Chapitre 6

Le protocole IP (*Internet Protocol*)

IP transfère les données à travers une interconnexion de réseaux. Il est utilisé par les protocoles de la couche de transport, TCP et UDP. Il cherche un chemin pour transférer les données (*datagrammes*) d'un équipement émetteur, identifié par son adresse IP, à un équipement destinataire, identifié lui aussi par son adresse IP. Dans une machine quelconque, le module IP ne fournit aucune garantie d'un acheminement correct des données et ne gère aucun dialogue avec le module IP d'une autre machine. Chaque datagramme est géré indépendamment des autres. Cela signifie que ceux-ci peuvent être mélangés, dupliqués, perdus ou altérés ! Pour comprendre le fonctionnement du protocole IP, nous allons d'abord voir les adresses IP elles-mêmes ainsi que leur correspondance avec les adresses physiques, le traitement effectué par un module IP et le format du datagramme IP.

1 Les adresses IP (*Internet Protocol*)

L'adressage utilisé dans Internet est un adressage logique. Chaque équipement possède un nom symbolique auquel on fait correspondre l'adresse logique appelée *adresse IP*. Celle-ci se décompose en deux parties : l'*identificateur du réseau*, où se trouve l'équipement, et l'*identificateur de la machine* elle-même (qui a une signification locale à ce réseau). L'ensemble tient sur 32 bits soit 4 octets. L'adresse IP est le plus souvent écrite en *notation décimale pointée* : les octets sont séparés par des points, et chaque octet est représenté par un nombre décimal compris entre `0` et `255`.

Exemple

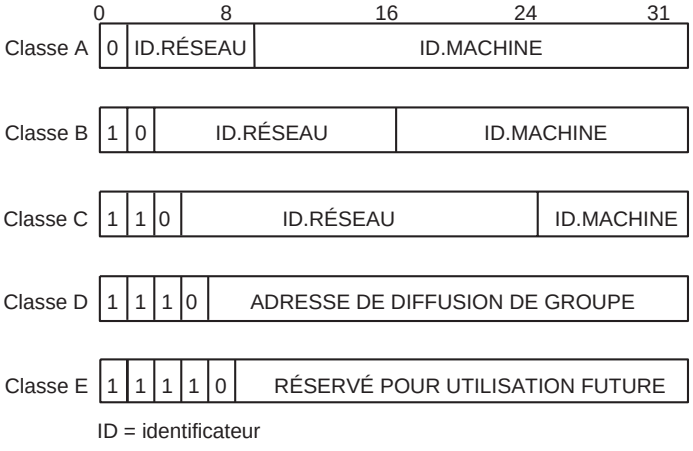
Adresse IP = `11000001 00011011 00101101 00100001` en binaire soit `193.27.45.33` en notation décimale pointée.

1.1 Les classes d'adresses

Plusieurs classes d'adresses sont définies : un réseau ayant beaucoup de machines dispose d'une adresse avec un champ *identificateur de réseau* court et un champ *identificateur de machine* long. En revanche, dans un petit réseau local, l'identificateur de machine sera codé sur peu d'éléments binaires. La classe d'adresse et l'identificateur de réseau sont attribués par un organisme central, l'ICANN (*Internet Corporation for Assigned Names and Numbers*), qui gère le plan d'adressage à l'échelle mondiale et garantit l'unicité des numéros de réseau. L'administrateur local du réseau attribue ensuite les numéros de machine aux différents équipements de son réseau, selon le plan d'adressage qu'il a conçu.

L'identificateur de réseau est codé sur 7, 14 ou 21 bits selon la classe d'adresse. Les adresses de classe *A* affectent 7 bits à l'identité de réseau et 24 bits à l'identité de machine ; les adresses de classe *B* affectent 14 bits à l'identité de réseau et 16 bits à l'identité de machine. Enfin, les adresses de classe *C* allouent 21 bits à l'identité de réseau et 8 bits à l'identité de machine. Les adresses de classe *D* sont réservées pour mettre en œuvre le mécanisme de diffusion de groupe. La classe d'une adresse IP est déterminée à partir des bits de poids fort, comme le montre la figure 6.1.

Figure 6.1
Différents formats d'adresse IP.



Les très grands réseaux ont des adresses de classe *A*, dont le premier bit du premier octet est à 0 tandis que les 7 autres bits servent à identifier 126 réseaux différents. Chaque réseau de classe *A* possède 24 bits d'identifiant de machine, ce qui permet d'adresser $2^{24} - 2$, soit 16 777 214 machines (les deux identifiants 0 et 16777215 sont, par convention, réservés à un autre usage).

Les réseaux de taille moyenne ont des adresses de classe *B*. Elles commencent en binaire par `10` et affectent 14 bits à l'identifiant de réseau ; il reste 16 bits pour identifier les machines, soit au maximum 65 534 (pour la même raison que précédemment, les identifiants 0 et 65535 ne sont pas attribués à une machine).

Enfin, pour les petits réseaux, les adresses de classe *C* commencent en binaire par `110` et allouent 21 bits à l'identifiant de réseau et 8 bits à l'identifiant de machine. On peut ainsi adresser jusqu'à 254 machines (les identifiants 0 et 255 ne sont pas utilisés).

Les adresses de classe *D*, commençant en binaire par `1110`, sont réservées à la mise en œuvre d'un mécanisme de diffusion de groupe (*multicast*). Dans une communication multicast, un utilisateur émet un message dont l'adresse de destination est celle du groupe. Le message est acheminé en un seul exemplaire le plus loin possible, jusqu'à ce qu'il soit indispensable de l'éclater en autant de messages individuels que le groupe possède de membres. La plupart des adresses multicast allouées le sont à des groupes d'utilisateurs concernés par une même application (la radio sur Internet par exemple. On comprend dans ce cas l'intérêt d'envoyer en multicast au lieu d'envoyer massivement autant de messages qu'il y a de récepteurs). Dans une adresse multicast, les 28 bits restants n'ont pas de structure particulière.

Exemple

Adresse IP = `11000001 00011011 00101101 00100001` soit `193.27.45.33`. Il s'agit d'une adresse de classe *C* puisqu'elle commence en binaire par `110`. Le découpage est alors le suivant : `110` (3 bits pour la classe *C*) `00001 00011011 00101101` (21 bits d'identifiant de réseau) et `00100001` (8 bits identifiant la machine dans le réseau).

1.2 Adresses particulières

Lorsqu'une machine ne possède pas d'adresse IP et qu'elle doit envoyer un (premier) message pour en obtenir une, elle remplit le champ Adresse du message par « plein 0 » ou `0.0.0.0` en notation décimale pointée. À l'opposé, remplir le champ Adresse par « plein 1 » permet de désigner l'ensemble des machines au sein du réseau dans lequel se trouve la machine.

Les réseaux eux-mêmes possèdent chacun une adresse : celle-ci est obtenue en remplaçant le champ Identifiant de machine par « plein 0 », conformément à la classe.

Exemple

La machine `37.194.192.21` appartient au réseau `37.0.0.0` (classe *A*).
La machine `137.194.192.21` appartient au réseau `137.194.0.0` (classe *B*).
La machine `197.194.192.21` appartient au réseau `197.194.192.0` (classe *C*).

Une *adresse de diffusion (broadcast address)*, désigne l'ensemble des machines d'un réseau distant. Elle est constituée en remplaçant le champ Identifiant de machine par « plein 1 ».

Exemple

L'adresse de diffusion du réseau `37.0.0.0` est `37.255.255.255` (classe *A*).
L'adresse de diffusion du réseau `137.194.0.0` est `37.194.255.255` (classe *B*).
L'adresse de diffusion du réseau `197.194.192.0` est `197.194.192.255` (classe *C*).

Certains identifiants de réseau n'existent pas, en particulier les réseaux de classe *A* : `0` et `127`. `0` est réservé à l'usage décrit ci-avant et `127` sert aux tests locaux. Par exemple, l'adresse `127.0.0.1` est *a priori* affectée à chaque carte réseau. Tout message envoyé à cette adresse est directement retourné à son expéditeur, sans aucune émission sur le réseau : cela permet de vérifier que la pile TCP/IP fonctionne correctement. Notons que cette adresse, appelée *adresse de boucle locale (loopback address)*, n'a aucun rapport avec la notion de boucle locale utilisée pour la desserte des usagers du réseau téléphonique.

1.3 Notions de sous-réseaux et de masque

La hiérarchie à deux niveaux (réseau et machine) de l'adressage IP s'est rapidement révélée insuffisante à cause de la diversité des architectures des réseaux connectés. La notion de sous-réseau (ou *subnet*), introduite en 1984, a conservé le format de l'adresse IP sur 32 bits. Dans un réseau subdivisé en plusieurs sous-réseaux, on exploite autrement le champ Identifiant de machine de l'adresse IP. Celui-ci se décompose désormais en un identifiant de sous-réseau et un identifiant de machine. Remarquons que ce découpage n'est connu qu'à l'intérieur du réseau lui-même. En d'autres termes, une adresse IP, vue de l'extérieur, reste une adresse sur 32 bits avec ses deux champs. On ne peut donc pas savoir si le réseau est constitué d'un seul réseau ou subdivisé en plusieurs sous-réseaux.

L'administrateur local choisit le nombre de bits à consacrer à l'identifiant de sous-réseau grâce au *masque de sous-réseau* (ou *subnet mask*). Celui-ci, également codé sur 32 bits, définit le découpage de l'identifiant machine en deux champs (Sous-réseau et Machine). Dans un réseau subdivisé, chaque machine connaît son adresse IP et le masque utilisé, ce qui lui permet de savoir dans quel sous-réseau elle se trouve. Il suffit de faire un ET logique entre son adresse IP et le masque :

Exemple

Adresse IP : `193. 27. 45. 33` = `11000001 00011011 00101101 00100001`
Masque : `255.255.255.224` = `11111111 11111111 11111111 11100000`
Comparaison sous masque : `11000001 00011011 00101101 00100001`
ET : `11111111 11111111 11111111 11100000`
`11000001 00011011 00101101 00100000`

Le résultat `193.27.45.32` est l'adresse du sous-réseau auquel appartient la machine `193.27.45.33`.

Lorsqu'un équipement d'un (sous-)réseau veut émettre un message à un autre, il compare sa propre adresse bit à bit avec celle du destinataire, en utilisant le masque de sous-réseau. S'il y a égalité sur toute la partie identifiée par les 1 du masque, les deux équipements se trouvent dans le même (sous-)réseau et le message peut donc être transmis directement. Sinon, il est envoyé au routeur, la machine qui assure l'acheminement du message vers l'extérieur du (sous-)réseau.

Exemple

Soit une adresse IP de réseau de classe *C* `193.27.45.0`. Le masque de sous-réseau vaut : `255.255.255.224`, soit en binaire : `11111111 11111111 11111111 11100000`.

Nous voyons donc que, dans l'octet réservé au champ Identifiant de machine, trois bits sont utilisés pour identifier des sous-réseaux interconnectés par un routeur. Sur le sous-réseau 1, l'adresse du sous-réseau est `193.27.45.32`. Si l'administrateur décide d'affecter l'identifiant 1 au routeur dans tous les sous-réseaux, l'adresse du routeur dans le sous-réseau 1 est : `193.27.45.33` ; l'adresse de diffusion dans le sous-réseau valant `193.27.45.63`, il reste donc 29 adresses disponibles sur les 32 possibles pour les stations du sous-réseau 1. De même, dans le sous-réseau 2, l'adresse de sous-réseau étant `193.27.45.64`, celle du routeur vaut `193.27.45.65` et l'adresse `193.27.45.95` est celle de diffusion dans le sous-réseau. Il reste également 29 adresses disponibles sur les 32 possibles pour les stations du sous-réseau 2.

Remarque

Dans un réseau plat (sans sous-réseau), on peut mettre 254 stations sur un réseau de classe *C*. Avec six sous-réseaux physiques comme dans cet exemple (on a exclu de fait les numéros `0` et `7`), on ne peut en mettre que 174, mais on dispose d'une identification plus fine et d'une possibilité de diffusion limitée à chaque sous-réseau.

1.4 Pénurie d'adresses

Le formidable succès d'Internet a mené à l'épuisement des adresses de classes *A* et *B* et à l'explosion des tables de routage des routeurs situés dans les réseaux de transit. Si beaucoup d'organisations possèdent plus de 254 ordinateurs, peu en possèdent plusieurs milliers (or, une adresse de classe *B* permet d'identifier jusqu'à 65 534 machines). Ce manque de souplesse entre les classes explique les différentes solutions mises en œuvre dès les années 1990 : l'utilisation d'adresses privées, d'adresses sans classe et la distribution dynamique des adresses.

Les adresses privées

Plusieurs plages d'adresses IP ont été réservées dans chaque classe d'adresses et sont d'utilisation libre. Elles sont appelées *adresses IP privées* et sont décrites dans la RFC 1918 (voir tableau 6.1).

Tableau 6.1

Les adresses IP privées en classes A, B et C

Classe	Adresses privées	Nombre maximal de machines
A	**10.x.y.z**, où $0 \leq x \leq 255$ $0 \leq y \leq 255$ et $0 \leq z \leq 255$	(256*256*256) – 2 = 16 777 214
B	**172.x.y.z**, où $16 \leq x \leq 31$ $0 \leq y \leq 255$ et $0 \leq z \leq 255$	(15*256*256) – 2 = 1 048 574
C	**192.168.x.y**, où $0 \leq x \leq 255$ et $0 \leq y \leq 255$	(256*256) – 2 = 65 534

Ces adresses ne peuvent être attribuées par l'ICANN à une organisation. Ainsi, des réseaux différents peuvent utiliser les mêmes adresses IP privées, pourvu qu'ils restent isolés les uns des autres.

Pour relier à Internet les machines d'un réseau utilisant des adresses privées, on met en place une traduction, gérée par le routeur, entre adresses IP privées (internes au réseau de l'organisation et inaccessibles de l'extérieur) et adresses IP publiques (visibles de l'extérieur, c'est-à-dire accessibles par Internet). Une adresse IP publique est unique ; elle est dite *routable*, car elle seule autorise l'accès à Internet. La correspondance entre les deux types d'adresses est assurée par le NAT (*Network Address Translation*), un mécanisme de conversion d'adresse décrit par la RFC 3022. De plus, les adresses IP privées garantissent une meilleure sécurité d'accès aux réseaux d'organisation, dans la mesure où les adresses réelles utilisées par les machines du réseau ne sont pas connues de l'extérieur.

Le NAT statique crée une bijection entre adresses IP publiques et adresses IP privées internes au réseau. Le routeur associe une adresse IP privée (par exemple `192.168.0.1`) à une adresse IP publique routable sur Internet. Il fait la traduction, dans un sens comme dans l'autre, en modifiant l'adresse dans le paquet IP. Le NAT statique connecte des machines du réseau interne à Internet de manière transparente, mais ne résout pas le problème de la pénurie d'adresses, dans la mesure où *n* adresses IP routables sont nécessaires pour connecter *n* machines du réseau interne.

Le NAT dynamique partage une adresse IP routable (ou un nombre réduit d'adresses IP routables) entre plusieurs machines dotées d'une adresse privée. Ainsi, vu de l'extérieur, toutes les machines du réseau interne possèdent la même adresse IP.

Les adresses IP privées sont donc la garantie d'une sécurité accrue. De plus, elles constituent une réponse au manque d'adresses IP. Leurs inconvénients sont : le travail supplémentaire lors de la configuration du réseau et la renumérotation à envisager lors de la fusion d'entreprises qui utiliseraient les mêmes adresses IP privées.

Les adresses sans classe CIDR (*Classless InterDomain Routing*)

Le CIDR a été proposé à partir de 1994. L'idée est d'organiser une adresse réseau indépendamment de sa classe ; le masque de sous-réseau indiquant le nombre de bits réservés à l'identifiant réseau est alors fixé librement par l'administrateur. Par exemple, pour réaliser l'équivalent de deux adresses de classe *C* contiguës, l'administrateur choisira un masque `/23`. Naturellement, le CIDR s'utilise également pour les adresses privées, telles qu'elles sont définies dans la RFC 1918.

Exemple

L'entreprise s'est vu attribuer l'adresse : `12.22.36.0 /22`. Cela veut dire que l'identifiant réseau tient sur 22 bits. Il reste 10 bits que l'administrateur peut affecter librement. Dans certaines parties du réseau, il peut décider d'utiliser un masque `/23`, dans d'autres il prendra un masque `/25` mais jamais un masque de taille inférieure à 22 bits dans notre exemple.

L'avantage de CIDR est de s'affranchir des contraintes imposées par le format des classes d'adresses. Les seules restrictions qui demeurent concernent les adresses dévolues au réseau lui-même, à la diffusion dans le réseau et aux anciennes classes *D* (réservée au multicast) et *E* (classe d'extension) d'IPv4.

Le tableau 6.2 donne des exemples d'allocation d'adresses en fonction des besoins de l'organisation (la liste est bien évidemment non exhaustive. Elle est à adapter aux besoins du réseau considéré). Nous donnerons d'autres exemples dans les compléments pédagogiques, sur le site www.pearsoneducation.fr.

Tableau 6.2

Exemples d'allocation d'adresses CIDR

Besoin de l'organisation	Allocation d'adresses	Masque utilisé
< 64 adresses	Subdivision de classe *C*	/26
< 128 adresses	Subdivision de classe *C*	/25
< 256 adresses	**1 réseau de classe C**	**/24**
< 512 adresses	2 réseaux de classe *C* contigus	/23
< 1 024 adresses	4 réseaux de classe *C* contigus	/22
< 2 048 adresses	8 réseaux de classe *C* contigus	/21
< 4 096 adresses	16 réseaux de classe *C* contigus	/20

Exemple

Soit une entreprise possédant 780 équipements dans son réseau qui obtient 4 adresses de classe *C* consécutives : `194.42.36.0`, `194.42.37.0`, `194.42.38.0` et `194.42.39.0`. Pour elle, tout se passe comme si son identifiant de réseau était `11000010 00101010 001001`, puisqu'il y a 22 bits en commun sur les identifiants réseau. En effet, dans le troisième octet, les nombres 36, 37, 38 et 39 ont en commun les 6 premiers bits `001001`. La notation de l'adresse IP d'un équipement quelconque dans ce réseau sera, par exemple, `194.42.37.156/22`, où `/22` (prononcer *slash* 22) signifie : « Dont les 22 premiers bits sont l'identifiant de réseau. » Sur les 32 bits de l'adresse, il en reste $32 - 22 = 10$ pour l'identifiant de machine, ce qui permet 2^{10} adresses différentes et convient à cette entreprise.

La distribution dynamique des adresses

Une autre solution pour gérer la pénurie des adresses consiste à utiliser une plage d'adresses (éventuellement trop petite pour le parc de machines), en allouant temporairement les adresses IP disponibles aux seules machines connectées et en partant de l'hypothèse que toutes ne le seront pas simultanément. Pour assurer la distribution dynamique des adresses, le protocole DHCP (*Dynamic Host Configuration Protocol*) fournit automatiquement à un ordinateur qui vient d'être installé dans le réseau de l'entreprise ses paramètres de configuration réseau (adresse IP et masque de sous-réseau). De plus, cette technique sim-

plifie la tâche de l'administrateur d'un grand réseau, en évitant les doublons d'adresses. Elle peut se mettre en œuvre aussi bien avec des adresses publiques qu'avec des adresses privées et peut évidemment servir lorsque la plage d'adresses IP est plus grande que le parc de machines à identifier. Cette technique étant vue comme une application au sens de l'architecture de communication, nous l'aborderons au chapitre 9 qui traite des applications.

1.5 Association des adresses IP et des adresses MAC (Medium Access Control)

Au chapitre précédent, nous avons vu que les équipements étaient identifiés dans les réseaux locaux par leur adresse MAC, numéro de série de la carte réseau qui y est installée. Maintenant, les réseaux locaux sont interconnectés et « plongés » dans des réseaux logiques, qui attribuent à leurs équipements des adresses IP. Le même équipement possède donc deux adresses, l'une physique, l'autre logique, et il est nécessaire de pouvoir faire l'association entre les deux.

Soit deux machines *A* et *B*, connectées à un même réseau local. Chaque machine a une adresse IP, respectivement IP_A et IP_B, et une adresse MAC (numéro de série de la carte Ethernet, par exemple), respectivement MAC_A et MAC_B. Le problème, nommé « problème de résolution d'adresse ou *address resolution problem*, consiste à faire la correspondance entre adresses IP et adresses MAC, sachant que les programmes d'application ne manipulent – au mieux – que des adresses IP et que le réseau local ne voit que les adresses MAC. Dans chaque machine, des tables contiennent des paires adresse IP/adresse MAC, mais elles ne peuvent maintenir qu'un petit nombre de paires d'adresses. Un protocole de résolution d'adresses (ARP, *Address Resolution Protocol*) fournit un mécanisme efficace et simple. Il est défini dans la RFC 826.

Le protocole ARP (*Address Resolution Protocol*) [RFC 826]

Le protocole ARP établit une correspondance dynamique entre adresses physiques et adresses logiques. Il permet à une machine de trouver l'adresse physique d'une machine cible située sur le même réseau local, à partir de sa seule adresse IP. Lorsqu'une machine *A* veut résoudre l'adresse IP_B, elle diffuse un message spécial (en utilisant l'adresse MAC `FF:FF:FF:FF:FF:FF` comme identité de destinataire). Celui-ci demande à la machine d'adresse IP_B de répondre en indiquant son adresse physique MAC_B. Toutes les machines, y compris *B*, reçoivent ce message puisqu'il est envoyé en diffusion ; seule la machine *B* reconnaît son adresse IP. Elle répond en envoyant son adresse MAC_B. Lorsque *A* reçoit cette réponse, elle peut alors communiquer directement avec *B*. Les messages spéciaux que nous venons de voir, ceux du protocole ARP, sont véhiculés dans les données de la trame du réseau local. Un protocole similaire, baptisé RARP (*Reverse Address Resolution Protocol*) permet de la même façon, pour une machine sans disque, de connaître son adresse IP auprès d'un serveur d'adresses.

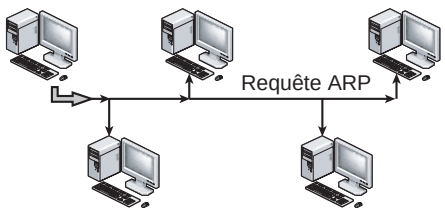
Le format de la requête/réponse ARP est très malléable car la taille des adresses n'est pas définie à l'avance et se trouve donc codée dans le message lui-même. La requête/réponse ARP contient (voir figure 6.2) :

- *L'adresse physique de l'émetteur.* Dans le cas d'une requête ARP, l'émetteur place son adresse ; dans une réponse ARP, ce champ révèle l'adresse recherchée.
- *L'adresse logique de l'émetteur* (l'adresse IP de l'émetteur).
- *L'adresse physique du récepteur.* Dans le cas d'une requête ARP, ce champ est vide.
- *L'adresse logique du récepteur* (l'adresse IP du récepteur).

Un mécanisme de cache permet de conserver les informations ainsi acquises : chaque système dispose d'une table pour sauvegarder les correspondances (adresse MAC, adresse IP). Ainsi, une requête ARP n'est émise que si le destinataire est absent dans la table.

Figure 6.2

Requête ARP en diffusion.



Exemple

La commande `arp -a` affiche le contenu de la table, aussi bien sous Windows que sous Unix :

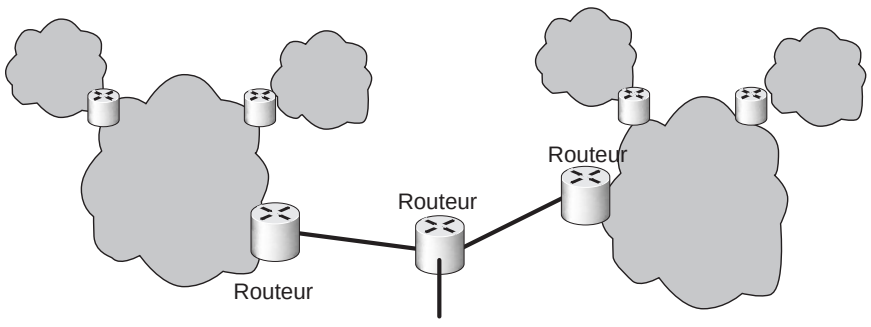
```
arp -a
poste_1(192.168.1.2) at 02:54:05:F4:DE:E5 [ether] on eth0
poste_2(192.168.1.1) at 02:54:05:F4:62:30 [ether] on eth0
```

2 Service rendu par le protocole IP

Un module logiciel gérant le protocole IP est installé dans tous les équipements du réseau local. Un équipement d'interconnexion, le *routeur*, gère l'accès au monde extérieur (voir figure 6.3). Pour bien comprendre le service rendu par le protocole IP, nous décrivons le principe de traitement d'un message au format IP (un *datagramme*) à travers l'interconnexion de réseau, en dehors de tout événement anormal.

Figure 6.3

Interconnexion de réseaux IP par des routeurs.



2.1 Création d'un datagramme par un équipement émetteur

Lorsqu'une application qui s'exécute sur l'équipement *A* veut transmettre des données à l'équipement *B*, elle donne un ordre au module IP de la machine *A*, traduit par une *requête d'émission* comprenant deux paramètres : les données et l'adresse du destinataire IP_B. Dans son fichier de configuration, le module IP de la machine *A* possède des informations comme : l'adresse IP_A, le masque de sous-réseau, l'adresse IP_R du routeur permettant de sortir du réseau. IP_R est considérée par l'équipement comme la route par défaut pour joindre tout destinataire extérieur au réseau. Le module IP de la machine *A* fabrique alors un datagramme, conformément au format défini par le protocole, puis il sollicite sa carte réseau pour que le datagramme soit encapsulé dans une trame du réseau local et transmis jusqu'au destinataire final (si *B* est dans le même réseau local) ou

jusqu'au routeur (si *B* est à l'extérieur du réseau). Le traitement de la requête d'émission est terminé pour la machine *A* : il n'y a aucun établissement de connexion entre *A* et *B*, aucun dialogue préalable entre eux. Le module IP gère les requêtes une par une, indépendamment les unes des autres. Aucun accusé de réception n'est prévu dans le protocole.

2.2 Traitement du datagramme par un routeur

Le routeur est, par définition, un équipement connecté à au moins deux réseaux IP. Il possède donc au moins deux interfaces réseaux (éventuellement différentes l'une de l'autre) et, de ce fait, a*u moins deux* adresses IP (en fait, il possède autant d'adresses IP que d'interfaces physiques différentes).

Lorsqu'un routeur reçoit sur l'une de ses interfaces réseau la trame qui lui est adressée, il la décapsule et en extrait le datagramme IP qu'elle contenait ; il lit alors l'adresse IP de destination (ici IP_B). Le rôle du routeur est de chercher dans sa *table de routage* la route qui permet d'atteindre le réseau de *B*. La table de routage contient la liste de réseaux connus ainsi que l'adresse du *routeur suivant* pour les atteindre (et l'indication locale de l'interface réseau concernée). À partir des informations de la table, il peut en déduire le chemin à suivre. Par exemple :

pour aller à	réseau *IP X*	passer par	routeur *R2*	en utilisant la carte d'interface	Ethernet2
pour aller à	réseau *IP Y*	passer par	routeur *R3*	en utilisant la carte d'interface	Ethernet2
pour aller à	réseau *IP Z*	passer par	routeur *R3*	en utilisant la carte d'interface	Ethernet1
pour aller à	ailleurs	passer par	routeur *R1*	en utilisant la carte d'interface	ls1

La table de routage contient toujours une entrée par défaut, dans le cas où le réseau du destinataire ne serait pas connu. Le routeur sollicite donc la carte d'interface spécifiée dans sa table et lui demande d'expédier le datagramme au routeur indiqué. Le datagramme est encapsulé dans une nouvelle trame, à destination d'un deuxième routeur, et ainsi de suite jusqu'au réseau du destinataire. Le routeur n'établit aucune connexion ni avec *A* ni avec le routeur suivant, il n'y a aucun dialogue préalable entre eux. Il gère les datagrammes, indépendamment les uns des autres. Aucun accusé de réception n'est prévu. Deux datagrammes qui se suivraient avec même émetteur et même destinataire peuvent ne pas suivre la même route et donc ne pas arriver au destinataire final (s'ils arrivent…), dans l'ordre où ils ont été émis : il suffit que la table de routage ait été mise à jour à la suite d'un incident par exemple.

2.3 Réception d'un datagramme par un équipement destinataire

Le datagramme traverse ainsi l'interconnexion de réseaux, de routeur en routeur, jusqu'à atteindre le routeur d'entrée du réseau de *B.* Encapsulé une dernière fois dans une trame du réseau local de *B*, il parvient à son destinataire. La carte réseau de la machine *B* reçoit cette dernière trame qui lui est adressée ; elle en sort le datagramme qu'elle livre à son module IP. Celui-ci analyse l'adresse et reconnaît la sienne : il signale alors à l'application concernée l'arrivée des données, ce qui se traduit par une *indication de réception* avec deux paramètres : les données et l'adresse de l'émetteur IP_A. Le traitement est terminé pour *B.*

2.4 CONCLUSION

Le protocole IP, implanté dans tous les équipements du réseau (machines et routeurs), assure un service de remise des données non fiable et sans connexion. Il comprend la définition du plan d'adressage, la structure des informations transférées (le datagramme IP) et les règles de routage. Nous avons vu que les datagrammes sont indépendants les uns des autres ; ils sont acheminés à travers l'interconnexion, en fonction des adresses IP publiques (source et destination). Les différents routeurs choisissent un chemin à travers les réseaux ; ils fragmentent les datagrammes lorsque le réseau suivant n'accepte que des messages de taille plus petite. La MTU (*Maximum Transfer Unit*) d'un réseau Ethernet est par exemple de 1 500 octets, celle d'un autre réseau peut être de 128 octets. Une fois le datagramme morcelé, les fragments sont acheminés comme autant de datagrammes indépendants jusqu'à leur destination finale où ils doivent être réassemblés.

Pour trouver un chemin jusqu'au destinataire, les routeurs s'échangent, dans des messages spéciaux, des informations de routage concernant l'état des différents réseaux. Ces informations sont véhiculées elles aussi par IP. Elles servent à mettre à jour les tables de routage qui indiquent, pour chaque identifiant de réseau, si les machines situées dans le réseau sont accessibles directement ou non. Le routage est *direct* si les machines appartiennent au même réseau, sinon il est *indirect*. Dans ce cas, le routeur émetteur envoie le datagramme au routeur suivant ; la coopération des deux routeurs permet de bien acheminer le datagramme. Différents protocoles comme GGP (*Gateway to Gateway Protocol*), EGP (*Exterior Gateway Protocol*), RIP (*Routing Information Protocol*), OSPF (*Open Shortest Path First*) sont utilisés entre les différents types de routeurs pour échanger et effectuer la mise à jour des informations de routage. Nous les aborderons au chapitre 8 consacré au routage.

L'envoi de messages d'erreur est prévu en cas de destruction de datagrammes, de problèmes de remise ou d'acheminement ; ces messages sont gérés par ICMP (*Internet Control Message Protocol*), un protocole que nous verrons à la section 4.

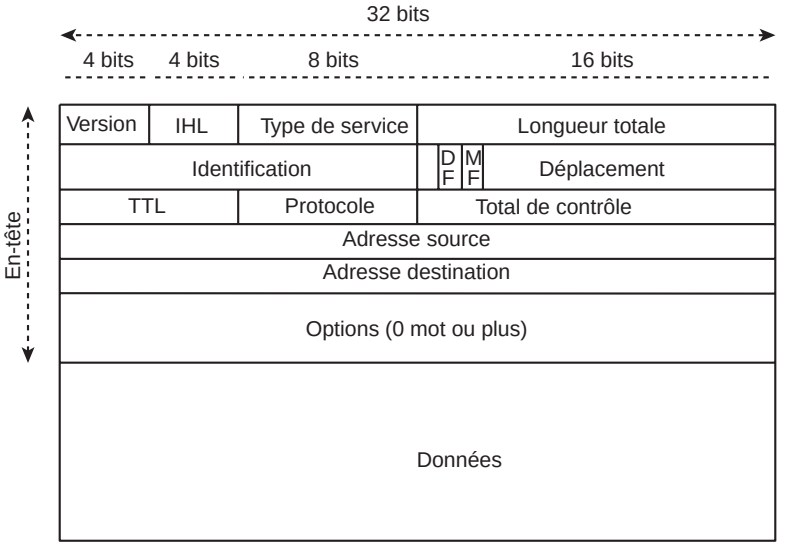
3 Format du datagramme IP

Le datagramme IP comprend un en-tête et des données. L'en-tête contient principalement les adresses IP de la source et du destinataire, et des informations sur la nature des données transportées (voir figure 6.4).

Classiquement, les différents champs sont décrits par des mots de 32 bits. La première ligne de la figure 6.4 contient quatre champs :

- *Version*. Il s'agit de la version du protocole IP qu'on utilise (actuellement, c'est la version 4 ou IPv4) afin de vérifier la validité du datagramme. La version est codée sur 4 bits.
- *Longueur de l'en-tête*. Le nombre de mots de 32 bits de l'en-tête (qui commence avec le champ version). La longueur est également codée sur 4 bits. De ce fait, un en-tête IP contient (en hexadécimal) au maximum F mots de 32 bits, soit 60 octets.
- *Type de services* (ToS). Ce champ de 8 bits indique la façon dont le datagramme doit être traité. Historiquement, il était possible de demander que le datagramme soit traité sur la route la plus rapide, sur celle qui offrait le meilleur débit, la plus fiable, etc. Encore fallait-il être capable de mesurer l'état des routes et de gérer les options… Les premières implémentations du protocole IP ont vite abandonné cette idée de services différenciés. Le champ ToS est resté à 0, d'autant que plusieurs propositions incompa-

Figure 6.4
Le datagramme IP.



tibles les unes avec les autres ont été faites pour modifier l'attribution de ce champ. Nous verrons à la section 5 que la version IPv6 reprend, sous une forme différente, l'idée de qualité de service.

- *Longueur totale*. Ce champ de 16 bits exprime en octets la taille totale du datagramme (en-tête + données). La longueur maximale d'un datagramme est donc 64 Ko, mais des raisons physiques imposent des tailles inférieures dans la plupart des réseaux.

Le deuxième mot de 32 bits concerne la fragmentation. Le champ *Identification* est un numéro de 16 bits attribué à chaque datagramme. Chaque fragment d'un même datagramme reprend le même identifiant, pour permettre le réassemblage correct du datagramme initial chez le destinataire. Après un premier bit non utilisé, les deux bits suivants sont des *drapeaux* qui permettent le réassemblage :

- *DF* : *Don't Fragment* (le deuxième bit). Autorise ou non la fragmentation du datagramme (si DF = 0 la fragmentation est autorisée, interdite si DF = 1). Par convention, toute machine doit pouvoir transmettre en un seul datagramme des données de 476 octets.
- *MF* : *More Fragments* (le dernier bit). Indique si le fragment de données est suivi ou non par d'autres fragments (si MF = 0, le fragment est le dernier du datagramme).

Le champ *Déplacement* permet de connaître la position du début du fragment par rapport au datagramme initial. Le fragment doit avoir une taille qui est un multiple entier de 8 octets. Le déplacement est codé sur les 13 derniers bits du mot.

Le troisième mot de 32 bits contient trois champs :

- *Durée de vie* (*TTL*, *Time To Live*). Indique sur 8 bits le nombre maximal de routeurs que le datagramme peut traverser. Ce champ était prévu à l'origine pour décompter un temps, d'où son nom. La durée de vie est choisie par l'émetteur ; elle est décrémentée chaque fois que le datagramme traverse un routeur. Lorsque la durée de vie atteint la valeur nulle, le datagramme est détruit.
- *Protocole*. Champ de 8 bits indiquant à quel protocole sont destinées les données véhiculées dans le datagramme. Les valeurs décimales les plus courantes sont : 1 pour

ICMP, 2 pour IGMP (*Internet Group Management Protocol*, ou protocole de gestion des groupes multicast), 6 pour TCP et 17 pour UDP.

- *Header checksum.* Ces 16 bits suivants constituent un bloc de contrôle d'erreur pour l'en-tête : ce champ permet de contrôler l'intégrité de l'en-tête. Celui-ci, en effet, transporte toutes les informations fondamentales du datagramme. Si, par hasard, il était détecté en erreur, le datagramme serait directement écarté. Remarquons qu'il n'y a aucune protection concernant les données transportées dans le datagramme.

Les deux derniers mots de 32 bits contiennent, dans cet ordre, l'*adresse IP source* et l'*adresse IP destination*. Ces cinq mots constituent l'en-tête minimal, commun à tous les datagrammes IP. En plus de ces informations, l'en-tête peut contenir en option des informations supplémentaires. C'est pourquoi il faut en indiquer la longueur.

Les options doivent tenir sur un nombre entier de mots de 32 bits. Parmi elles, le *routage* et l'*horodatage* sont particulièrement intéressantes (l'horodatage demande à chaque routeur d'estampiller le datagramme avec la date et l'heure à laquelle il a été traité). Elles constituent un bon moyen de surveiller ou de contrôler la traversée des datagrammes dans le réseau. Une autre option, l'*enregistrement de route*, demande à chaque routeur traversé de placer sa propre adresse dans le datagramme. Le destinataire reçoit ainsi un datagramme contenant la liste des adresses des routeurs traversés. Le *routage défini par la source*, lui, permet à l'émetteur d'imposer le chemin par lequel doit passer un datagramme.

3.1 Contrôle du datagramme dans un routeur

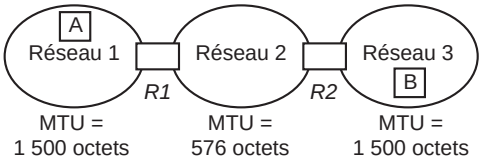
L'en-tête du datagramme est protégé par un bloc de contrôle d'erreur. Avant tout traitement, celui-ci est vérifié ; si des erreurs y sont détectées, le datagramme est détruit. D'autre part, le passage par un routeur provoque la diminution de 1 du champ durée de vie. Celui-ci faisant partie de l'en-tête, il est inclus dans le calcul du bloc de contrôle d'erreur. Le routeur l'ayant modifié, il doit donc recalculer le bloc de contrôle d'erreur avant de transférer le datagramme.

3.2 Gestion de la fragmentation

Maintenant que nous connaissons les détails du format d'un datagramme, examinons comment le fragmenter. Soit un réseau 1 où la MTU est 1 500 octets ; le routeur *R1* de ce réseau le relie à un réseau 2 de MTU égale à 576 octets. Un routeur *R2* le relie à un réseau 3 de MTU 1 500 octets (voir figure 6.5). La machine *A* du réseau 1 envoie un datagramme contenant 1 480 octets de données à la machine *B* située sur le réseau 3.

Figure 6.5

Nécessité de fragmentation dans un réseau intermédiaire.



Le datagramme fabriqué par *A* porte l'identification 17C9 (hexadécimal), le bit DF à 0 (fragmentation autorisée), le bit MF et le champ Déplacement à 0. Sachant que l'en-tête d'un datagramme contient 20 octets, le routeur *R1* va fragmenter le datagramme en trois morceaux pour le passer dans le réseau 2. Les deux premiers fragments contiendraient

556 octets de données s'il n'y avait la contrainte d'une taille multiple de 8 octets. Il faut donc choisir la taille la plus proche soit 552 octets. Le champ Déplacement vaut alors 552/8 = 69 pour le deuxième fragment puisque le déplacement est exprimé en bloc de 8 octets. Le dernier fragment n'en compte que 376. Les trois fragments auront chacun un en-tête de 20 octets qui est celui du datagramme initial, sauf pour le bit MF et le champ Déplacement, comme le montre le tableau 6.3.

Tableau 6.3

En-tête des différents fragments

	Identification	MF	Déplacement
Fragment 1	17C9	1	0
Fragment 2	17C9	1	69
Fragment 3	17C9	0	138

Tous les datagrammes issus d'une fragmentation deviennent des datagrammes IP comme les autres. Ils peuvent arriver à destination, éventuellement dans le désordre. IP doit faire le tri en utilisant les informations de l'en-tête pour faire le réassemblage. Au bout d'un certain temps, si un fragment manque toujours, la totalité du datagramme est considérée comme perdue. Puisque aucun mécanisme de contrôle n'est implémenté dans IP, la fragmentation est une source d'erreurs supplémentaire.

4 Protocole ICMP (*Internet Control Message Protocol*)

Dans l'interconnexion de réseaux, chaque routeur fonctionne de manière autonome. Des anomalies, dues à des pannes d'équipement ou à une surcharge temporaire, peuvent intervenir. Pour réagir correctement à ces défaillances (qu'il faut déjà connaître), le protocole de diagnostic ICMP (RFC 590) s'occupe de la transmission des messages de contrôle. Chaque équipement surveille son environnement et signale les événements anormaux.

Nous avons vu que pour contrôler le trafic dans le réseau, un champ, dans l'en-tête du datagramme, indique la *durée maximale de séjour* dans l'interconnexion ; chaque routeur traitant le datagramme décrémente la durée de vie. Lorsqu'elle vaut zéro, le datagramme est détruit et le routeur envoie un message d'erreur à l'émetteur du datagramme.

ICMP est donc un mécanisme de contrôle des erreurs au niveau IP[1]. Initialement prévu pour permettre aux routeurs d'informer les utilisateurs des erreurs de transmission, il n'est pas restreint à cette fonction puisque des échanges entre utilisateurs sont tout à fait possibles.

Chaque message ICMP traverse le réseau en tant que données d'un datagramme IP. La conséquence directe est que les messages ICMP sont traités comme les autres datagrammes à travers le réseau. On comprend donc mieux la nécessité d'indiquer, dans son entête, le contenu du datagramme IP par le champ *Protocole*, bien qu'ICMP ne soit pas considéré comme un protocole de niveau plus élevé que IP. En fait, si les messages doivent traverser plusieurs réseaux avant d'arriver à leur destination finale, IP est le seul protocole commun à l'interconnexion.

1. Habituellement, les protocoles contiennent eux-mêmes les mécanismes nécessaires pour signaler les erreurs qui sont de leur ressort. IP fait exception, et le protocole ICMP a été défini pour les situations d'anomalies.

Chaque message ICMP possède un type particulier pour caractériser le problème qu'il signale. L'en-tête ICMP sur 32 bits contient le *type* (code de l'erreur sur 8 bits), un champ d'information complémentaire selon le type (sur 8 bits) et un bloc de contrôle d'erreur sur 16 bits utilisant le même mécanisme de vérification que pour les datagrammes IP. De plus, les messages ICMP transportent des données qui sont en fait le début du datagramme à l'origine du problème.

L'utilitaire *ping* crée un message ICMP de type 8 (*Echo Request*) que la machine envoie à l'adresse IP spécifiée pour tester si ce dernier est opérationnel. Si tel est le cas, le destinataire répond par un message ICMP de type 0 (*Echo Reply*), en renvoyant les données contenues dans le message émis.

L'utilitaire *traceroute* exploite, quant à lui, les messages ICMP de type 11 (*Time Exceeded*), en envoyant des datagrammes IP dont le champ TTL est délibérément trop petit ; ces datagrammes sont écartés par l'un des routeurs de l'interconnexion. L'événement donne lieu à message d'erreur ICMP retourné à l'expéditeur.

Quand un routeur ne peut pas délivrer un datagramme, il envoie un message ICMP de type 3 (*Destination unreachable*) à l'émetteur. Dans ce cas, le champ d'information complémentaire précise si c'est le réseau, la machine, le protocole ou le port qui est inaccessible. Le même message ICMP de type 3 est utilisé lorsqu'un routeur doit fragmenter un datagramme et que celui-ci porte le bit DF à 1 (le champ d'information complémentaire spécifie le cas).

5 Protocole IPv6 (IP version 6)

La croissance exponentielle du nombre d'ordinateurs connectés à Internet pose de nombreux problèmes. Le plan d'adressage IP atteint un seuil de saturation, les adresses disponibles commencent à manquer. Par ailleurs, le protocole IP dans sa version 4, présente plusieurs défauts : nécessité de recalculer le bloc de contrôle de l'en-tête dans chaque routeur, de configurer les machines avec une adresse IP, un masque de sous-réseau et une route par défaut, sans parler de l'absence de sécurité : il n'y a aucun service pour assurer la confidentialité des données transmises, pour authentifier les adresses utilisées… Pour terminer, IPv4 est incapable de traiter de façon satisfaisante des flux audio ou vidéo ou des flux à contraintes temporelles fortes comme les jeux interactifs. Toutes ces raisons ont motivé le développement d'une nouvelle version d'IP, appelée IPv6, qui prévoit un nouveau plan d'adressage, un format différent pour le datagramme, la notion de qualité de service et des mécanismes de sécurité.

5.1 Plan d'adressage

IPv6 prévoit des adresses sur 128 bits, ce qui est gigantesque : on pourrait utiliser plusieurs millions d'adresses par m^2 sur terre, y compris dans les océans ! Les types d'adresses sont globalement conservés, sauf les adresses de diffusion (*broadcast*) qui sont remplacées par une généralisation du *multicast* (adressage multipoint).

On ne parle plus de classes d'adresses, et de nombreux nouveaux types, déterminés par un préfixe, existent. Le préfixe `0000 0000` binaire est utilisé pour la compatibilité avec les adresses IP classiques. L'adressage IPv6 résout non seulement le problème de la saturation des adresses mais il offre, en plus, de nouvelles possibilités comme la hiérarchisation à plusieurs niveaux ou l'encapsulation d'adresses déjà existantes, ce qui facilite leur résolution.

5.2 Nouveau format et qualité de service

IPv6 utilise un format de datagramme incompatible avec IP classique. Il se caractérise par un en-tête de base de taille fixe et plusieurs en-têtes d'extension optionnels suivis des données. Ce format garantit une souplesse d'utilisation et une simplicité de l'en-tête de base. Seize niveaux de priorité sont définis et respectés par les routeurs : le traitement des applications interactives et des transferts de fichiers vidéo peuvent alors être différents. Un identificateur de *flot* relie les datagrammes d'une même connexion applicative afin de leur garantir la même qualité de service[2]. L'utilisation combinée de la priorité et de l'identificateur de flot permet d'ajuster la qualité de service offerte par le routage aux besoins de l'application. Elle répond donc à la demande des nouvelles applications (temps réel, multimédia…).

Le nombre de routeurs que peut traverser le datagramme avant d'être détruit remplace le champ Durée de vie d'IPv4. Sa gestion est plus simple. La fragmentation est désormais traitée de bout en bout : l'algorithme PMTU (*Path Maximum Transfer Unit*) détermine la taille maximale des datagrammes sur le chemin prévu. Les paquets sont ensuite fragmentés par la source et réassemblés par le destinataire.

Grâce à l'utilisation d'en-têtes optionnels, le routeur n'a qu'à extraire l'en-tête de base ainsi que l'en-tête optionnel *hop by hop* (littéralement saut par saut) qui suit l'en-tête de base et qui contient des options devant être traitées dans les nœuds intermédiaires.

Avec le développement des portables, il est intéressant de pouvoir rediriger les messages adressés à la station fixe habituelle vers sa localisation actuelle en cas de déplacement. Cela se fait désormais au niveau du protocole IPv6 (et non au niveau des protocoles de couches supérieures, comme c'est le cas avec la redirection des courriers électroniques, par exemple). Un redirecteur, placé à l'entrée du réseau, connaît l'adresse IPv6 de la personne en déplacement. Il encapsule le datagramme dans un nouveau datagramme IPv6 et l'expédie à la nouvelle adresse. Le destinataire peut ainsi connaître l'identité de l'émetteur.

Lorsqu'il existe des contraintes de délai et de débit (temps réel), les routeurs mettent également en œuvre un mécanisme de réservation de ressources adapté aux exigences stipulées dans les champs priorité et identificateur de flot des datagrammes. Enfin, IPv6 implémente des éléments d'authentification et de confidentialité, thèmes qui n'étaient pas abordés dans IP, mais seulement dans la version IPSec utilisée pour créer des réseaux privés virtuels.

Résumé

Le protocole IP (*Internet Protocol*) est implémenté dans toutes les machines hôtes d'Internet ainsi que dans tous les routeurs. Il assure un service de remise non fiable sans connexion. Il comprend la définition du plan d'adressage, la structure de l'unité de données transférée (le *datagramme IP*) et des règles de routage. Les datagrammes sont transmis à travers l'interconnexion, au coup par coup, indépendamment les uns des autres. IP inclut un protocole ICMP de génération de messages d'erreur en cas de destruction de datagrammes, de problèmes d'acheminement ou de remise. La saturation du plan d'adressage, l'absence de qualité de service et de sécurité ont conduit à de nombreuses études dont est issu IPv6, la nouvelle version du protocole IP.

2. Mais IPv6 reste un protocole sans connexion !