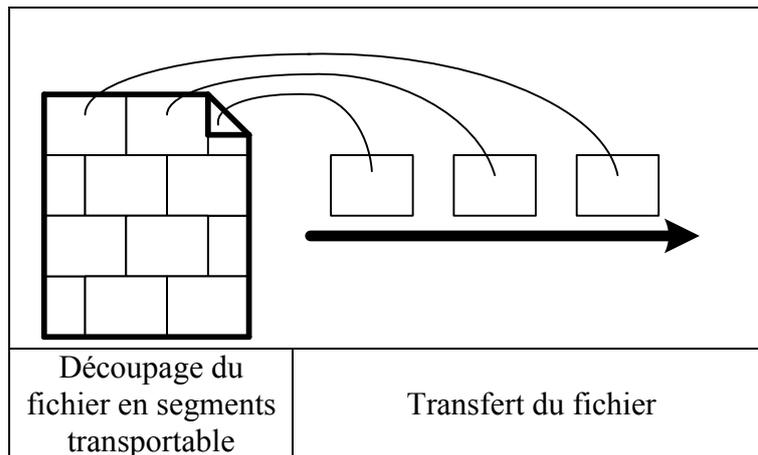


# TCP/UDP

## 1) Rôles communs

### a) Segmentation

Dans les communications réseaux, la taille des données peut aller jusqu'à plusieurs Giga Octet, il est impossible d'envoyer des paquets IP de cette taille. Il est obligatoire de découper les données en segment transportable sur le réseau. La taille maximale de chacun de ces segments est de 1,5 Ko.



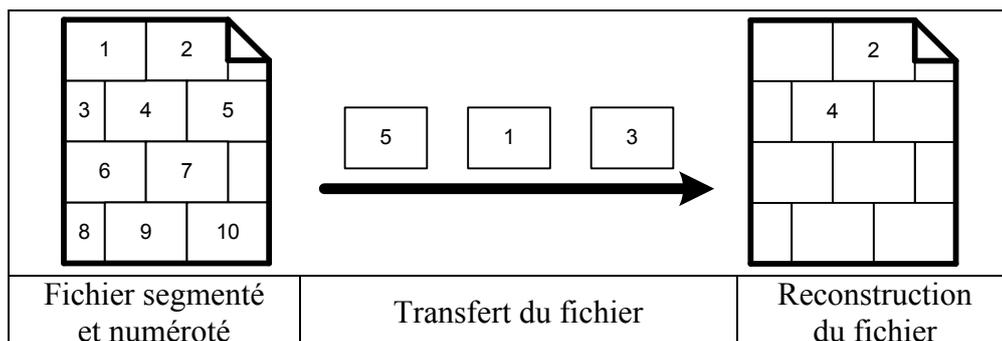
- Exercice : calculez du nombre de segment (1,5Ko) pour transférer un fichier de 700Mo.
- Rappel 1 Mo = 1024 Ko

### b) Numérotation

Problème ! Le protocole IP n'est pas fiable, tous les paquets ne mettrons pas le même temps pour arriver à destination. Dans l'exemple suivant, on voit bien que les paquets (réponse à un ping) ne mettent pas le même temps pour arriver à destination (entre 32 et 37 ms).

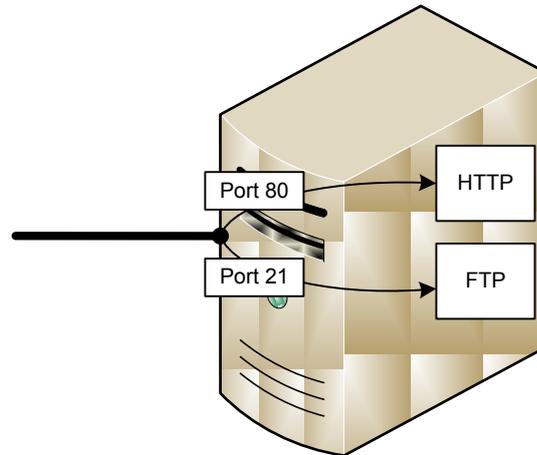
```
Réponse de 66.249.92.104 : octets=32 temps=32 ms
Réponse de 66.249.92.104 : octets=32 temps=34 ms
Réponse de 66.249.92.104 : octets=32 temps=33 ms
Réponse de 66.249.92.104 : octets=32 temps=37 ms
Réponse de 66.249.92.104 : octets=32 temps=33 ms
```

On ne peut donc pas reconstituer les données en fonction de l'ordre d'arrivée. Il y a obligation de numéroter chaque segment pour pouvoir reconstituer les données à l'arrivée.



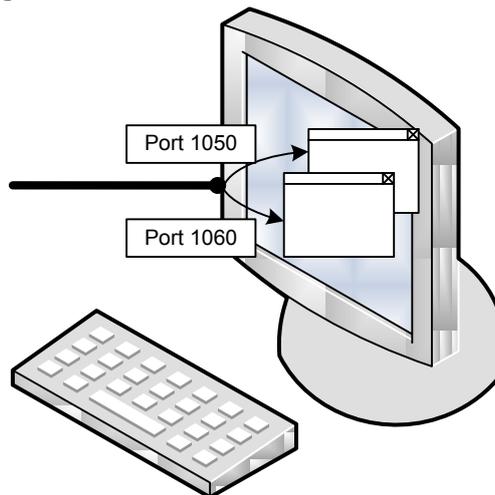
### c) Diriger les paquets vers les applications

Les ports serveurs sont fixes et connus, ils sont référencés et listés [ici](#). Dans l'exemple suivant, le trafic à destination du port 80 est redirigé vers l'application serveur HTTP et le trafic à destination du port 21 est redirigé vers l'application serveur FTP.



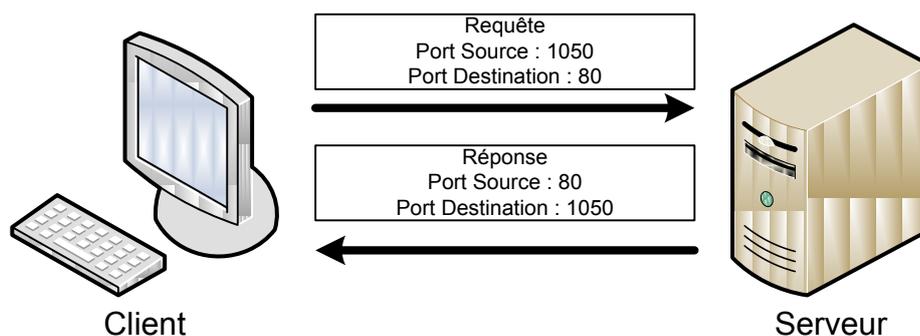
Serveur HTTP et FTP

Les ports client sont définis dynamiquement par le système d'exploitation. Le choix des ports est fait en fonction des ports libres. A l'ouverture d'une application, le système d'exploitation choisit un port parmi les ports libres. Le principe est le même que pour les applications serveur à part que les ports sont définis dynamiquement.



Client

Il y a deux types d'applications, les applications clientes et serveurs. C'est toujours le client qui demande un service au serveur. Pour rediriger les segments vers la bonne application les protocoles TCP et UDP utilisent des numéros de ports source et destination.



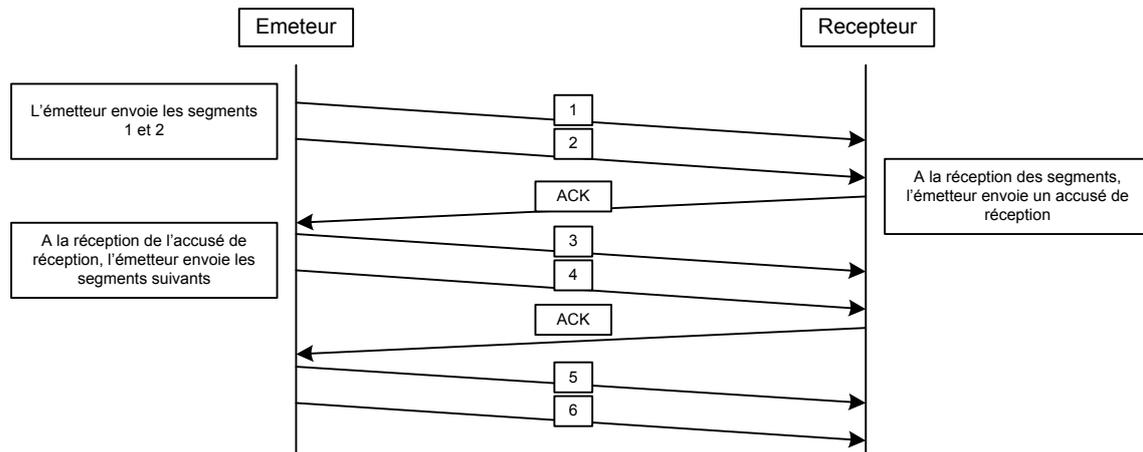
## II) TCP

### a) La fiabilité de TCP

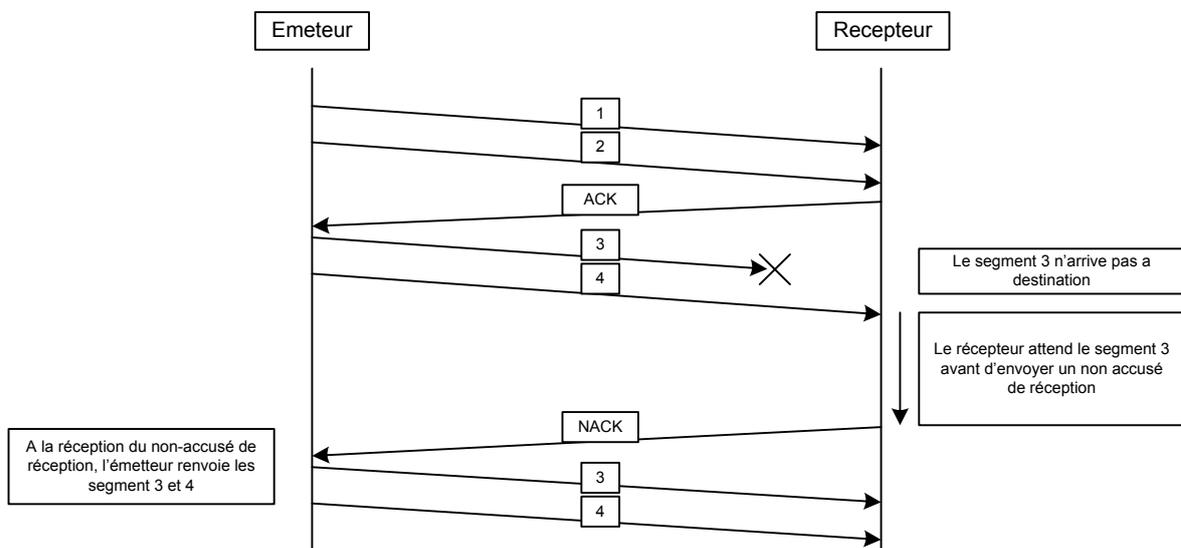
IP ne permet pas garantir la bonne réception des paquets, c'est pourquoi on l'associe à TCP. On parle du couple TCP/IP, ce couple permet de transporter des données d'un point à l'autre de la planète de manière fiable.

Pour assuré le transport fiable des données TCP utilise un système d'accusé de réception comme à la poste. Un accusé de réception se nomme acknowledgement (ACK) en anglais

### b) TCP en exemples



Dans cet exemple, l'émetteur envoie deux segments, attend que le récepteur envoie un accusé de réception pour en envoyer deux autres.



Dans cet exemple, un des segments n'arrive pas à destination. Après un instant, le récepteur envoie un non accusé de réception pour signaler à l'émetteur que les données ne sont pas arrivées normalement. A la réception du non accusé de réception l'émetteur renvoie les segments manquants.

### c) Utilisations de TCP

TCP est utilisé quand on veut envoyer des données sans perdre des segments. Ces données sont de type page web (HTTP), fichier (FTP), mail (POP, SMTP, IMAP) et bien d'autres protocoles.

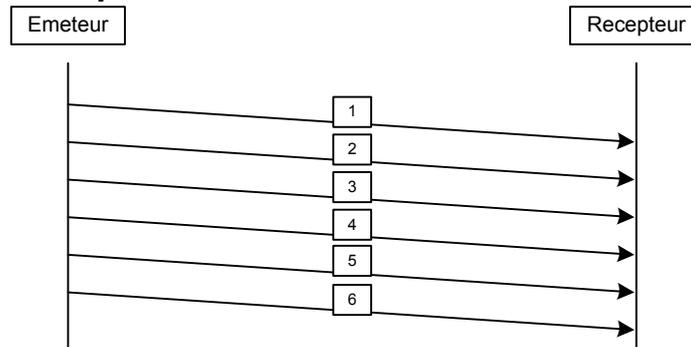
### III) UDP

#### a) La simplicité de UDP

Les accusés de réceptions de TCP nécessitent une gestion complexe et ne peut pas être utilisé dans le cas d'équipements réseaux simple. De plus les accusés de réceptions provoquent des lenteurs, l'émetteur doit attendre une réponse du récepteur avant d'envoyer des segments.

UDP n'utilise pas d'accusé de réception, les segments sont envoyés les uns à la suite des autres sans que l'émetteur sache si les segments sont bien arrivés ou pas.

#### b) UDP en exemple



Dans cet exemple, les segments sont envoyés en continu.

#### c) Utilisations de UDP

UDP permet une gestion simple des communications, c'est pourquoi il est utilisé pour les transferts de fichier avec TFTP. Le protocole TFTP est utilisé pour mettre à jour le firmware des équipements défectueux. Le protocole TFTP est aussi utilisé pour le transfert de fichier dans le cas de boot PXE.

TCP fonctionne uniquement en point à point, c'est pourquoi UDP est utilisé dans le cas de communications multicast. Le multicast permet l'envoi de paquet d'une source vers plusieurs destinataire.

UDP n'utilise pas d'accusé de réception, donc le transfert de données est rapide. C'est pourquoi UDP est utilisé quand on a des flux (streaming) audio ou vidéo à transporter. Les domaines d'applications sont la vidéo surveillance et la VOIP.

Pourquoi segmenter les données avant de les envoyer ?

---

Pourquoi numéroter les segments avant de les envoyer ?

---

Quel est le rôle des numéros de ports ?

---

Pourquoi les numéros de ports serveur sont ils fixe ?

---

Pourquoi les numéros de ports clients sont ils dynamique ?

---

Pourquoi a ont besoin d'un protocole fiable comme TCP ?

---

Quel mécanisme permet à TCP d'être fiable ?

---

Dans quels cas utiliser TCP ?

---

Quel sont les avantages d'utiliser UDP par rapport à TCP.

---

Dans quels cas utiliser UDP ?

---