Architecture TCP/IP

par **Guy PUJOLLE**
Professeur à l'Université de Versailles

*TCP/IP est un sigle qui recouvre deux protocoles utilisés par de nombreuses sociétés commercialisant des équipements de réseau. Ces deux protocoles IP (**Internet Protocol**) et TCP (**Transmission Control Protocol**) forment respectivement la couche réseau et la couche transport qui ont été développées pour les besoins d'interconnexion des divers réseaux hétérogènes de la défense américaine. L'idée de base est simple, rendre ces réseaux homogènes en leur imposant un protocole commun, le protocole IP. De cette façon, pour passer d'un sous-réseau à un autre sous-réseau, il faut passer par le protocole IP qui gère le routage.*

Dans les faits, ce sigle TCP/IP représente beaucoup plus que les deux protocoles développés pour interconnecter des sous-réseaux entre eux ; il désigne tout un environnement qui contient, bien sûr, les protocoles TCP et IP mais aussi les applications qui ont été développées au-dessus de ces deux protocoles : la messagerie électronique dénommée SMTP (Simple Mail Transport Protocol), le transfert de fichiers FTP (File Transfer Protocol), l'accès à des bases d'informations WWW (World Wib Web), etc.

Le succès de cet environnement provient au départ de son utilisation dans le réseau Internet et, pour bien en comprendre les fondements, il faut revenir aux structures de base de ce réseau Internet.

Internet *est un réseau de réseaux développé par le ministère de la Défense aux États-Unis dans la fin des années 70 et le début des années 80 pour interconnecter les différentes machines informatiques de ce ministère. La solution a été de développer un protocole commun que l'ensemble des réseaux et des machines connectées doit posséder. Ce protocole commun, c'est précisément le protocole IP (Internet Protocol). Les réseaux interconnectés, que nous avons appelés les sous-réseaux, peuvent être quelconques. Il leur est juste demandé de transporter d'une extrémité à l'autre des paquets IP, c'est-à-dire des paquets conformes aux spécifications du protocole IP. Ces sous-réseaux peuvent aussi bien être du type X.25, Ethernet, relais de trames, qu'ATM, etc. Le protocole IP représente le protocole de base obligatoire dans l'environnement Internet.*

Le réseau Internet a été créé pour transporter des données informatiques, et les sous-réseaux sont de type divers et utilisent classiquement une commutation de paquets adaptés aux applications asynchrones.

Comme nous le verrons, le protocole IP est très simple, au moins dans sa première version, et utilise une technique de routage. En d'autres termes, les paquets d'un même utilisateur sont indépendants les uns des autres et sont routés par les nœuds gérant le protocole IP. De ce fait, deux paquets du même utilisateur peuvent prendre des chemins différents.

L'évolution des réseaux utilisant les protocoles TCP et IP est dictée par la perspective de transporter des applications multimédias et non plus uniquement des données informatiques. Pour y arriver, les protocoles de la première génération devront être remplacés par ceux d'une nouvelle génération, capables de prendre en charge la synchronisation indispensable pour acheminer des applications isochrones comme la parole ou les applications vidéo.

Une autre particularité de l'ensemble TCP/IP est de bien représenter les protocoles des réseaux informatiques. Ils sont simples avec pour but de mettre en place une structure dans laquelle l'intelligence est au niveau des extrémités, dans l'équipement terminal. Ce type de réseau ne peut empêcher l'entrée d'un nouveau client mais, au contraire, favorise une nouvelle arrivée en lui faisant une place au côté de tous ceux déjà présents. Les ressources du réseau qui étaient divisées entres N clients vont être divisées entre N + 1 clients. Si l'un des clients n'a plus les ressources réseaux nécessaires pour maintenir la qualité de transport de son application, il devra la dégrader pour se mettre au niveau de ce que le réseau laisse passer. Cette solution se modifie et la deuxième génération des protocoles TCP/IP prendra beaucoup plus en compte des possibilités de réservation pour essayer de maintenir la qualité de service.

Le premier élément qui permettra cette évolution vers une garantie de la qualité de service, se trouve dans la puissance des machines terminales qui ne fait que croître. De plus, ces évolutions des protocoles TCP/IP vers le multimédia vont se retrouver dans les réseaux ***Intranet****, c'est-à-dire les réseaux à base de l'environnement TCP/IP mis en place pour gérer et accéder aux informations de l'entreprise.*

Si l'on regarde plus spécifiquement l'utilisation des protocoles TCP/IP pour réaliser le réseau Internet, on découvre que la structure en sous-réseaux porte en elle un défaut important : un manque de contrôle de l'ensemble puisque tous les sous-réseaux sont indépendants. Le comportement global est imprévisible puisqu'il n'y a aucun opérateur capable d'avoir une vue synthétique du réseau. Un Intranet est un réseau qui va utiliser les protocoles du réseau Internet mais dans un environnement privé. Le gestionnaire de ce réseau privé peut, de ce fait, mettre des bornes aux entrées et refuser un nouveau client qui dégraderait trop sensiblement les utilisateurs déjà présents.

La parole représente la première application à prendre en charge pour aller vers le multimédia. Cette application est isochrone et il faut remettre les octets de parole à des instants extrêmement précis. Pour cela, on se sert d'un temps de latence qui peut atteindre 400 ms. Ce temps est beaucoup plus important que celui qui a servi de base pour le dimensionnement de l'ATM. En effet, dans ce dernier cas, on doit éviter les échos provenant des extrémités analogiques. Dans Internet, les terminaux, qui ne sont autres que des PC ou des postes de travail,

sont numériques. La chaîne étant toute numérique, il suffit de ne pas dépasser un temps acceptable de l'ordre de 300 à 400 ms pour que la conversation reste compréhensible en temps réel.

Le délai de traversée d'un réseau TCP/IP est variable mais si l'on suppose l'existence d'une borne maximale, sauf pour un très petit nombre de paquets que l'on peut qualifier de négligeable, la solution retenue est de remettre au récepteur les paquets après ce temps de latence ; cela permet de retrouver la synchronisation.

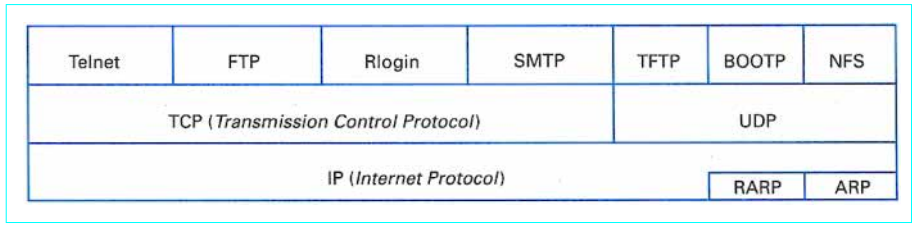
Il en est de même pour les protocoles de transfert de canaux de visioconférence et des autres applications isochrones qui nécessitent une resynchronisation forte.

Cette première approche est, cependant, insuffisante dans beaucoup de cas car il faut effectivement assurer une traversée du réseau qui ne dépasse pas une valeur déterminée. Si le réseau est légèrement congestionné, les temps de traversée peuvent dépasser les bornes fixées au départ. Les réseaux ATM permettent d'atteindre cet objectif par une qualité de service qui est déterminée lors de la mise en place de la connexion par une réservation de ressources. L'environnement Internet s'oriente, comme nous le verrons, vers le même type de procédures.

*L'architecture TCP/IP est constituée de trois couches, le niveau réseau de la hiérarchie du modèle de référence avec comme protocole principal IP (**Internet Protocol**), le niveau transport avec les protocoles TCP (**Transmission Control Protocol**) et UDP (**User Data Protocol**). Ces deux protocoles contiennent les éléments de la couche session du modèle de référence. Enfin, la dernière couche regroupe les couches présentation et application du modèle de référence.*

*Le protocole IP est un protocole très simple qui a pour but de transporter des paquets, que nous appellerons **datagrammes**, d'une porte d'entrée du réseau à une porte de sortie. C'est une couche dans un mode sans connexion, c'est-à-dire qu'un émetteur peut envoyer des datagrammes sans au préalable avertir l'entité correspondante de l'autre côté du réseau. La version actuelle, celle utilisée dans le réseau Internet, est IPv4 (IP version 4). Une nouvelle version, IPv6, prendra bientôt sa place.*

Avant même de détailler les différents protocoles de cette architecture, nous allons introduire un des concepts clefs d'Internet : le routage et donc l'adressage.



Architecture TCP/IP

1. Adressage IPv4

Les machines travaillant sous le protocole IP possèdent une adresse tenant sur 32 bits. Cette adresse est souvent représentée par une suite de quatre nombres séparés par des points ; par exemple 191.92.34.223.

L'adresse est constituée de deux parties : un identificateur de réseau et un identificateur de la machine à l'intérieur de ce réseau. L'identificateur de réseau est précédé par un numéro de classe de réseau. Il existe quatre classes d'adresses, chacune permettant de coder un nombre différent de réseaux et de machines :

— classe A – 128 réseaux (codés sur 7 bits) et 16 777 216 hôtes (codés sur 24 bits) ;
— classe B – 16 384 réseaux (codés sur 14 bits) et 65 535 hôtes (codés sur 16 bits) ;
— classe C – 2 097 152 réseaux (codés sur 21 bits) et 256 hôtes (codés sur 8 bits) ;
— classe D – adresses de groupe (codés sur 28 bits).

La figure 1 indique ces quatre classes.

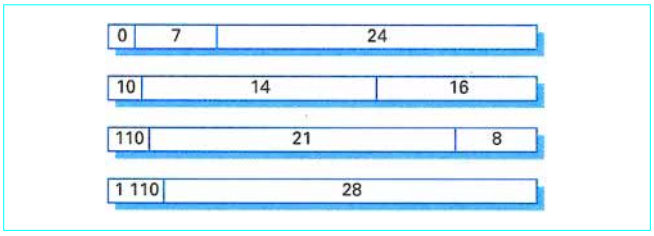


Figure 1 – Les zones d'adresse

Les adresses IP ont été définies pour être traitées rapidement. Les routeurs qui effectuent le routage en se basant sur le numéro de réseau, sont dépendants de cette structure. Un hôte relié à plusieurs réseaux aura plusieurs adresses IP. En fait, une adresse n'identifie pas simplement une machine mais une connexion à un réseau.

Pour assurer l'unicité des numéros de réseaux, les adresses IP sont attribuées par un organisme central, le NIC (*Network Information Center*). On peut également définir ses propres adresses si l'on n'est pas connecté à l'Internet. Mais il est vivement conseillé d'obtenir une adresse officielle pour garantir l'interopérabilité dans le futur.

Une des difficultés majeures que doit affronter l'Internet est l'épuisement des adresses et en particulier des adresses de classe A et B. Une solution, définie en 1992 et appelée CDR (*Classless Internet Domain Routing*), consiste à donner aux entreprises non plus un réseau de classe B, mais plusieurs réseaux de classe C, avec pour principal inconvénient une gestion plus complexe du routage du fait de la multiplication des réseaux de petite taille. La technique adoptée pour résoudre ce problème est l'agrégation des tables de routage, qui permet de regrouper un ensemble de réseaux sous la même adresse. Le coût d'utilisation de plusieurs classes C est alors identique à celui d'un réseau de classe B en termes de complexification du routage.

La détection par un routeur de l'appartenance de deux machines au même sous-réseau s'effectue par une technique de masque. Le masque d'un sous-réseau possède la forme d'une adresse IP avec une suite de bits, tous à 1, suivie d'une autre suite de bits tous à zéro. Lorsque l'on restreint l'adresse de deux terminaux aux bits correspondant aux 1 d'un masque, et que l'on obtient la même adresse, c'est que les deux machines appartiennent au même sous-réseau.

Une autre solution pour pallier le problème de manque d'adresse consistait à définir un nouveau protocole IP. C'est finalement le moyen qui a été choisi et que nous verrons, dans le paragraphe dévolu à cette nouvelle version du protocole IP appelée IPv6 ; dans laquelle la longueur des adresses demande un champ de 16 octets. Nous étudierons également le DNS (*Domain Name Service*) qui permet de substituer à l'adresse exprimée en chiffres décimaux, une adresse qui s'écrit avec des lettres et qui est beaucoup plus simple à retenir.

2. ARP, RARP : les protocoles de résolution des adresses

Les adresses IP sont attribuées indépendamment des adresses matérielles des machines. Pour envoyer un datagramme dans l'Internet, le logiciel réseau doit convertir l'adresse IP en une adresse physique qui est utilisée pour transmettre la trame. Si l'adresse physique est un entier court, elle peut être facilement modifiée pour lui faire correspondre l'adresse machine IP. Sinon, la traduction doit être effectuée dynamiquement.

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

Au moment de son initialisation (*bootstrap*), une machine sans mémoire de masse (*diskless*) doit contacter son serveur pour déterminer son adresse IP, afin de pouvoir utiliser les services TCP/IP. Le protocole RARP (*Reverse ARP*) permet à une machine d'utiliser son adresse physique pour déterminer son adresse logique dans l'Internet. Le mécanisme RARP permet à un calculateur de se faire identifier comme cible en diffusant sur le réseau une requête RARP. Les serveurs recevant le message examinent leur table et répondent au client. Une fois l'adresse IP obtenue, la machine la stocke en mémoire vive et n'utilise plus RARP jusqu'à ce qu'elle soit réinitialisée.

3. IPv4 : Internet Protocol version 4

Un réseau Internet est vu de l'utilisateur comme un réseau virtuel unique qui interconnecte toutes les machines et au travers duquel on peut communiquer. L'architecture sous-jacente est à la fois cachée et hors de propos. Un réseau Internet est une abstraction d'un réseau physique, car, à son niveau le plus bas, il fournit les mêmes fonctions, comme accepter des paquets ou les remettre au destinataire.

Le service rendu par le protocole IPv4 est déterminé par un système de remise de paquets, non fiable, « au mieux » et sans connexion. Le service est dit non fiable car la remise n'est pas garantie. Un paquet peut être perdu, dupliqué, ou remis hors séquence, mais le protocole IP ne détectera rien et n'en informera ni l'émetteur, ni le récepteur. Il est dit sans connexion car chaque paquet est traité indépendamment des autres. Les paquets d'un même message, transitant d'une machine à une autre, peuvent utiliser des routes différentes et certains peuvent être perdus, les autres arrivant à leur destination.

Le protocole IP définit l'unité de donnée de protocole de base et le format exact de toutes les données qui transitent dans le réseau. IP inclut également un ensemble de règles qui définissent comment traiter les paquets et les cas d'erreurs et effectue la fonction de routage.

La version utilisée actuellement est la version 4 dénommée IPv4. La structure de la trame est décrite dans la figure **2**.

Il y a une analogie entre un réseau physique et l'Internet. Dans un réseau, l'unité transférée entre deux nœuds est la trame qui contient un en-tête et des données. L'en-tête contient des informations comme l'adresse source et destinataire. Dans l'Internet, l'unité de base à transférer est le **datagramme Internet**, souvent appelé datagramme IP, ou paquet IP, ou simplement datagramme. Le datagramme est également divisé en un en-tête et une partie données.

Contrairement aux trames, les datagrammes sont manipulés par le logiciel. Ils peuvent être de longueur quelconque. Cependant, comme ils doivent transiter de machine en machine, ils sont toujours transportés dans des trames physiques.

Ce concept est appelé **l'encapsulation**. Pour le sous-réseau, un datagramme est une entité comme une autre. Dans le meilleur des

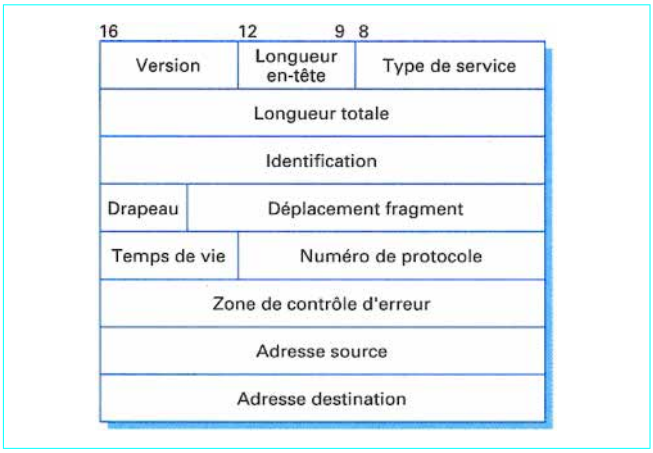


Figure 2 – Le datagramme IP

cas, le datagramme est contenu dans une seule trame, ce qui rend la transmission plus performante.

La figure **3** décrit ce processus d'encapsulation et de décapsulation.

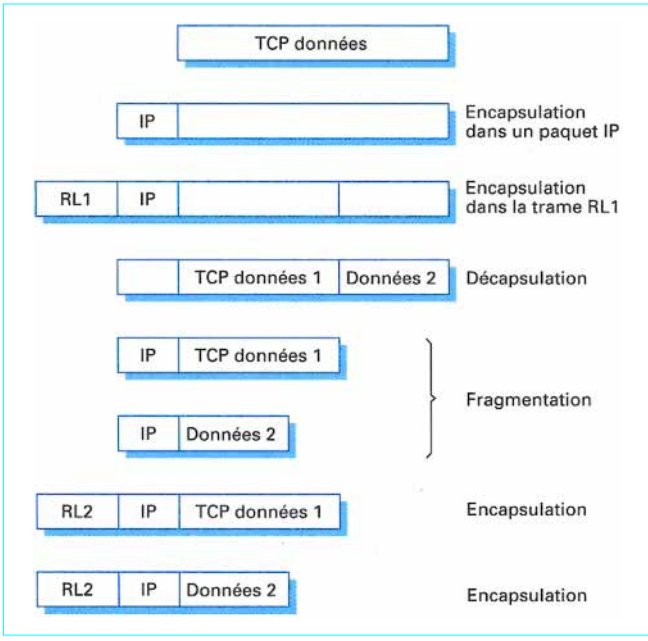


Figure 3 – Encapsulation et décapsulation

Le but de l'environnement Internet est justement de cacher les couches inférieures des réseaux. Aussi, au lieu de prévoir la taille des datagrammes en fonction des contraintes des sous-réseaux, choisira-t-on une taille convenable pour les datagrammes, et prévoira-t-on une façon de les découper en fragments pour qu'ils puissent être transportés dans des petites trames, puis réassemblés. L'Internet ne limite pas les datagrammes à une taille précise, mais suggère que les réseaux et les passerelles puissent supporter ceux de 576 octets sans les fragmenter.

Fragmenter un datagramme revient à le diviser en plusieurs morceaux. Chaque morceau a le même format que le datagramme d'origine. Chaque nouveau fragment a un en-tête, qui reprend la plupart des informations de l'en-tête d'origine, et le plus de données possible, sachant que le fragment doit tenir une seule trame.

Dans l'Internet, dès qu'un datagramme a été fragmenté, les fragments sont transmis indépendamment les uns des autres jusqu'à leur destination, où ils doivent être réassemblés. Si l'un des fragments est perdu, le datagramme ne peut pas être récupéré complètement et les autres fragments doivent être éventuellement détruits sans être traités.

Les faiblesses d'IPv4 concernent d'abord l'adressage qui est limité par les quatre octets disponibles. En fait, la distribution des adresses n'a pas été faite avec suffisamment de soin et de nombreuses adresses A et surtout B sont excessivement mal utilisées. Le second problème concerne l'arrivée d'applications multimédias qui contiennent des synchronismes forts comme celui de la parole. Dans la version IPv4, il est impossible de discerner, dans la zone d'information du paquet, une application qui possède des contraintes par rapport à une application qui n'a pas de contraintes particulières. Il n'y a pas non plus de possibilité de faire transiter de la signalisation ou de l'information de gestion.

Une autre lacune importante concerne la sécurité de la communication dans le réseau.

Nous allons voir que le nouveau protocole IPv6 va résoudre ces problèmes.

4. IPv6 : Internet Protocol version 6

Le protocole IPv6 représente la nouvelle génération du protocole IP. Les fonctionnalités ont été entièrement repensées et le protocole IPv6 forme réellement une nouvelle génération, d'où le nom IPng (next generation) qu'on lui donne également.

Le format du paquet IPv6 est décrit dans la figure **4**.

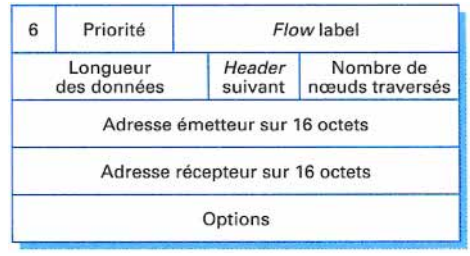


Figure 4 – Format du paquet IPv6

Le premier champ porte le numéro de version (6 pour IPv6). Le champ suivant indique un niveau de priorité permettant un traitement plus ou moins prioritaire dans les nœuds du réseau. Les principales valeurs sont les suivantes :

0 pas de priorité particulière ;
1 trafic de base (news) ;
2 transfert de données sans contrainte temporelle (email) ;
3 réservé pour le futur ;
4 transfert en blocs avec attente du récepteur (transfert de fichiers) ;
5 réservé pour le futur ;
6 traficinteractif (rlogin, terminal virtuel) ;
7 trafic pour le contrôle (routage, contrôle de flux).

Le champ suivant permet d'indiquer la qualité de service des informations transportées dans le paquet IPv6. Cette indication permet aux routeurs de prendre des décisions adaptées aux données transportées ; des algorithmes d'ordonnancement des trames pourront être implantés dans les routeurs. Le champ indiquant la longueur des données précise la longueur totale du datagramme en octets (sans tenir compte de l'en-tête). Ce champ étant de 2 octets, la longueur maximale est de 64 Ko.

Le champ suivant identifie le protocole qui sera utilisé à l'intérieur du champ de données. Les options sont les suivantes :

0 Hop-by-Hop Option Header ;
4 IP ;
6 TCP ;
17 UDP ;
43 Routing Header ;
44 Fragment Header ;
45 Interdomain Routing Protocol ;
46 Resource Reservation Protocol (RSVP) ;
50 Encapsulating Security Payload ;
51 Authentification Header ;
58 ICMP ;
59 No Next Header ;
60 Destination Options Header.

La limite du nombre de nœuds à traverser indique le nombre maximal de nœuds traversés par le paquet avant que celui-ci soit détruit.

L'adresse IPv6 tient sur 16 octets au lieu des 4 de la première génération. La difficulté réside dans la représentation et l'utilisation rationnelle de ces 128 bits. La représentation s'effectue par groupe de 16 bits sous la forme :

123 : FCBA : 1 024 : AB23 : 0 : 0 : 24 : FEDC

Une série d'adresses égales à 0 peut être abrégée par le signe :: qui ne peut apparaître qu'une seule fois dans l'adresse. En effet, il faut pouvoir en déduire le nombre d'adresses 0 en série et si deux séries de 0 existaient, il ne serait plus possible d'en déduire la longueur de chacune. L'adressage IPv6 constitue un adressage hiérarchique avec beaucoup plus de niveaux que les trois disponibles dans IPv4. Un avantage immédiat sera de réduire la taille des tables de routage des routeurs et donc d'augmenter le temps de recherche des informations pour effectuer la procédure de routage.

La zone complémentaire permet l'ajout de fonctions supplémentaires qui sont optionnelles. La figure 5 donne un exemple du processus d'extension.

Figure 5 – Un champ d'extension avec quatre options

Chaque zone d'extension commence par un champ indiquant, par un numéro, le type d'extension. Nous avons les options suivantes, qui ont déjà pu être utilisées dans la partie « en-tête suivante » :

0 Hop-by-Hop Option Header ;
43 Routing Header ;
44 Fragment Header ;
51 Authentification Header ;
59 No Next Header ;
60 Destination Options Header.

5. Routage IP

Un environnement Internet résulte de l'interconnexion de réseaux physiques par des routeurs dont le nom indique bien la fonction : à partir de l'adresse contenue dans le datagramme, le routeur détermine le meilleur chemin à l'instant du routage grâce à une table de routage qui se met à jour automatiquement. Chaque routeur est connecté directement à deux ou plusieurs réseaux, les hôtes pouvant également être connectés à un ou plusieurs réseaux. La figure 6 représente schématiquement le passage d'un sous-réseau à un autre sous-réseau.

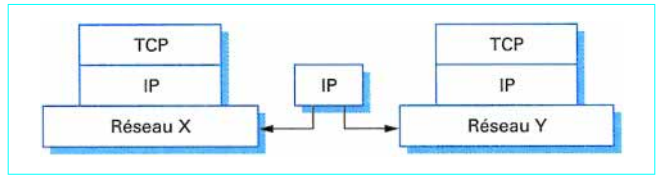


Figure 6 – Utilisation du protocole IP dans un routeur d'Interconnexion

Il y a un routage direct si les deux machines qui veulent communiquer sont rattachées au même réseau (elles ont donc le même numéro de réseau IP). Ce peut être le cas entre deux hôtes ou d'un routeur vers un hôte. Il suffit donc de déterminer l'adresse physique du destinataire et d'encapsuler le datagramme dans une trame avant de l'envoyer sur le réseau.

Dans le cas où les deux machines ne se trouvent pas sur le même réseau, il y a un routage indirect. Celui-ci est beaucoup plus complexe, car il faut déterminer le routeur auquel les datagrammes doivent être envoyés. Les datagrammes peuvent ainsi être transmis de routeur en routeur jusqu'à ce qu'ils atteignent l'hôte destinataire.

Le routage est effectué à partir du numéro de réseau de l'adresse IP de l'hôte de destination. La table contient, pour chaque numéro de réseau à atteindre, l'adresse IP du routeur auquel il faut envoyer le datagramme. Elle peut également contenir une adresse de routeur par défaut et l'indication de routage direct. La difficulté du routage provient de l'initialisation et de la mise à jour des tables de routage.

5.1 RIP *(Routing Information Protocol)*

C'est le protocole le plus utilisé dans l'environnement TCP/IP, pour router les paquets entre les passerelles du réseau Internet. C'est un protocole IGP (*Interior Gateway Protocols*) qui utilise un algorithme permettant de trouver le chemin le plus court. Par « chemin », on entend ici le nombre de nœuds traversés qui doit être compris entre 1 et 15. La valeur 16 indique une impossibilité. En d'autres termes,

si le nombre de nœuds traversés sur le chemin pour aller d'un point à un autre du réseau Internet est supérieur à 15, la connexion ne pourra pas être mise en place. Les messages RIP permettant de dresser les tables de routage sont envoyés approximativement toutes les 30 s. Si un message RIP n'est pas arrivé à son voisin au bout de 3 min, ce dernier considère que la liaison n'est plus valide, c'est-à-dire que le nombre de liens est supérieur à 15. Le protocole RIP est basé sur une diffusion périodique des états du réseau d'une passerelle vers ses voisines. Une version RIP2 améliore la version RIP1 sur plusieurs points : routage par sous-réseau, authentification des messages, transmission multipoint, etc.

5.2 OSPF *(Open Shortest Path First)*

OSPF fait partie d'une deuxième génération de protocoles de routage. Il est beaucoup plus complexe que RIP mais ses performances sont supérieures. Le protocole OSPF utilise une base de données distribuée qui garde en mémoire l'état des liaisons. Ces informations forment une description de la topologie du réseau et de l'état des nœuds. Ces informations permettent de définir le routage à introduire dans la table, par un calcul des chemins les plus courts. Cinq types de liaisons sont définis : les liaisons à partir d'un routeur, les liaisons du réseau de transit, les récapitulations de réseaux IP qui partent des routeurs interzones, les récapitulations de routeurs externes qui atteignent les routeurs interzones et les liaisons externes. L'algorithme OSPF est mis en œuvre à partir des bases de données. Il permet, à partir d'un nœud, de calculer le chemin le plus court avec les contraintes indiquées dans les contenus associés à chaque liaison. Les routeurs OSPF communiquent entre eux par l'intermédiaire du protocole OSPF qui est placé au-dessus d'IP.

La protocole OSPF et beaucoup plus complexe que le protocole RIP. Ce dernier est adapté à la gestion du routage dans de petits réseaux. Le routage OSPF s'applique à des réseaux beaucoup plus complexes d'interconnexion de sous-réseaux. Cependant, d'autres protocoles peuvent aussi être envisagés dans le contexte TCP/IP.

5.3 IGRP *(Interior Gateway Routing Protocol)*

Le protocole RIP de base s'est rapidement trouvé en défaut devant le nombre grandissant de nœuds dans les sous-réseaux. La compagnie Cisco, pour ses routeurs, à développé un nouveau protocole de routage dénommé IGRP. En effet, Cisco utilisait à ses débuts presque exclusivement le protocole RIP mais celui-ci ne pouvait plus supporter la masse d'informations à digérer pour réaliser la table de routage dans les grands réseaux. Cette compagnie fut amenée à mettre en place un routage plus performant, IGRP, qui n'est autre, en fait, qu'une version améliorée de RIP. Cette extension du routage RIP intègre le routage multichemins, la gestion des routages par défaut, la diffusion de l'Information toutes les 90 secondes au lieu de 30 secondes, la détection des bouclages, etc. Ce protocole a lui-même été étendu pour réaliser une meilleure protection contre les boucles : il s'agit du protocole EIGRP (Extended IGRP).

D'autres protocoles plus spécialisés ont été développés dans le cadre du réseau Internet et de l'environnement TCP/IP : en particulier, les protocoles EGP et BGP que nous allons introduire brièvement.

5.4 EGP *(Exterior Gateway Protocol)*

Le réseau Internet s'étant tellement étendu, la communauté Internet a décidé de le scinder en systèmes autonomes pour en faciliter la gestion. Pour router un paquet d'un système vers un autre, il a fallu développer, au début des années 80, un protocole de routage spécifique. Le premier fut EGP : ce protocole est composé essentiellement de trois procédures qui permettent la mise en place de l'échange d'information entre systèmes autonomes. La première procédure concerne la définition d'une passerelle voisine. Celle-ci étant connue, les deux voisins déterminent la liaison qui va leur permettre de communiquer. La troisième procédure concerne alors l'échange de paquets entre les deux voisins sur la liaison entre systèmes autonomes. Les faiblesses d'EGP sont apparues avec le développement exponentiel d'Internet et le besoin d'éviter des routeurs situés dans des zones sensibles politiquement.

5.5 BGP *(Border Gateway Protocol)*

Pour répondre aux faiblesses d'EGP, un nouveau protocole fut mis en chantier par l'IETF sous le nom de BGP. Une première version, BGP-1, fut implantée en 1990 et suivie de peu par une deuxième version, BGP-2, puis une troisième, BGP-3. Après une utilisation de quelques années, une quatrième version a été déployée, EGP-4, qui permet de gérer beaucoup plus efficacement les tables de routage de grande dimension en rassemblant en une seule ligne plusieurs sous-réseaux.

BGP apporte de nouvelles propriétés par rapport à EGP, en particulier, celle de gérer les boucles qui devenaient courantes dans EGP puisque ce protocole ne s'occupait que des couples de voisins sans prendre en compte les rebouclages possibles par un troisième réseau autonome.

L'extension du réseau Internet introduit de nouveaux problèmes par l'explosion des tables de routage. Pour revenir à des tailles raisonnables de façon à assurer un traitement plus court de l'information de routage, un nouveau protocole a été développé par l'IETF : IDRP.

5.6 Routage IDRP *(Interdomain Routing Protocol)*

L'extension extrêmement rapide de l'Internet posait le problème de la pérennité des mécanismes de routage de base : ces mécanismes allaient-ils continuer à opérer avec des configurations multipliées par plusieurs milliers de fois ?

Le protocole IDRP (*Interdomain Routing Protocol*) a été conçu pour répondre à ces problèmes dans le cadre de l'environnement IPv6. Le protocole IDRP provient d'études qui ont été faites à l'ISO sur le routage entre les domaines de routage (routing domain). Ce protocole a été adapté au monde Internet pour réaliser le routage entre l'équivalent des domaines de routage qui s'appellent les systèmes autonomes (*Autonomous Systems*).

Le but d'IDRP est légèrement différent de celui des protocoles à l'intérieur d'un domaine. Il s'agit de définir une politique de routage entre systèmes autonomes et non pas seulement un algorithme de routage. La solution retenue pour ce routage consiste à définir une politique dans laquelle les routeurs d'un système autonome se mettent d'accord pour, par exemple, ne pas passer par un domaine prédéterminé à l'avance ou ne pas donner l'autorisation pour la traversée de son système autonome à d'autres systèmes autonomes. En d'autres termes, il doit y avoir une concertation entres routeurs pour ne donner que les indications correspondant à la politique définie.

Les algorithmes de routage de type OSPF ou RIP sont appliqués par des routeurs qui ont tous le même but et qui s'appuient sur des notions de poids. Le routage IDRP a aussi pour but de trouver les meilleurs chemins en tenant compte des restrictions de chaque système autonome. L'algorithme s'appuie sur des vecteurs de distance (*path vector routing*). Ces vecteurs de distance tiennent compte du chemin de bout en bout au lieu de ne tenir compte que des poids pour aller vers les nœuds voisins.

Comme le nombre de systèmes autonomes peut croître plus vite que les capacités de traitement des routeurs, il a été décidé de regrouper des systèmes autonomes en confédération. Le protocole IDRP travaille alors sur le routage entre ces confédérations.

Pour s'échanger l'information du routage, IDRP utilise des paquets spécifiques portés dans des paquets IP. Dans la zone IP, le prochain en-tête porte le numéro 45 et indique le protocole IDRP. IDRP est un protocole en mode connexion entre les routeurs. Il possède des fonctions de sécurité comme la garantie de l'intégrité des données et l'authentification. Enfin, IDRP possède de nombreuses autres fonctions comme un temps de vie de la connexion, la taille maximale d'un message IDRP, la liste des systèmes autonomes auxquels l'émetteur appartient, et la possibilité d'envoyer des données sur la connexion dans une zone d'option.

6. ICMP : messages de contrôle et d'erreur

Dans le système en mode non connecté d'Internet, chaque passerelle et chaque machine fonctionnent de façon autonome. Le routage et l'envoi des datagrammes se font sans coordination avec l'émetteur. Ce système fonctionne bien, tant que toutes les machines n'ont pas de problème et que le routage est correct, mais cela n'est pas toujours le cas.

En dehors des pannes du réseau et des équipements terminaux, les problèmes arrivent lorsqu'une machine est temporairement, ou de façon permanente déconnectée du réseau, ou lorsque la durée de vie du datagramme expire, ou enfin, lorsque la congestion d'une passerelle est trop importante.

Pour permettre aux machines de rendre compte de ces anomalies de fonctionnement, on a ajouté à Internet un protocole d'envoi de messages de contrôle appelé ICMP (*Internet Control Message Protocol*).

Le destinataire d'un message ICMP n'est pas un processus application, mais le logiciel Internet de la machine concernée. Quand un message est reçu, IP traite le problème.

Les messages ICMP ne sont pas uniquement transmis à partir des passerelles. En effet, n'importe quelle machine du réseau peut envoyer des messages à n'importe quelle autre. Cela permet d'avoir un protocole unique pour tous les messages de contrôle et d'information.

Chaque message a son propre format et permet de rendre compte de l'erreur jusqu'à l'émetteur du message. Les messages ICMP sont transportés dans la partie données des datagrammes IP. comme n'importe quel autre datagramme, ils peuvent être perdus. En cas d'erreur d'un datagramme contenant un message de contrôle, aucun message de rapport de l'erreur ne sera transmis, et cela, pour éviter les avalanches.

Les **principales informations de contrôle** qui peuvent être acheminées par ICMP sont les suivantes :

Code message	Type de message ICMP ;
0	Echo reply ;
3	Destination Unreachable ;
4	Source Quench ;
5	Redirect (change a route) ;
8	Echo Request ;
11	Time Exceeded for a datagram ;
12	Parameter Problem on a datagram ;
13	Timestamp Request ;
14	Timestamp reply ;
17	Address Mask Reply ;
18	Address Mask Reply.

7. IGMP : adressage multipoint

Le multipoint ou multicasting IP permet l'envoi de datagrammes vers plusieurs destinations de façon performante. Le protocole IP utilise les adresse de classe D pour indiquer qu'il s'agit d'un envoi multipoint et s'appuie sur le service réseau s'il existe.

Les groupes de diffusion sont dynamiques : une machine peut se rattacher ou quitter un groupe à tout moment, l'hôte devant seulement être capable d'émettre et de recevoir des datagrammes en multicast. Cette fonction IP n'est pas limitée au seul sous-réseau physique, mais les passerelles propagent aussi les informations d'appartenance à un groupe et gèrent le routage de façon à ce que chaque machine reçoive une copie de chaque datagramme envoyé au groupe.

Les machines communiquent aux passerelles leur appartenance à un groupe, en utilisant le protocole IGMP (*Internet Group Management Protocol*). Le protocol a été conçu pour être efficace et optimiser l'utilisation des ressources du réseau. Dans la plupart des cas, le trafic IGMP introduit est un message périodique envoyé par la passerelle gérant le multipoint, et une seule réponse pour chaque groupe de machines d'un sous-réseau.

IGMP a été développé pour le multicast avec le protocole IPv4. Pour le protocole IPv6, la gestion du multicast sera effectuée par le protocole ICMP que nous avons décrit sommairement dans le paragraphe précédent.

8. UDP : service de transport non fiable

Le protocole UDP (*User Datagram Protocol*) permet aux applications d'échanger des datagrammes. Ce protocole UDP utilise la notion de « port » qui permet de distinguer les différentes applications qui s'exécutent sur une machine. En plus du datagramme et de ses données, un message UDP contient, à la fois, un numéro de port source et un numéro de port destination.

UDP s'appuie sur les services du protocole Internet et fournit un service en mode non connecté, sans reprise sur erreur. Il n'utilise aucun acquittement, ne reséquence pas les messages et ne met en place aucun contrôle de flux. Les messages UDP peuvent être perdus, dupliqués, remis hors séquence ou arriver trop tard pour être traités en réception.

En conclusion, UDP fournit un service de transport le plus simple possible pour les applications. En l'absence de garantie, les applications ne perdent qu'un temps minimal aux traitements liés à l'interface avec la couche transport. Les applications qui ont besoin d'accéder rapidement à des données distantes pour lesquelles la garantie de transport peut être faible, trouveront dans UDP le protocole adéquat.

9. TCP : service de transport fiable

TCP (*Transport Control Protocol*) est un service de transport fiable ; pour arriver à cette fonctionnalité, TCP définit un certain nombre de caractéristiques :

— **flot d'octets** : les données échangées sont vues comme un flot de bits, divisé en octets et les octets sont reçus dans l'ordre où ils ont été envoyés ;

— **circuit virtuel en mode connecté** : le transfert des données ne peut commencer qu'après l'établissement d'une connexion entre les deux machines. Durant le transfert, les deux machines continuent à vérifier que les données sont transmises correctement. Le terme de circuit virtuel est employé, car les deux programmes d'application voient la connexion comme un circuit physique, la fiabilité de la transmission étant une illusion créée par le service de transport ;

— **transfert par paquet** : les programmes d'application envoient leurs données sur le circuit virtuel en les passant régulièrement au système d'exploitation de la machine. Chaque application choisit la taille de données qui lui convient, exprimée en nombre d'octets. Le protocole TCP est libre de découper les données en paquets de tailles différentes de ce qu'il a reçu de l'application. Pour rendre le transfert plus performant, le protocole TCP attend d'avoir suffisamment de données pour remplir un datagramme avant de l'envoyer sur le sous-réseau ;

— **flot de données non structurées** : le service de transport ne prend pas en compte les données structurées (cela est du ressort de l'application) ;

— **connexion duplex** : la connexion permet un transfert de données bidirectionnel. Ce sont deux flots de données inverses, sans interaction apparente. Il est possible de déterminer l'envoi dans un sens, sans arrêter l'autre sens. Ce principe permet de renvoyer des acquittements d'un sens de transmission, en même temps que les données de l'autre sens.

Le protocole TCP définit la structure des données et des acquittements échangés, et les mécanismes permettant de rendre le transport fiable. Il spécifie comment distinguer plusieurs connexions sur une même machine, et comment faire la détection et la correction, lors de la perte ou duplication de paquets. Il définit comment établir une connexion et comment la terminer.

Le protocole TCP permet à plusieurs programmes d'établir une connexion en même temps et démultiplexe les données reçues, provenant d'applications différentes. TCP utilise la notion abstraite de port qui identifie la destination ultime dans la machine.

TCP est donc un protocole en mode connecté qui n'a de sens qu'entre deux points d'extrémité de connexion. Pour cela, le programme d'une extrémité effectue une ouverture de connexion « passive » qui permet d'accepter une connexion entrante en lui affectant un numéro de port. L'autre programme d'application exécute une ouverture de connexion « active ». Une fois la connexion établie, le transfert de données peut commencer.

Le protocole TCP voit un flot de données comme un suite d'octets qu'il divise en segments. Généralement, chaque segment est transmis dans un seul datagramme IP.

TCP utilise un mécanisme de fenêtre pour réaliser une transmission performante par un contrôle de flux adapté aux caractéristiques de l'application et du réseau. Le mécanisme de fenêtre permet l'anticipation, c'est-à-dire l'envoi de plusieurs messages sans attendre d'acquittement. Cela permet d'éviter les congestions, si les fenêtres sont bien adaptées. La fenêtre permet également de réaliser un contrôle au niveau de la machine terminale, en autorisant le récepteur à limiter l'envoi des données s'il n'a pas la place nécessaire pour les recevoir dans ses mémoires.

Le mécanisme de fenêtre opère au niveau de l'octet et non pas du message. Les octets à transmettre sont numérotés séquentiellement, et l'émetteur gère trois pointeurs pour chaque fenêtre. De la même façon, le récepteur doit tenir à jour une fenêtre en réception. Pour une connexion, il est possible d'échanger des données indépendamment dans chaque sens, et chaque extrémité de connexion doit ainsi maintenir deux fenêtres, l'une en émission et l'autre en réception.

Une différence importante entre un mécanisme de fenêtre classique et celui employé par TCP provient de la taille de la fenêtre qui peut varier dans le temps. Chaque acquittement, spécifiant combien d'octets ont été reçus, contient une information de taille de fenêtre qui indique combien d'octets supplémentaires le récepteur est en mesure d'accepter. La taille de fenêtre peut être vue comme la taille libre des mémoires. Le récepteur ne peut réduire la fenêtre en deçà d'une valeur qu'il a déjà acceptée précédemment. En revanche, une taille de fenêtre plus petite peut accompagner un acquittement, de façon à ce qu'elle diminue en même temps qu'elle se déplace.

L'unité de protocole de TCP est appelée un **segment**. Ces segments sont échangés pour établir la connexion, pour transférer des données, pour les acquittements, pour modifier la taille de la fenêtre et enfin pour fermer une connexion. Les informations de contrôle de flux peuvent être transportées dans le flot de données inverses. Chaque segment est composé de deux parties : l'en-tête suivi des données. Nous avons représenté dans la figure 7 le format d'un segment.

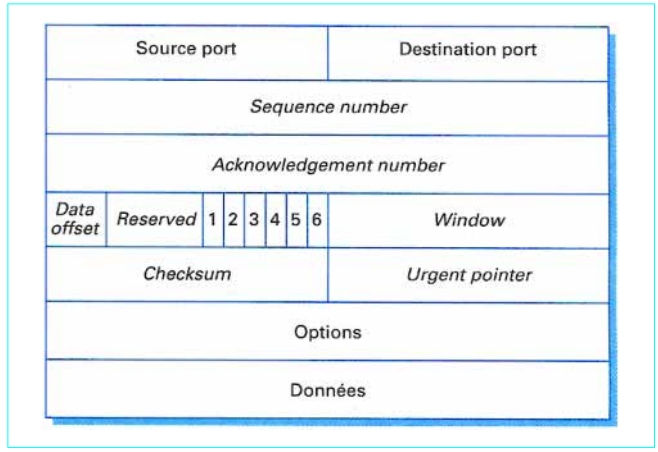


Figure 7 – Format d'un segment

Dans le segment, on trouve les zones suivantes :

1 – Source port sur 16 bits. Ce champ contient l'adresse du port d'entrée. Associée avec l'adresse IP, cette valeur donne un identificateur unique appelé *socket* ;

2 – Destination port sur 16 bits. Même chose que le précédent mais pour l'adresse destination ;

3 – Sequence number (SEQ) sur 32 bits. Ce champ indique le numéro du premier octet porté par le segment ;

4 – Acknowledgement number (ACK) sur 32 bits. Cette valeur indique le numéro « sequence number » du prochain segment attendu. En d'autres termes, c'est un acquittement de tous les octets qui ont été reçus auparavant ;

5 – Data offset sur 4 bits. Cette valeur indique la longueur de l'en-tête par un multiple de 32 bits. Si la valeur 8 se trouve dans ce champ, la longueur totale de l'en-tête est de 8×32 bits. Cette valeur est nécessaire parce que la zone d'option est de longueur variable ;

6 – La zone suivante est réservée pour une utilisation ultérieure. Ce champ doit être rempli de 0 ;

7 – Champ numéroté 1 sur la figure, Urgent Pointer (URG) sur 1 bit. Si ce bit est positionné à 1, cela indique que le champ Urgent Pointer dans la suite est utilisé ;

8 – Champ numéroté 2 sur la figure, Synchronisation (SYN) sur 1 bit. Si SYN = 1, cela indique une demande d'ouverture de connexion ;

9 – Champ numéroté 3 sur la figure, Acknowledgement (ACK) sur 1 bit. Si ACK = 1, cela indique que le champ Acknowledgement number est utilisé ;

10 – Champ numéroté 4 sur la figure, Reset (RST) sur 1 bit. Si RST = 1, cela signifie que l'émetteur demande que la connexion TCP soit redémarrée ;

11 – Champ numéroté 5 sur la figure, Push function (PSH) sur 1 bit. Si PSH = 1, cela indique que l'émetteur souhaite que les données de ce segment soient délivrées le plus tôt possible au destinataire ;

12 – Champ numéroté 6 sur la figure, Terminale (FIN) sur 1 bit. Si FIN = 1, cela signifie que l'émetteur souhaite fermer la connexion ;

13 – Window (WNDW) sur 16 bits. La valeur indiquée dans ce champ donne le nombre d'octets que le récepteur accepte de recevoir. Plus exactement, la valeur de WNDW contient le numéro du dernier octet que l'émetteur du segment peut prendre en compte. En retranchant le numéro indiqué dans Acknowledgement number, on obtient le nombre d'octets que le récepteur accepte de recevoir ;

14 – Checksum sur 16 bits. Les deux octets permettent de détecter les erreurs dans l'en-tête et le corps du segment ;

15 – Urgent Pointer (URGPTR) sur 16 bits. Ce champ spécifie le dernier octet d'un message urgent ;

16 – Options (OPT). Cette zone contient les différentes options du protocole TCP. On y trouve principalement des options de routage.

Le segment se termine par les données transportées.

Donnons quelques précisions sur les mécanismes que nous avons introduits en décrivant le format du segment.

Acquittements

Les segments étant de taille variable, les acquittements se rapportent à un numéro d'octet dans le flot de données. Chaque acquittement spécifie le numéro du prochain octet à transmettre et acquitte les précédents.

Le principe des acquittements TCP est appelé « cumulatif » car il spécifie le nombre d'octets du flot de données reçues. Cela a des avantages mais aussi des inconvénients. Des acquittements simples à générer et non ambigus forment un avantage. D'autre part, la perte d'un acquittement n'implique pas nécessairement une retransmission. En revanche, l'émetteur ne reçoit pas les acquittements de toutes les transmissions réussies, mais seulement la position dans le flot des données qui ont été reçues.

La façon de gérer les temporisateurs et les acquittements est une des idées importantes de TCP. Le protocole TCP se base sur le principe des acquittements positifs. Chaque fois qu'un segment est émis, un temporisateur est armé en attente de l'acquittement. Si le temporisateur expire avant que les données du segment n'aient été acquittées, TCP suppose que le segment a été perdu et le retransmet.

TCP ne faisant aucune hypothèse sur les réseaux traversés (temps de transit), il est impossible *a priori* de savoir quand l'acquittement va être reçu en retour. De plus, le temps de traversée d'une passerelle dépend de la charge du réseau qui varie dans le temps.

TCP utilise un algorithme adaptatif pour prendre en compte ces variations. Il enregistre l'heure à laquelle il a envoyé le segment et l'heure à laquelle il reçoit l'acquittement correspondant. Cette mesure lui permet de calculer la durée du temporisateur de retransmission.

Congestions du réseau

TCP doit également réagir aux congestions du réseau. La congestion est considérée comme un accroissement important du temps de transit, suite à une surcharge de datagrammes dans un ou plusieurs nœuds ou routeurs intermédiaires. Cette congestion peut aboutir à la suppression des datagrammes en surcharge.

Quand une congestion survient, TCP doit réagir en réduisant le débit de la connexion. Les passerelles peuvent utiliser ICMP pour prévenir les machines de la congestion, mais les protocoles de transport peuvent se rendre compte du problème en observant l'augmentation du temps de réponse. Si le protocole ne réagit pas aux congestions, le nombre de retransmissions peut continuer à augmenter, et aggraver la congestion.

Connexions duplex

Pour établir une connexion, TCP utilise un dialogue en trois étapes. Le dialogue a été prévu pour prendre en compte les demandes d'ouvertures simultanées de la part des deux machines différentes. Lorsque la connexion est établie, le transfert des données peut commencer en mode équilibré. Il n'y a pas de notion de maître ou d'esclave.

Ce mécanisme répond à deux fonctions importantes. Il garantit que les deux extrémités de la connexion sont prêtes à transférer des données (elles savent qu'elles le sont toutes les deux), et il permet aux deux parties de se mettre d'accord sur le numéro de séquence initial.

Les connexions TCP sont duplex, et sont vues comme deux voies de transmission indépendantes, chacune dans son sens. Quand un programme indique à TCP qu'il n'a plus de données à émettre, TCP fermera la connexion dans ce sens. Pour fermer sa demi-connexion, TCP transmet les données restantes, et envoie un segment avec le bit FIN positionné (en émission). En réception, TCP acquitte le segment FIN et informe le programme d'application récepteur qu'il n'y a plus de données disponibles. TCP refusera toutes données pour ce sens de transmission, mais l'envoi peut continuer pour le sens inverse. Quand les deux sens sont fermés, la connexion est supprimée.

Optimisation des performances

TCP est libre de découper le flot de données en segments sans s'occuper de la taille des données transférées par l'application. Cela permet d'optimiser les performances du protocole. Il peut garder assez d'octets dans ses mémoires pour préparer un segment raisonnablement long, qui réduit l'overhead du protocole quand la partie donnée du segment est trop courte.

Ce mécanisme améliore le débit, mais cela peut gêner certaines applications. En prenant l'exemple d'une connexion TCP utilisée pour transférer les données d'un terminal interactif vers une station éloignée, les données saisies au clavier ne seront pas disponibles immédiatement.

Pour tenir compte de ce problème, TCP offre une fonction PUSH, qui peut être utilisé par l'application pour forcer l'envoi des données présentes dans la mémoire sans attendre qu'il soit plein. Les données seront remises au programme d'application extrémité dès leur réception.

En résumé, TCP est un protocole de niveau transport en mode connexion, très complet, permettant de supporter des applications qui souhaitent une bonne qualité du transport de leur information.

10. RSVP (*Reservation Protocol*)

La communauté Internet souhaite faire évoluer le réseau vers les applications multimédias. Dans ce cas, le réseau Internet devra faire transiter des applications isochrones ou au moins des applications avec une qualité de service déterminée. La synchronisation nécessaire ou la qualité de service ne pourra être obtenue que par l'adjonction de nouveaux protocoles qui permettront la réservation de ressources.

Plusieurs solutions se sont fait jour, et deux écoles principales semblent s'affronter sur l'utilisation des nouveaux mécanismes développés par l'IETF.

La première vision, qui regroupe la communauté classique de l'Internet, reste sur la position que seul le **service « best effort »** est intéressant et que les sources et les routeurs doivent s'adapter pour essayer d'écouler un maximum de trafic. L'optique est toujours un

réseau de réseaux et il est demandé à chaque réseau d'essayer de faire au mieux. La qualité de service de bout en bout sera réalisée en optimisant les qualités de service des sous-réseaux. L'arrivée d'un protocole avec réservation permet de définir des fonctions supplémentaires ; ce protocole doit indiquer aux routeurs situés entre les sous-réseaux le type de trafic qui va passer et prie ces routeurs d'en tenir compte de la façon la plus efficace possible.

La deuxième vision est d'utiliser ce **protocole de réservation en gardant effectivement des ressources dans les routeurs** pour assurer une qualité de service de bout en bout. Cette seconde direction est celle des opérateurs de réseaux IP au sens « télécom » du terme, l'environnement Internet ayant plutôt choisi la première solution. Nous reviendrons dans la conclusion sur ces deux visions.

Le protocole qui semble le plus intéressant dans la nouvelle génération est RSVP (*ReSerVation Protocol*). Ce protocole est un protocole de signalisation qui a pour but de signaler aux nœuds intermédiaires l'arrivée de flux correspondant à des qualités de service. Par lui-même, il ne permet pas de lancer explicitement la réservation de ressources, à la demande d'une application et de les relâcher à la fin.

Cette signalisation s'effectue sur un flot (*flow*) qui est envoyé vers un ou plusieurs récepteurs. Le flot est identifié par une adresse IP ou un port de destination ou une étiquette de flot (*flow label* dans IPv6).

Dans la vision opérateur, le protocole est lié à une réservation à effectuer dans les nœuds du réseau sur une route particulière ou sur les routes déterminées par un multipoint. Les difficultés rencontrées pour mettre en œuvre ce mécanisme sont de deux ordres : comment déterminer la quantité de ressources à réserver à tout instant et comment réserver des ressources sur une route unique étant donné que le routage des paquets IP fait varier le chemin à suivre.

Le protocole RSVP effectue la réservation à partir du récepteur ou des récepteurs dans le cas d'un multipoint. Cela peut paraître surprenant à première vue mais, en fait, cette solution s'adapte bien à beaucoup de cas de figure, en particulier le multipoint. Lorsqu'un nouveau point s'ajoute au multipoint, celui-ci peut réaliser l'adjonction de réservation d'une façon plus simple que cela pourrait être fait par l'émetteur.

Le protocole RSVP est transporté dans les paquets IP. Plus précisément dans le champ de données du paquet IP. La valeur 46 dans l'en-tête IP indique une zone RSVP dans les données du paquet. La figure **8** précise la forme du champ RSVP ; dans cette figure, nous avons repris l'en-tête d'IPv6. La zone RSVP commence après les adresses de la source et du destinataire.

L'en-tête même de RSVP contient huit champs plus deux champs qui sont réservés pour des utilisations ultérieures. Ces dix champs sont les suivants :

1 – Le premier champ contient le numéro de la version actuelle de RSVP : la valeur 2 ;
2 – Les quatre bits « flags » sont réservés pour une utilisation ultérieure ;
3 – Le champ « RSVP type », caractérise le message RSVP ; actuellement, deux types sont plus spécifiquement utilisés : le message de chemin et le message de réservation mais d'autres possibilités sont également définies dans la norme. Les différentes valeurs de ce champ qui ont été retenues sont les suivantes :

1 path message,
2 reservation message,
3 error indication in response to path message,
4 error indication in response to reservation message,
5 path teardown message,
6 reservation terdown message ;

4 – Le champ cheksum permet classiquement de détecter les erreurs ;
5 – La longueur du message indique le nombre d'octets transportés dans le datagramme ;
6 – Le champ suivant est réservé pour un usage ultérieur ;

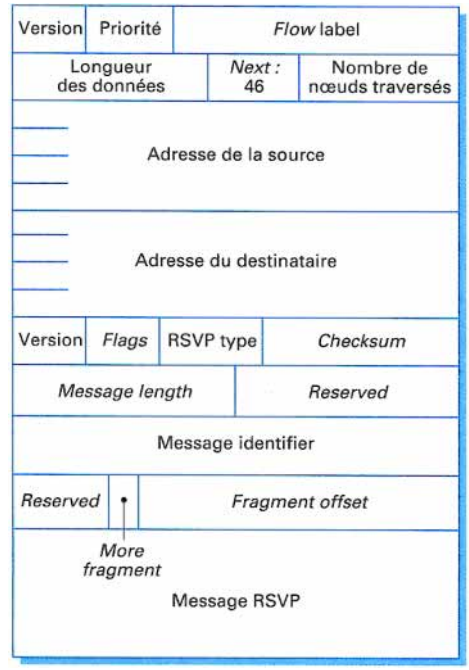


Figure 8 – Format du message RSVP

7 – Le champ « message identifier » est utilisé dans la fragmentation et il donne une valeur commune à l'ensemble des fragments d'un même message pour permettre au récepteur de pouvoir les reconnaître ;
8 – Le champ suivant est réservé pour un usage ultérieur ;
9 – Le bit « more fragment » indique que le fragment n'est pas le dernier. Un zéro sera mis dans ce champ pour le dernier fragment ;
10 – Le champ « fragment offset » indique l'emplacement du fragment dans le message.

La dernière partie du message regroupe une série de champs optionnels que l'on nomme objet. Chaque objet se présente de la même façon avec un champ longueur de l'objet sur deux octets puis le numéro de l'objet sur un octet qui détermine l'objet puis un octet pour indiquer le type d'objet.

Ces champs optionnels sont principalement dévolus au routage et à la sécurité. Les spécifications de RSVP contiennent des descriptions précises des chemins suivis par les messages en incluant les objets nécessaires et l'ordre dans lequel ces objets vont apparaître dans le message. De même, des possibilités d'identification et de sécurité sont accessibles par l'utilisation de ces champs.

11. RTP (*Real Time Protocol*)

Un des problèmes posés par Internet est l'existence d'applications temps réel comme la parole numérique ou la visioconférence. Ces applications demandent des qualités de service que les protocoles classiques d'Internet ne peuvent pas offrir. C'est la raison de la naissance de RTP. De plus, RTP a été conçu directement dans un environnement multipoint. RTP aura donc à sa charge aussi bien la gestion du temps réel que l'administration de la session multipoint.

Deux intermédiaires sont nécessaires : les translateurs (*translator*) et les mixeurs (*mixer*). Le translateur a pour fonction de translater une application codée dans un certain format en un autre format mieux adapté pour le passage par un sous-réseau. Par exemple, une application de visioconférence codée en MPEG pourrait être décodée et recodée en H.261 pour réduire la quantité d'informations transmises. Le mixeur a pour but de regrouper plusieurs applications correspondant à plusieurs flots distincts en un seul flot gardant le même format. Cette approche est particulièrement intéressante pour les flux de paroles numériques.

Pour réaliser le transport en temps réel, un second protocole RTCP (*Real Time Control Protocol*) a été ajouté à RTP. En effet, les paquets RTP ne transportent que les données des utilisateurs et non les informations de supervision. Pour cela, RTCP a été créé et permet cinq types de paquets :

200 rapport de l'émetteur ;
201 rapport du récepteur ;
202 description de la source ;
203 au revoir ;
204 application spécifique.

Ces différents paquets vont permettre de donner les instructions nécessaires dans les nœuds du réseau pour contrôler au mieux les applications temps réel.

12. Extensions diverses

L'installation et l'exploitation des logiciels TCP/IP requièrent une certaine expertise. Une première extension consiste à automatiser l'installation et la maintenance des logiciels de façon à permettre à un utilisateur de relier sa machine au réseau sans avoir à la paramétrer manuellement. De ce fait, un utilisateur peut connecter son ordinateur à l'Internet sans faire appel à un spécialiste pour installer les logiciels, mettre à jour les paramètres de configuration et de routage. En particulier, il est possible d'obtenir une configuration automatique d'un calculateur par de nouveaux protocoles permettant à une machine d'obtenir et d'enregistrer automatiquement toutes les informations sur les noms et les adresses dont elle a besoin.

Des groupes de travail examinent les améliorations qui peuvent être encore effectuées sur l'environnement Internet. Le groupe « apprentissage des routeurs » travaille sur des protocoles qui permettent à une machine de **découvrir les routeurs qu'elle peut utiliser**. Actuellement, il est nécessaire de configurer l'adresse d'un routeur par défaut. Le protocole permettra de découvrir les adresses des passerelles locales et de tester en permanence ces adresses pour savoir lesquelles peuvent être utilisées à tout instant.

Une autre extension est celle du **calcul du MTU** (taille maximale des données pouvant être contenues dans une trame physique). L'idée est d'avoir inventé une méthode pour qu'une machine puisse rechercher le plus petit MTU sur un chemin particulier vers une destination donnée. La taille optimale d'un segment TCP dépend du MTU car les datagrammes plus grands que le MTU seront fragmentés, et les datagrammes plus petits augmentent proportionnellement la partie de contrôle. Ainsi, si le MTU est connu, TCP peut optimiser le débit en construisant des segments assez larges pour tenir dans un datagramme, transporté dans une seule trame physique la plus grande possible. De la même façon, UDP peut améliorer le débit en tenant compte du MTU pour choisir la taille des datagrammes.

Les nouvelles technologies de transmission permettent d'**augmenter la bande passante** tout en diminuant la taux d'erreur. La fibre optique en est un bon exemple. Elle est de plus en plus souvent utilisée pour relier deux ordinateurs, pour câbler des réseaux locaux ou des réseaux métropolitains et pour réaliser des artères haute vitesse dans les réseaux étendus. La demande pour des connexions haut débit commence à dépasser la capacité des réseaux existants. L'objectif est d'améliorer les performances des applications actuelles, mais aussi de permettre à de nouvelles applications de voir le jour. De nombreuses propositions portent sur l'association des environnements TCP/IP et ATM pour réaliser des transports à très haut débit.

L'idée est d'utiliser des sous-réseaux ATM et de réaliser un circuit virtuel passant par ces sous-réseaux. Les paquets IP sont encapsulés dans des cellules ATM qui sont de ce fait transportés avec la qualité de service voulue jusqu'à l'autre extrémité du réseau. La façon de réaliser la route qui sera suivie par les différents paquets IP, c'est-à-dire par les cellules transportant ces paquets, n'est pas normalisée et plusieurs solutions s'affrontent. Avant de donner quelques informations sur ces méthodes, indiquons que les deux solutions préconisées jusqu'ici utilisaient dans l'émulation de réseau local LANE (Local Area Network Emulation) et l'encapsulation dénommée Classical IP over ATM.

La première solution proposée sur ATM est l'IP-switching qui consiste à tracer le chemin par le premier paquet IP émis. Celui-ci est encapsulé dans des cellules ATM pour traverser le premier sous-réseau puis récupéré en tant que paquet IP dans le premier routeur pour déterminer la suite du chemin grâce à l'adresse IP puis réencapsulé dans des cellules ATM et ainsi de suite. Tous les paquets IP suivants sont décomposés en cellules et commutés directement en ATM dans les routeurs/commutateurs IP-switching.

Le Tag-switching proposé par la société CISCO est du même ordre d'idée ; une étiquette est donnée à la route pour permettre aux paquets IP encapsulés dans de l'ATM ou du relais de trame de suivre une route déterminée.

L'IETF a également proposé sa solution dénommée NHRP (*Next Hop Resolution Protocol*) qui, comme les précédentes, consiste à choisir le nœud suivant pour réaliser la route qui sera suivie.

D'autres possibilités sont également envisageables au travers de la proposition MPOA (*Multiple Protocol Over ATM*) de l'ATM Forum qui, toujours sur un des sous-réseaux ATM, permet de transporter des paquets IP mais aussi d'autres protocoles comme IPX.

13. Sécurité

TCP/IP rend possible une interopérabilité universelle. Cependant, dans plusieurs environnements, les administrateurs ont besoin de limiter cette interopérabilité pour protéger les données privées. Ces restrictions correspondent au problème général de la sécurité, mais l'Internet est plus difficile à rendre sûr qu'un simple ordinateur car il offre des services de communication bien plus puissants. Le problème est de savoir comment un utilisateur s'appuyant sur TCP/IP peut s'assurer de la protection de ses machines et de ses données contre les accès non autorisés.

Un groupe de travail a exploré la question de sécurisation de la messagerie en expérimentant un service de messagerie privée améliorée. L'idée est de mettre en œuvre un mécanisme qui permet à l'émetteur de crypter son message et de l'envoyer sur Internet ouvert sans permettre à quelqu'un d'autre que le destinataire de le décrypter.

Des travaux sur le filtrage des paquets dans les passerelles ont produit une variété de mécanismes qui permettent aux administrateurs de fournir des listes explicites de contrôle d'accès. Une liste d'accès spécifie un ensemble de machines et de réseaux au travers desquels la passerelle peut router les datagrammes. Si l'adresse n'est pas autorisée, le datagramme est détruit et, dans la plupart des implémentations, la passerelle enregistre la tentative de violation dans un journal. Ainsi, il est possible d'utiliser des filtres d'adresses pour surveiller les communications entre les machines. Plusieurs comités techniques continuent de travailler sur les différents aspects de la sécurité dans l'Internet.