoles similaires est que son unique objectif est la résolution de noms; NIS et LDAP fournissent d'autres opérations de résolution.   
DNS est également la solution de résolution de noms standard de facto pour la majorité des systèmes connectés à Internet.

**Fichier hosts.txt**

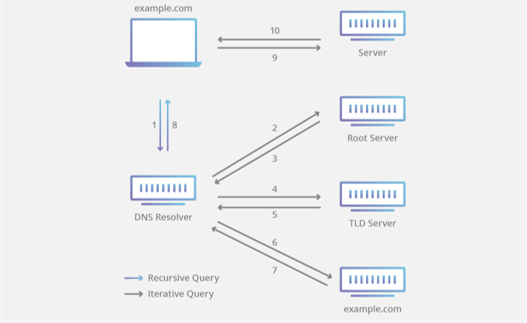
Avant le DNS, la résolution d'un nom sur Internet devait se faire grâce à un fichier texte appelé HOSTS.TXT (RFC 60813) et copié sur chaque ordinateur du réseau par transfert de fichier.

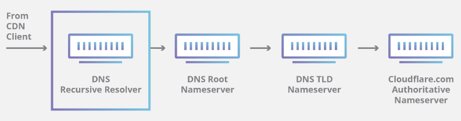
En 1982, ce système centralisé montre ses limites et plusieurs propositions de remplacement voient le jour, parmi lesquelles le Domain Name System afin de gérer la croissance de l'internet en déléguant la gestion des noms de domaine à des serveurs de noms distribués.

En 1987, le fichier hosts.txt contenait 5 500 entrées, tandis que 20 000 hôtes étaient définis dans le DNS.

### Processus simplifié de résolution de noms

Voici un processus de résolution de noms générique …





### Nom de domaine

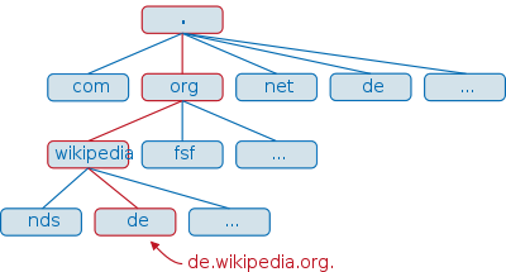
On représente un nom de domaine en indiquant les domaines successifs séparés par un point, les noms de domaines supérieurs se trouvant à droite.

Par exemple, profsavard.info. …

* Le domaine info. est un TLD, sous-domaine de la racine ;
* Le domaine profsavard.info. est un sous-domaine de .info.

Cette délégation est accomplie en indiquant la liste des serveurs DNS associée au sous-domaine dans le domaine de niveau supérieur.

Les noms de domaines sont donc résolus en parcourant la hiérarchie depuis le sommet et en suivant les délégations successives, c'est-à-dire en parcourant le nom de domaine de droite à gauche.



### Résolveur

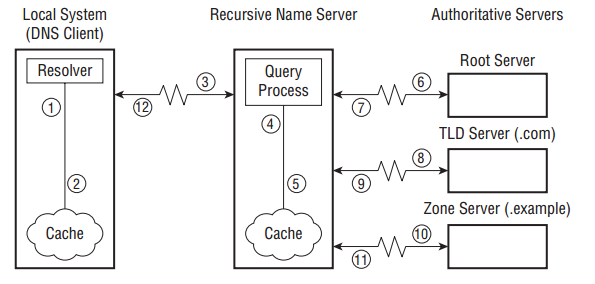
Le résolveur est un acteur important dans la résolution de noms.   
Le résolveur est un programme ou une routine de bibliothèque qui forme une requête   
(par exemple … Quelle est l'adresse IP de [www.profsavard.info](http://www.profsavard.info/) ?) et qui doit obtenir la réponse à la requête.  
Un résolveur peut également faire référence à un client DNS.

Lorsqu'un résolveur crée sa requête, il vérifie d'abord son propre cache DNS, s'il est disponible.   
Si la réponse à cette requête est récente et que le cache du résolveur contient la réponse, le processus de requête s’arrête.

Les différentes résolutions de noms qui existent dans les différents caches DNS sont souvent régies par un paramètre de durée de vie (TTL).   
Une fois ce délai écoulé, la résolution de noms est supprimée du cache.

Si le cache du résolveur ne contient pas la réponse, le résolveur envoie la requête à un serveur de noms interne ou externe (généralement un FAI). Cette requête est une requête UDP unique et le serveur de noms est généralement appelé serveur de noms récursif. Si les résolutions de noms précédentes ont été stockées sur ce serveur (mis en cache) ou sur un serveur de mise en cache local, le serveur de noms récursif répond à la requête en utilisant une seule réponse UDP contenant l'adresse IP appropriée. Si la résolution de nom n'a pas été mise en cache, le serveur de noms récursif démarrera son propre processus de requête.

Par exemple, lorsqu’un résolveur tente de déterminer l'adresse IP du site Web www.example.com.   
L'exemple de processus de résolution de noms est illustré à la figure suivante.



Chaque étape du processus de résolution de nom dans cet exemple de requête est …

1. Le résolveur recherche l’adresse IP de www.example.com dans le cache de son propre serveur ;
2. Le résolveur est informé que les données de résolution de nom ne se trouvent pas dans le cache local ;
3. Le résolveur interroge le serveur de noms récursif de son FAI, qui lance un processus de requête ;
4. Le processus de requête du serveur de noms récursif vérifie dans son cache local l'adresse IP de www.example.com ;
5. Le serveur de noms récursif est informé que les données de résolution de noms ne se trouvent pas dans le cache local ;
6. Le serveur de noms récursif interroge le serveur racine faisant autorité pour connaître l'emplacement du serveur TLD .com ;
7. Le serveur racine répond avec les informations faisant autorité indiquant l'emplacement du serveur TLD.com ;
8. Le serveur de noms récursif interroge le serveur TLD .com faisant autorité afin de connaître l'emplacement du serveur de zone .exemple ;
9. Le serveur TLD .com répond avec les informations faisant autorité indiquant l'emplacement du serveur de la zone .exemple ;
10. Le serveur de noms récursif interroge le serveur de la zone .example faisant autorité pour connaître l'adresse IP de l'hôte www ;
11. Le serveur de la zone .exemple répond avec les informations faisant autorité indiquant l'adresse IP de l'hôte www ;
12. Le serveur de noms récursif transmet l'adresse IP de www.example.com au résolveur.

Ce processus de résolution de nom se déroulerait beaucoup plus lentement si la base de données DNS n'était pas distribuée. En outre, la mise en cache des résolutions de nom récentes est bénéfique pour les performances.

Du côté client de la résolution du nom, différentes configurations de serveur de transfert ou de mise en cache sont possibles.   
La configuration côté client DNS particulière sera dictée par les applications et les besoins en données du/des système (s) ainsi que par leurs ressources.

### Type de requêtes DNS

Il existe t types de requêtes DNS …

* **Requête récursive**   
  Dans une requête récursive, un client DNS nécessite qu'un serveur DNS réponde au client avec…   
  soit l'enregistrement de ressource demandé soit un message d'erreur si le résolveur ne trouve pas l'enregistrement ;   
  soit un message d'erreur si le résolveur ne trouve pas l'enregistrement.  
  Cette requête est habituellement faite par un clients DNS.
* **Requête itérative**   
  Dans cette situation, le client DNS permettra à un serveur DNS de renvoyer la meilleure réponse possible.  
  Si le serveur DNS interrogé n'a pas de correspondance pour le nom de la requête, il renverra une référence à un serveur DNS faisant autorité pour un niveau inférieur de l'espace de noms de domaine.   
  Le client DNS fera alors une requête à l'adresse de référence.   
  Ce processus se poursuit avec des serveurs DNS supplémentaires dans la chaîne de requête jusqu'à ce qu'une erreur ou un délai d'attente se produise.   
  Cette requête est habituellement faite par un ou des serveurs DNS,
* **Requête non récursive**   
  Ce type de requête se produit généralement lorsqu'un client résolveur DNS interroge …   
  un serveur DNS pour un enregistrement auquel il a accès ;   
  soit parce qu'il fait autorité pour l'enregistrement ;   
  soit que l'enregistrement existe à l'intérieur de son cache.   
  En règle générale, un serveur DNS mettra en cache les enregistrements DNS afin d’empêcher la consommation de bande passante supplémentaire et la charge sur les serveurs en amont.

**Serveurs DNS racine**

Les serveurs racine sont gérés par douze organisations différentes …

* deux sont européennes ;
* une japonaise   
  et
* neuf autres sont américaines.

Sept de ces serveurs sont en réalité distribués dans le monde et neuf disposent d'une adresse IPv6.   
Plus de 200 serveurs répartis dans 50 pays du monde assurent ce service.

Il existe 13 autorités de nom appelées de a.root.servers.net à m.root-servers.net.   
Le serveur k reçoit par exemple de l'ordre de 70 000 à 100 000 requêtes par seconde en avril 2019.

Le DNS ne fournit pas de mécanisme pour découvrir la liste des serveurs racine, chacun des serveurs doit donc connaître cette liste au démarrage grâce à un encodage explicite.

La mise à jour de cette liste est peu fréquente de façon que les serveurs anciens continuent à fonctionner.

### Termes essentiels

Les termes suivants sont associés à DNS …

* **Hôte**  
  Un hôte est généralement un ordinateur connecté à un réseau.   
  Une autre façon de voir le terme est qu'un hôte est un périphérique ou nœud capable de communiquer sur un réseau ;
* **Nom de domaine**  
  **L**es hôtes Internet établissent mutuellement un contact à l'aide de numéros d'adresse IP.   
  Les humains ont du mal à se souvenir de ces numéros, c'est pourquoi un nom unique est souvent attribué à un hôte.   
  Lorsque ce nom est enregistré sur un serveur DNS autorisé, le nom est considéré comme un nom de domaine ;
* **Domaine de premier niveau**  
  **L**es noms de domaine sont structurés en arborescence, un peu comme les fichiers sont organisés en structure de système de fichiers virtuel.   
  Le niveau supérieur de la structure DNS est simplement appelé **point** et est symbolisé par le **caractère** **.**.   
  Les domaines directement sous la racine (.) sont les domaines de premier niveau.   
  Les domaines de premier niveau d'origine étaient **.com**, **.org**, **.net**, **.int**, **.edu**, .**gov** et **.mil**.   
  Beaucoup d'autres ont été ajoutés au fil des années ;
* **FQDN** (***Full Qualified Domain Name***)  
  Un nom de domaine pleinement qualifié est le nom de domaine d'un hôte commençant au sommet de la structure DNS.   
  Par exemple, le nom www.profsavard.info. serait un FQDN.   
  Il est important de ne pas oublier le caractère . à la fin du nom de domaine complet.   
  C'est le domaine situé au-dessus des domaines de premier niveau.   
  Ce caractère est souvent omis lorsque les utilisateurs habituels fournissent un nom de domaine car le «.» Est supposé être le dernier caractère d'un nom de domaine complet dans la plupart des cas.   
  Toutefois, il est important de s’habituer à toujours inclure le caractère .   
  Si on souhaite administrer des serveurs DNS, comme cela est requis dans certains fichiers de configuration du serveur DNS ;
* **Sous-domaine**  
  Un sous-domaine est un domaine qui est un composant d'un domaine plus grand. Par exemple, supposons que l’on souhaite avoir trois domaines pour son organisation afin d’organiser de manière fonctionnelle les hôtes. On peut appeler ces domaines ventes, ingénierie et support. Si le domaine de l’entreprise est profsavard.info., ces trois sous-domaines s’appelleraient ventes.profsavard.info., ingénierie.profsavard.info. et   
  support.profsavard.info ;
* **Serveur de noms**  
  Un serveur de noms est un système qui répond aux demandes du client DNS.   
  Les serveurs de noms fournissent la traduction d'adresse IP en noms de domaine (et parfois l'inverse: traduction de nom de domaine en adresse IP).   
  **Remarque** …  
  Un serveur de noms possède une copie de ces informations stockées localement (appelée fichier de zone) ou stocke en mémoire les informations obtenues temporairement par d'autres serveurs de noms, ou transmet la requête à un autre serveur (ou serveurs) disposant de ces informations ;
* **Serveur de noms faisant autorité**  
  **U**n serveur de noms faisant autorité renvoie des résultats en fonction des informations stockées localement sur le système (les enregistrements originaux) ;
* **Fichier de zone**  
  **N**om du fichier utilisé pour stocker les informations de conversion d'adresse IP en nom de domaine (également appelées enregistrements DNS).   
  Ce fichier contient également des informations permettant de définir le domaine lui-même ;
* **Enregistrement**  
  Dans le fichier de zone, un enregistrement est une entrée qui définit un bloc d’informations unique pour la zone, telles que les données qui traduiraient une adresse IP en nom de domaine ;
* **Mise en cache du serveur de noms**  
  Un serveur de noms en mise en cache est un serveur qui renvoie des résultats en fonction d'informations obtenues d'un autre serveur de noms, tel qu'un serveur de noms faisant autorité.   
  Le principal avantage d'un serveur de noms en cache est qu'il peut accélérer la résolution d'adresse IP en noms de domaine, car il peut mettre en cache les résultats et répondre aux futures demandes en utilisant les informations contenues dans ce cache ;
* **TTL** (*Time to Live*)  
  Les données stockées dans un serveur de noms en cache ne sont généralement pas stockées de manière permanente.   
  Le serveur de noms qui fournit les données fournit également au serveur de noms de mise en cache une durée de vie (TTL). Le serveur de noms de mise en cache stocke les informations en mémoire jusqu'à la fin de cette période TTL. Cette période est généralement de 24 heures, mais cela peut varier en fonction du nombre de mises à jour des enregistrements du serveur de noms faisant autorité ;
* **Redirecteur DNS**  
  **Ce** serveur DNS est conçu afin de prendre des requêtes DNS d'un réseau interne et les envoyer à un serveur DNS externe ;
* **Recherche directe** (*Forward lookup*)  
  Processus de traduction d'une adresse IP en un nom de domaine.   
  La plupart des serveurs DNS fournissent cette fonctionnalité ;
* **Recherche inversée** (*Reverse lookup)*  
  *P*rocessus de traduction d'un nom de domaine en une adresse IP.   
  Alors que de nombreux serveurs DNS fournisse cette fonctionnalité, elles sont toutefois moins fréquentes que les recherches directes ;
* **BIND** — Berkeley Internet Name Domain  
  Logiciel DNS le plus largement utilisé sur Internet.   
  La version actuelle de BIND est appelée bind9 ;
* **dnsmasq**  
  **S**erveur pouvant être utilisé comme redirecteur DNS.   
  Il peut également être utilisé comme serveur DHCP (*Dynamic Host Configuration Prot*ocol).   
  Conçu pour les petits réseaux, il est connu pour sa facilité de configuration et ses faibles frais généraux en ressources système ;

### Fonctionnement de la résolution de nom

L'exemple suivant est volontairement simpliste afin de donner une idée du fonctionnement de la résolution de noms.   
De nombreux facteurs peuvent modifier la nature exacte du fonctionnement de ce processus, notamment la configuration de chaque serveur DNS décrit dans l'exemple.

Pour cet exemple, on considère une situation dans laquelle on utilise un fureteur (*browser*) Web pour surfer sur le domaine [www.profsavard.info](http://www.profsavard.info).   
Pour déterminer l'adresse IP de ce nom de domaine, le système doit d'abord déterminer quels serveurs DNS il peut interroger.

Pour l’**environnement Windows** …  
Les serveurs sont défénis dans les propriétés TCP/IP de l’interface réseau.

Pour l’environnement Linux …  
Le fichier /etc/resolv.conf est consulté pour ces informations …  
# **cat /etc/resolv.conf**  
**search profsavard.info**  
**nameserver 192.168.1.1**

**Remarque** …  
Dans la plupart des cas, il est préférable d’avoir au moins deux paramètres de serveur de noms.   
Si le premier serveur de nom est indisponible, le second peut répondre aux requêtes.   
Cependant, on constatera que certaines petites entreprises ne fournissent qu'un seul serveur de noms pour leurs propres systèmes.

De plus, si on utilise une machine virtuelle, le gestionnaire de machines virtuelles sert généralement de serveur de noms unique pour la machine virtuelle. Il ne fait donc pas être surpris si on se retrouve avec un seul paramètre de serveur d.

La requête est envoyée au serveur de noms avec l'adresse IP 192.168.1.1, qui est le serveur DNS de sa propre organisation. On peut le déterminer car l'adresse IP est une adresse IP privée (non routable sur Internet).   
Ce serveur DNS peut avoir mis en cache les résultats d’une requête précédente pour www.profsavard.info, mais on suppose que le serveur DNS local ne dispose pas de ces informations.   
Dans ce cas, le serveur DNS doit transmettre la demande à un autre serveur DNS.

Bien qu'il soit possible de configurer un serveur DNS pour transmettre des demandes à d'autres serveurs DNS spécifiques, la requête est généralement transmise aux serveurs DNS situés en haut de la structure de domaine DNS. Ce sont les serveurs racine. Il existe 13 serveurs racine.

La sortie suivante affiche ces serveurs tels qu'ils sont décrits dans les fichiers de zone BIND.

**;; ANSWER SECTION:**  
**.                      518400  IN      NS      a.root-servers.net.**  
**.                      518400  IN      NS      b.root-servers.net.**  
**.                      518400  IN      NS      c.root-servers.net.**  
**.                      518400  IN      NS      d.root-servers.net.**  
**.                      518400  IN      NS      e.root-servers.net.**  
**.                      518400  IN      NS      f.root-servers.net.**  
**.                      518400  IN      NS      g.root-servers.net.**  
**.                      518400  IN      NS      h.root-servers.net.**  
**.                      518400  IN      NS      i.root-servers.net.**  
**.                      518400  IN      NS      j.root-servers.net.**  
**.                      518400  IN      NS      k.root-servers.net.**  
**.                      518400  IN      NS      l.root-servers.net.**  
**.                      518400  IN      NS      m.root-servers.net.**  
**;; ADDITIONAL SECTION:**  
**a.root-servers.net.    3600000 IN      A       198.41.0.4**  
**a.root-servers.net.    3600000 IN      AAAA    2001:503:ba3e::2:30**  
**b.root-servers.net.    3600000 IN      A       192.228.79.201**  
**c.root-servers.net.    3600000 IN      A       192.33.4.12**  
**d.root-servers.net.    3600000 IN      A       199.7.91.13**  
**d.root-servers.net.    3600000 IN      AAAA    2001:500:2d::d**  
**e.root-servers.net.    3600000 IN      A       192.203.230.10**  
**f.root-servers.net.    3600000 IN      A       192.5.5.241**  
**f.root-servers.net.    3600000 IN      AAAA    2001:500:2f::f**  
**g.root-servers.net.    3600000 IN      A       192.112.36.4**  
**h.root-servers.net.    3600000 IN      A       128.63.2.53**  
**h.root-servers.net.    3600000 IN      AAAA    2001:500:1::803f:235**  
**i.root-servers.net.    3600000 IN      A       192.36.148.17**  
**i.root-servers.net.    3600000 IN      AAAA    2001:7fe::53**  
**j.root-servers.net.    3600000 IN      A       192.58.128.30**  
**j.root-servers.net.    3600000 IN      AAAA    2001:503:c27::2:30**  
**k.root-servers.net.    3600000 IN      A       193.0.14.129**  
**k.root-servers.net.    3600000 IN      AAAA    2001:7fd::1**  
**l.root-servers.net.    3600000 IN      A       199.7.83.42**  
**l.root-servers.net.    3600000 IN      AAAA    2001:500:3::42**  
**m.root-servers.net.    3600000 IN      A       202.12.27.33**  
**m.root-servers.net.    3600000 IN      AAAA    2001:dc3::35**

**Remarque** …  
Il existe actuellement plus de 13 serveurs racine car de nombreux serveurs partagent un nom d'hôte et une adresse IP via un processus appelé mise en miroir.

Ces serveurs racine connaissent les serveurs DNS des domaines de premier niveau (.com, .edu, ...).   
Ainsi, bien qu’ils ne connaissent pas le domaine profsavard.info., ils peuvent diriger la requête vers les serveurs DNS responsables du domaine .info.

Les serveurs DNS .com ne sont pas non plus conscients du domaine profsavard.info., mais ils savent quels serveurs DNS sont responsables du domaine profsavard.info.. \*

La requête est transmise au serveur DNS responsable du domaine profsavard.info., qui retourne l'adresse IP du domaine [www.pro](http://www.onecoursesource.com)fsavard.info.

\* En réalité, ce n’est pas une affirmation tout à fait exacte.

Bien qu'il enfreigne les règles DNS, il est courant qu'un domaine Web tel que [www.pr](http://www.onecoursesource.com)ofsavard.info. soit un alias du domaine profsavard.info.. C'est pourquoi on peut souvent omettre la partie www.   
D'une adresse URL (*Uniform Resource Locator*) lors de la saisie d'une adresse dans un fureteur Web.   
Par conséquent, les serveurs DNS .com peuvent bien avoir un enregistrement pour le domaine [www.prof](http://www.onecoursesource.com)avard.info. dans leurs fichiers de zone.



Gestion de périphériques réseau

## SNMP (Simple Network Management Protocol)

Le protocole SNMP (*Simple Network Management Protocol*) a été défini afin de servir de standard aux échanges d’informations concernant l’état de santé et la configuration des périphériques d’un réseau.   
C’est un protocole, mais comme d’autres protocoles (par exemple DNS) il repose aussi sur un modèle.

Il existe plusieurs versions de SNMP …

* **SNMPv1** est simple mais a des lacunes en sécurité – il ne définit aucun mécanisme d’authentification ni de chiffrement ;
* **SNMPv2** vise à corriger ce problème, mais ceux qui ont participé au développement de cette version ne se sont jamais vraiment entendus; il existe donc plusieurs sous-versions de SNMPv2 ;
* **SNMPv3** est le standard officiel, le plus récent. Mais en réalité les 3 versions sont utilisées. Dans ce cours, nous allons voir deux versions du protocole : la 2c et la 3.

Le modèle SNMP définit deux types d’entités :

* Les **entités gérées** (hôtes, commutateurs, routeurs) ;
* Les **entités de gestion** (les hôtes qui reçoivent les informations provenant des entités gérées).

Avec SNMP, on ne parle pas de client ou de serveur, mais plutôt d’agent et de gestionnaire.

**MIB**

Les informations qui caractérisent les hôtes et périphériques sont contenues dans des MIB (*Management Information Base*), elles aussi définies dans un standard.   
Les MIB sont composées d’un ensemble de variables ayant un nom et une valeur; ces valeurs peuvent être lues ou écrites à partir d’un gestionnaire SNMP. Certaines variables sont en lecture seule.

La SMI (*Structure of Management Information*) définit la structure des MIB (hiérarchie, types de data, nomenclature, ...). Les objets MIB sont regroupés en modules MIB, et chaque entité SNMP peut être définie par plusieurs modules.

Chaque objet MIB possède les 6 propriétés suivantes (au minimum) ...

* **OID**  
   Un identifiant unique ;
* **Nom**  
  Désignation de l’objet ;
* **Syntaxe**  
  Type de donnée et structure, par exemple un tableau de strings, nombre entier unique, … ;
* **Statut**  
  Définition si l’objet est conforme au standard actuel ou non ;
* **Description**  
  Texte descriptif ;
* **Accès**  
  Type d’accès à l’objet (lecture seule, lecture-écriture, …).

**Remarque** …  
Des attributs optionnels sont possibles.

Les MIB sont organisées selon une hiérarchie, et chaque nœud de cette hiérarchie est désigné par un nombre.   
Donc n’importe quel objet dans une MIB est désigné par une séquence de nombres : c’est ce nombre qu’on nomme OID. Par exemple, les adresses IP d’un hôte sont contenues dans une table nommée IpAddrTable dont l’OID est 1.3.6.1.2.1.4.20.

La plupart des objets utilisés couramment sont dans la branche 1.3.6.1, plus particulièrement 1.3.6.1.2.1 (les objets communs, qui forment la grande majorité des objets utilisés) et 1.3.6.1.4.1 (les objets définis par des entreprises privées).

Beaucoup d’informations ne font pas partie de la hiérarchie de base, mais sont plutôt définis dans des modules.

**Messages**

Le protocole SNMP permet 2 types de communication ...

* Un **gestionnaire** interroge les agents (*polling*) ;
* Les **agents** envoient des alertes (*trap*) au gestionnaire.

Le *polling* ou sondage est utile pour collecter des statistiques d’usage ou des informations de configuration.   
Toutefois, lorsqu’on veut être prévenu des problèmes dès qu’ils surviennent, un mécanisme d’alerte est nécessaire.

UDP est utilisé comme protocole de transport; le port 161 est le port d’écoute sur les agents (requêtes) et le port 162 est le port d’écoute sur les gestionnaires (alertes).

Les classes de messages sont ...

* **Lecture**   
  *GetRequest*, *GetNextRequest*, *GetBulkRequest* ;
* **Écriture**   
  SetRequest ;
* **Réponse**   
  Response ;
* **Alertes**   
  *Trapv2*, *InformRequest*

**Description des principaux messages**

* **GetRequest**  
   Le gestionnaire envoie un une requête à l’agent, qui répond avec un message Response ;
* **GetNextRequest**  
   Similaire à un GetRequest, mais le gestionnaire demande d’envoyer l’objet suivant celui qui est spécifié dans la requête ;
* **GetBulkRequest**  
   Similaire à un GetRequest, mais Response contient un ensemble d’objets ;.
* **SetRequest**  
   Le gestionnaire envoie une requête à l’agent, qui retourne un message Response après avoir fait les modifications (si possible/permis) ;
* **Trapv2**  
   L’agent envoie une alerte au gestionnaire; le gestionnaire peut retransmettre cette alerte en envoyant un message InformRequest à un autre gestionnaire, qui doit lui retourner une confirmation avec un message Response.

**Remarque** …  
Un agent qui envoie un message Trapv2 ne reçoit jamais de confirmation.

**Sécurité**

Avec SNMPv1 et v2, il n’existe pas de méthode uniforme pour assurer l’authentification et la confidentialité.

Avec SNMPv2c, on utilise les communautés pour définir l'appartenance à un groupe d'hôtes surveillés.   
Il y a deux types de communautés : celles qui sont en lecture seule (*rocommunity*) et celles qui sont en lecture/écriture (*rwcommunity*).

Avec SNMPv3, la sécurité est basée sur un système d’utilisateurs. Il y a trois niveaux de sécurité, qui peuvent être différents entre les utilisateurs ...

* **Aucune authentification et absence de chiffrement des communications** ;
* **Authentification sans chiffrement des communications** ;
* **Authentification et chiffrement des communications**.



## Protocole NTP

Dans un système d'information moderne, la synchronisation des hôtes sur une horloge commune est un pré-requis pour de nombreuses actions comme l'analyse des journaux (*logs*) ou la supervision système et réseau.

Si les hôtes du réseau sont directement connectés à Internet, il n'y a pas de problème car les distributions GNU/Linux, Windows et Mac OS X embarquent des mécanismes afin de mettre automatiquement à l'heure en se synchronisant sur des serveurs publics (par exemple 0.ca.pool.ntp.org).

Si ce n'est pas le cas et que les hôtes sont isolés et/ou bien filtrés lors de leurs accès à Internet, il peut être intéressant d'installer un serveur de temps local sur un des hôtes.

**Protocole NTP**

Le Protocole d'Heure Réseau (*Network Time Protocol* ou NTP) est un protocole qui permet de synchroniser, à partir d’un réseau informatique, l'horloge locale d'hôtes sur une référence d'heure.

Le protocole NTP utilise le port UDP 123 pour effectuer les requêtes sur le réseau. Le serveur NTP doit donc être en écoute sur ce port.

Si le réseau est protégé par un pare-feu, il faudra bien entendu le configurer pour autoriser les requêtes sur ce port.

Le NTP met à disposition des mécanismes de protocole fondamentaux et nécessaires à la synchronisation de l’heure de différents systèmes jusqu’à une précision de l’ordre de la nanoseconde. D’autre part, il contient des dispositions visant à spécifier la précision et les sources d’erreur possibles de l’horloge système locale ainsi que les propriétés de l’heure de référence.   
Le protocole fixe toutefois uniquement le type de présentation des données et les formats de messages, sans fournir personnellement d’algorithmes de synchronisation et de filtrage.

Afin de synchroniser les horloges des ordinateurs à la nanoseconde près, NTP utilise le Coordinated Universal Time, appelé temps universel coordonné (UTC), applicable uniformément depuis 1972.

Les services importants tels que le Global Positioning System (GPS ; Système Mondial de Positionnement) sont équipés de récepteurs spécifiques permettant de capter les signaux correspondants. Étant donné qu’il ne serait ni rentable ni réalisable d’équiper chaque ordinateur de tels récepteurs, il existe par ailleurs des serveurs primaires (Primary Time Servers) disposant également d’un récepteur UTC.   
À travers des protocoles comme NTP, ces serveurs procèdent alors à la synchronisation des horloges des ordinateurs dans leur réseau.

Ce processus de synchronisation dispose de différents niveaux hiérarchiques symbolisant la distance par rapport à la source UTC, que l’on appelle également stratum (mot latin pour couches) ou strate. L’ensemble des appareils techniques obtenant l’heure à partir d’un serveur primaire ou d’un système de navigation par satellite sont classés dans la catégorie stratum 0. Les horloges atomiques ou radiopilotées en font notamment partie.   
Un hôte obtenant l’UTC à partir d’une telle horloge atomique ou radiopilotée entre dans la catégorie stratum 1 et ainsi de suite. Chaque système est ainsi à la fois client du stratum précédent et serveur des systèmes du stratum suivant (tout du moins en théorie).

**pool.NTP.org -- Cluster Internet de milliers de serveurs NTP**

Il existe un immense cluster virtuel composé de plus de 4 000 serveurs d'horloge NTP. La croissance constante du pool de NTP au fil du temps est due aux efforts de la communauté de ce projet réussi : toute personne disposant d’un serveur avec une adresse IP statique et connecté en permanence à Internet peut faire enregistrer ce serveur dans le cluster. Malgré une demande grandissante, le service peut donc être utilisé gratuitement et sans restriction jusqu’à aujourd’hui.



# Références

* http://cvardon.fr/plaquetteOSI.html
* http://www.linux-france.org/prj/edu/archinet/systeme/ch01s03.html
* https://fr.wikipedia.org/wiki/IPv4
* http://www.laissus.fr/cours/node9.html#SECTION04414000000000000000
* http://www.supinfo-projects.com/fr/2006/
* http://cisco.goffinet.org/s1/modele-tcp-ip-et-protocoles#.Vssi60aj92A
* http://www.diml.org/goodies/standards/rfc768/768.dim?session
* http://www.laissus.fr/cours/node9.html#SECTION04470000000000000000
* https://fr.wikipedia.org/wiki/Domain\_Name\_System
* http://www.kloth.net/services/dig.php
* http://fr.wikipedia.org/wiki/Nmap
* <http://blog.nicolargo.com/2007/08/nmap-le-scanneur-de-reseau.html>
* Openclassroom  
  https://openclassrooms.com/fr/courses/857447-apprenez-le-fonctionnement-des-reseaux-tcp-ip

|  |  |
| --- | --- |
| **Document distribué sous licence** |  |

1. **Une annecdote** …  
   Pourquoi 802.2?   
   Tout simplement parce que la première norme a été établie en février 1980 (80 2) et que cette spécification est la seconde. [↑](#footnote-ref-2)
2. DHCP (*Dynamic Hosts Configuration Protocol*) est un protocole de la couche Applications. [↑](#footnote-ref-3)
3. Deux fournisseurs pour l’attribution des adresses réseau IPv4 ..  
    ICANN ou *Internet Corporation for Assigned Names and Numbers* et  
    RIR ou *Regional Internet Registries*. [↑](#footnote-ref-4)
4. NAT ou *Network Address Translation* [↑](#footnote-ref-5)