UDP/TCP

Introduction :

Les ports :

De nombreux programmes TCP/IP peuvent être exécutés simultanément sur Internet (vous pouvez par exemple ouvrir plusieurs navigateurs simultanément ou bien naviguer sur des pages HTML tout en téléchargeant un fichier par FTP). Chacun de ces programmes travaille avec un protocole, toutefois l'ordinateur doit pouvoir distinguer les différentes sources de données.

Ainsi, pour faciliter ce processus, **chacune de ces applications se voit attribuer une adresse unique sur la machine, codée sur 16 bits: un port** (la combinaison adresse IP + port est alors une adresse unique au monde, elle est appelée **socket**).

L'adresse IP sert donc à identifier de façon unique un ordinateur sur le réseau tandis que le numéro de port indique l'application à laquelle les données sont destinées. De cette manière, lorsque l'ordinateur reçoit des informations destinées à un port, les données sont envoyées vers l'application correspondante. S'il s'agit d'une requête à destination de l'application, l'application est appelée application serveur. S'il s'agit d'une réponse, on parle alors d'application cliente.

La fonction de multiplexage

Le processus qui consiste à pouvoir faire transiter sur une connexion des informations provenant de diverses applications s'appelle le **multiplexage**. De la même façon le fait de répartir le flux de données sur les diverses applications s'appelle le **démultiplexage**.

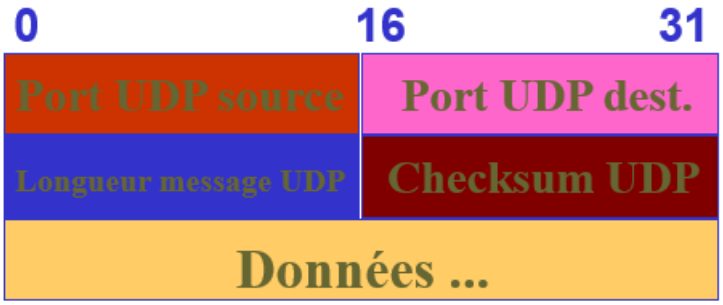
User Datagram Protocol :

Caractèristiques :

- Pas d'établissement de connexion
- Pas fiable :
  - la réception du message par le récepteur n'est pas garantie. (ne fournit pas de contrôle d'erreurs)
  - L'ordonnancement n'est pas garanti.

Remarque : La fonction principal qu'ajoute l'UDP au protocole IP est la notion de port, qui permet d'adresser un processus spécifique dans une machine.

Format du **datagramme UDP** :



Port Source: il s'agit du numéro de port correspondant à l'application émettrice du segment UDP. Ce champ représente une adresse de réponse pour le destinataire. Ainsi, **ce champ est optionnel**, cela signifie que si l'on ne précise pas le port source, les 16 bits de ce champ seront mis à zéro, auquel cas

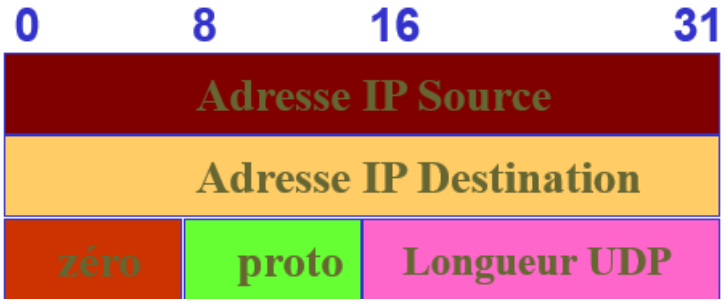
le destinataire ne pourra pas répondre (cela n'est pas forcément nécessaire, notamment pour des messages unidirectionnels).

Port Destination: Ce champ contient le port correspondant à l'application de la machine destinataire à laquelle on s'adresse.

Longueur: Ce champ précise la longueur totale du segment en octets, en-tête comprise, or l'en-tête a une longueur de 4 x 16 bits donc le champ longueur est nécessairement supérieur ou égal à 8 octets (2 mots de 4 octets minimum).

Somme de contrôle: Il s'agit d'une somme de contrôle réalisée de telle façon à pouvoir contrôler l'intégrité du segment. Celle-ci est optionnelle aussi, quand elle utilisée, voilà sa forme :

La somme de contrôle (pour ipv4) est caclulé sur le **pseudo en-tête** suivant :



Pour le troisième mot, Le premier octet est toujours à 0, le deuxième représente le protcole utilisé (17 : UDP) et le dernier contient la longueur, le champ « longueur UDP » contient la longueur du datagramme original (en-tête + données).

Comment ça marche :

UDP fait le multiplexage et démultiplexage les datagrammes en sélectionnant les numéros de ports :

- une application obtient un numéro de port de la machine locale; dès lors que l'application émet un message via ce port, le champ PORT SOURCE du datagramme UDP contient ce numéro de port,
- une application connaît (ou obtient) un numéro de port distant afin de communiquer avec le service désiré.
- Lorsque UDP reçoit un datagramme, il vérifie que celui-ci est un des ports actuellement actifs (associé à une application) et le délivre à l'application responsable (mise en queue)
- si ce n'est pas le cas, il émet un message ICMP *port unreachable*, et détruit le datagramme.

Il y a des numéros de port réservés à des services (well-known port assignements), les ports non réservés sont assignés dynamiquement aux processus quand ils l'ont besoin.

Transmission control procotol :

Caractèristiques :

- Le message TCP est appelé **ségment**.
- fiabilité assuré par le service : service en mode connecté avec la garantie de transfert de données sans perte et avec ordonnancement.
- Transport tamponnés : découpage en ségments.
- Connexion bidirectionnelles et simultanées

Puisqu'on parle d'un service en mode connecté, on parle de :

l'établissement de la connexion : **appel, négociation et puis commence le transfert.**

les transferts de données ;

la fin de la connexion.

Une connexion est composé d'**une paire d'extrémités de connexion**, c'est-à-dire, une paire des adresses de cette forme (IP,PORT), une connexion peut donc être représenté comme suit :

((IP1,PORT1) , (IP2,PORT2))

Une extrémité de connexion peut établir une connexion avec plusieurs extrémités de connexion simultanément (**multi-instanciation**).

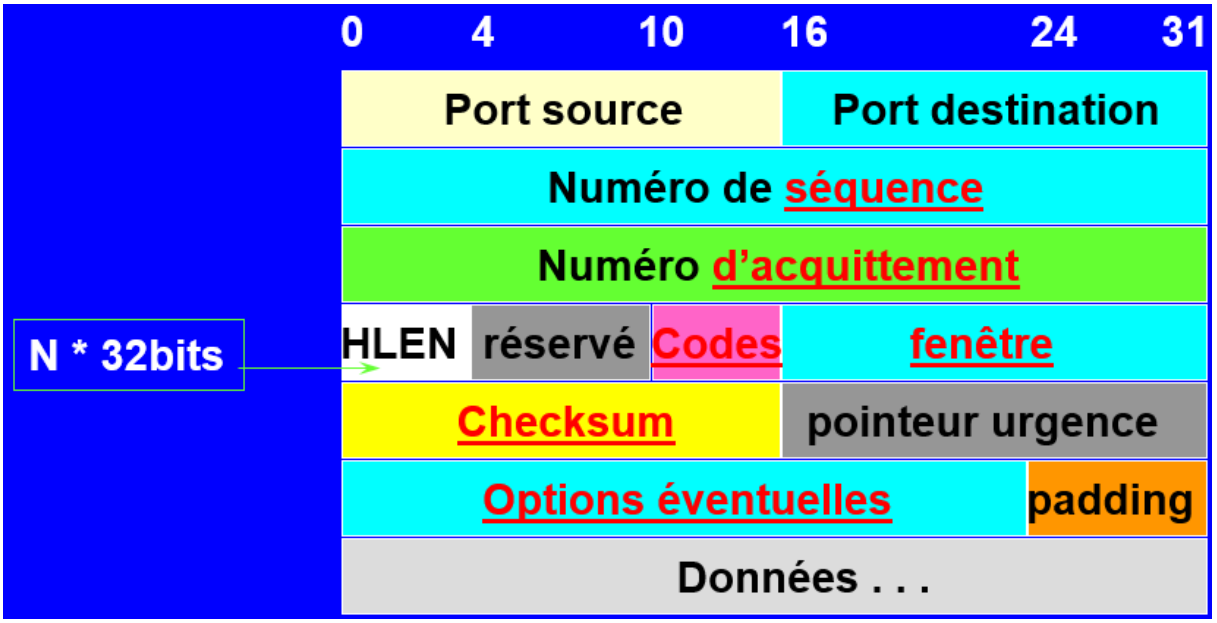
Comment la connexion est elle établie ?

-Une application (ou processus, bref une extrémité) effectue **une ouverture passive** en indiquant qu'elle accepte une connexion entrante.

-une autre application (extrémité) effectue **une ouverture active** pour demander l'établissement de la connexion avec cette première.

La ségmentation :

Ségment TCP :



Un ségment TCP permet de :

- établir les connexions,
- transférer les données,
- émettre des acquittements,
- fermer les connexions;