Couche Transport

Laurent Signac – CC-BY-SA – 30-12-20 0908 ad05cfc112fc6da93301

Même si le terme le plus employé, lorsqu'on parle d'internet est TCP/IP, *deux protocoles* sont majoritairement employés dans la couche transport :

- TCP pour *Transmission Control Protocol*, qui concerne l'émission de données avec contrôle de la bonne réception, gestion du flux etc.
- UDP pour *User Datagram Protocol*, qui concerne l'émission de données sans contrôle de la bonne réception.

En ce qui concerne le vocabulaire, on parle d'un **flux** TCP issu de la couche application, découpé en **segments** TCP, et de **datagrammes**, ou **paquets** UDP.

Le protocole TCP est orienté connexion, il s'assure de la bonne réception des segments, et gère le flux. Au contraire, UDP fournit un service sans connexion. Les datagrammes UDP peuvent être perdus sans que l'émetteur en soit informé. Ils sont plutôt réservés aux messages courts.

Avec TCP et UDP, la notion de *port* permet de multiplexer les communications : le numéro de port permet d'adresser une application particulière sur la machine cible.

1 User Datagram Protocol (RFC 768[1])

L'en-tête UDP (User Datagram Protocol) est formé de deux mots de 32 bits (figure 1). Les paquets UDP ont une taille limite de 512 octets.

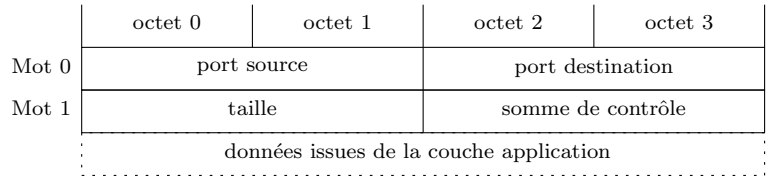

Figure 1: En-tête UDP

Port source : Permet de repérer l'application émettrice.

Port destination : Permet de repérer l'application réceptrice.

Taille : Longueur totale du paquet, en-tête compris.

Somme de contrôle : Complément à 1 sur 16 bits de la somme des compléments à 1 sur 16 bits des mots de 16 bits de l'en-tête associé à d'autres informations (adresses IP notamment).

En raison de son caractère non-sûr, UDP est principalement utilisé pour la résolution de noms (DNS) ou dans le cas de petits messages.

Exemple d'un entête Udp

```
0x0020:  .... 8451 0035 002e eb9a .... .... ....
```

2 Transmission Control Protocol (RFC 793[2])

L'entête TCP est formé d'au moins cinq mots de 32 bits, comme indiqué figure 2.

Port source : Permet de repérer l'application émettrice.

Port destination : Permet de repérer l'application réceptrice.

Séquence : Utilisé pour la synchronisation (voir plus loin).

[1] https://www.rfc-editor.org/info/rfc768
[2] https://www.rfc-editor.org/info/rfc793

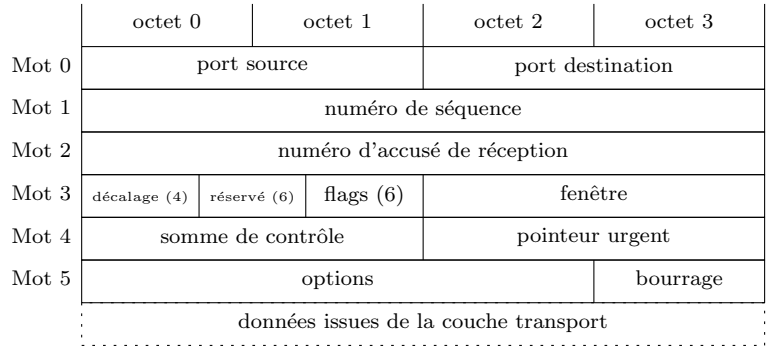

Figure 2: En-tête TCP

Numéro d'accusé de réception : Utilisé pour la synchronisation (voir plus loin).

Décalage : Début des données, par rapport au début du paquet, exprimé en nombre de mots de 32 bits (sur 4 bits).

Réservé : Six bits non utilisés actuellement.

Flags : Six drapeaux de 1 bit (poids fort vers poids faible):

Urg : Pointeur de données urgentes significatif.

Ack : Accusé de réception significatif.

Psh : Fonction Push.

Rst : Réinitialisation de la connexion.

Syn : Synchronisation des numéros de séquence.

Fin : Fin de transmission.

Fenêtre : Nombre d'octets que le récepteur souhaite recevoir sans renvoyer d'accusé de réception (voir plus loin).

Somme de contrôle : Complément à 1 sur 16 bits de la somme des compléments à 1 sur 16 bits des mots de 16 bits de l'en-tête associé à d'autres informations (adresses IP notamment).

Pointeur urgent : Position d'une donnée urgente, exprimée comme le décalage par rapport au numéro de séquence.

Options : Un en-tête TCP peut contenir ou non des options. Nous ne les détaillerons pas ici (voir Rfc).

Bourrage : Suite de 0 pour que la longueur de l'en-tête soit multiple de 32 bits.

Exemple de segment tcp

```
0x0020:  .... 0015 835d 7639 7848 fce5 80a9 8018  .....]v9xH......
0x0030:  2338 e818 0000 0101 080a 31cb b8ed 01f7  #8........1.....
0x0040:  7b19 .... .... .... .... .... .... ....  {...............
```

Du fait que TCP est un protocole *orienté connexion*, lors de la transmission d'un flux TCP par la couche application, on distingue le début de la connexion, le transfert des données, puis la fin de la connexion. De plus, l'ordre des segments transmis a son importance. En raison du fonctionnement d'IP, il se peut que les segments n'arrivent pas à destination dans le même ordre que lorsqu'ils ont été émis. L'ordre des segments doit pouvoir être retrouvé. Ce réarrangement est possible grâce au *numéro de séquence*, qui permet de repérer l'ordre des segments. La phase de connexion a entre autres pour rôle de permettre aux machines communicantes de s'échanger leurs numéros de séquence initiaux.

L'établissement d'une connexion se fait en trois phases. Pour se connecter à la machine B, la machine A envoie un premier segment à B, avec le drapeau SYN positionné. Ceci permet à B de savoir que A souhaite se connecter et de récupérer son numéro de séquence. B répond à A avec les drapeaux SYN et ACK positionnés. Ceci permet à A de savoir que B a bien reçu son premier segment (drapeau ACK) et de connaître le numéro

de séquence de B. Puis, A peut transmettre les premières données à B, en mettant le drapeau ACK dans le premier segment, ce qui permet à B de savoir que A a bien reçu son numéro de séquence initial (figure 3).

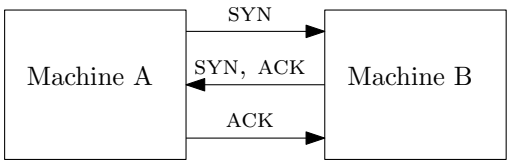

Figure 3: Établissement d'une connexion TCP

Les deux champs *numéro de séquence* et *numéro d'accusé de réception* sont utilisés pour numéroter les octets transmis dans le flux TCP. Supposons que A souhaite transmettre 2000 octets à B. Chaque segment envoyé contiendra comme numéro de séquence le numéro d'ordre du premier octet de données du segment dans le flux. Le numéro de séquence initial est généralement choisi aléatoirement. Dans la suite, nous noterons ce numéro ISN. Précisons que dans la numérotation des octets transmis, seuls les données et les deux drapeaux SYN et FIN comptent (les deux drapeaux comptent pour un octet chacun).

Le champs *numéro d'accusé de réception* permet comme son nom l'indique d'accuser réception d'une partie du flux. Il contient le numéro de séquence que le récepteur s'attend à présent à recevoir. Si le récepteur a par exemple reçu 3 segments contenant chacun 110 octets, alors il s'attend à recevoir dans le prochain segment le numéro de séquence ISN+331. Il accusera donc réception du troisième segment en indiquant dans le champs *numéro d'accusé de réception* la valeur ISN+331 et en positionnant le drapeau ACK. Lorsqu'une machine accuse réception, le champs *Fenêtre* est aussi renseigné. Il indique combien d'octets le récepteur est encore disposé à recevoir. Ceci permet au récepteur de contrôler le débit du flux qui lui est envoyé. La machine réceptrice n'a pas obligation d'accuser réception de chaque segment (cela ralentirait beaucoup les connexions). L'émetteur va donc émettre *même s'il n'a pas reçu tous les accusés de réception.* La quantité d'information qu'il pourra ainsi émettre sans attendre d'accusé de réception ne devra pas dépasser la taille de la fenêtre. Un exemple est donné figure 4.

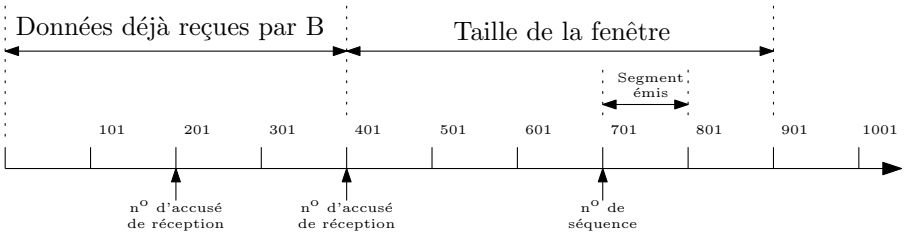

Figure 4: Utilisation du numéro d'accusé de réception pour réguler le flux

Le dernier accusé de réception reçu par A comportait comme numéro d'accusé de réception ISN+401. La machine A sait donc que B a reçu les 400 premiers octets. Dans le segment accusant réception, B a indiqué pour taille de fenêtre la valeur 500. A sait, par conséquent, qu'il pourra transmettre les données jusqu'à l'octet 900 inclus sans attendre de nouvel accusé de réception. Si une fois l'octet 900 transmis, A n'a toujours pas de nouvel accusé de réception, A attend d'en recevoir un sans plus rien transmettre. Si entre temps (avant la transmission de l'octet 900) A a reçu un accusé de réception, il recalcule le numéro d'ordre du dernier octet qu'il pourra transmettre sans attendre de nouvel accusé de réception.

Exemple de connexion tcp

```
A>B
0x0020:  .... 835d 0015 fce5 80a8 0000 0000 a002  ...]............
0x0030:  16d0 1ec4 0000 0204 05b4 0402 080a 01f7  ................
0x0040:  7b07 0000 0000 0103 0300                 {.........
B>A
0x0020:  .... 0015 835d 7639 7847 fce5 80a9 a012  .....]v9xG......
0x0030:  2338 3cc4 0000 0101 080a 31cb b8db 01f7  #8<.......1.....
0x0040:  7b07 0103 0300 0204 0514                 {.........
A>B
0x0020:  .... 835d 0015 fce5 80a9 7639 7848 8010  ...]......v9xH..
```

```
0x0030:  16d0 743e 0000 0101 080a 01f7 7b19 31cb  ..t>........{.1.
0x0040:  b8db                                     ..
B>A
0x0020:  .... 0015 835d 7639 7848 fce5 80a9 8018  .....]v9xH......
0x0030:  2338 e818 0000 0101 080a 31cb b8ed 01f7  #8........1.....
0x0040:  7b19 .... .... .... .... .... .... ....  {...............
```

Parmi les autres drapeaux, RST permet de couper brutalement une connexion.

La fermeture d'une connexion s'effectue à la manière de son ouverture : La machine A qui désire fermer la connexion envoie les drapeaux FIN et ACK. La machine B répondra avec les drapeaux FIN et ACK (ce dernier pour acquitter le drapeau FIN de A). Enfin la machine A confirmera la réception en envoyant un drapeau ACK.

Une connexion TCP peut se trouver dans plusieurs états, dont les noms sont normalisés. Nous listons ici les principaux :

Listen : Attente d'une requête de connexion externe.

Established : La connexion s'est effectuée.

Closed : Pas de connexion.

Les autres états (nombreux) décrivent les différentes étapes de l'établissement ou de la fermeture d'une connexion.