Le protocole TCP

1) Présentation de TCP

TCP (**T**ransmission **C**ontrol **P**rotocol) est le protocole le plus utilisé de la **couche transport** d'internet.

TCP gère les accusés de réception, la détection des pertes de données et la gestion de l'ordre de réception. Ces fonctionnalités sont indispensables pour de nombreuses applications : navigateurs web, messageries, transfert de fichiers, ….

Une session TCP fonctionne en trois phases :

- établissement de la connexion ;
- transferts de données ;
- fin de la connexion.

Chaque paquet reçu est contrôlé et un accusé de réception est envoyé à l'expéditeur. Si le paquet est altéré ou non reçu au bout d'un certain délai, alors l'expéditeur le renvoie.

Grâce **aux numéros de séquence** et **numéros d'acquittement**, les systèmes terminaux peuvent remettre les données reçues dans le bon ordre.

Les segments sont encapsulés dans un paquet IP.

2) Établissement d'une connexion TCP

Selon le protocole de communication TCP, une connexion entre deux hôtes s'établit en trois étapes : c'est le **three-way handshake**.

Etape 1 : Le client qui désire établir une connexion avec un serveur va envoyer un premier paquet SYN (synchronized) au serveur. Le numéro de séquence de ce paquet est un nombre aléatoire x.

Etape 2 : Le serveur va répondre au client à l'aide d'un paquet SYN-ACK (*synchronize- acknowledge*). Le numéro du ACK est égal au numéro de séquence du paquet précédent (SYN) plus un ($x + 1$) tandis que le numéro de séquence du paquet SYN-ACK est un nombre aléatoire y.

Etape 3 : Pour terminer, le client va envoyer un paquet ACK au serveur qui va servir d'accusé de réception. Le numéro de séquence de ce paquet est défini selon la valeur de l'acquittement reçu précédemment ($x + 1$) et le numéro du ACK est égal au numéro de séquence du paquet précédent (SYN-ACK) plus un ($y + 1$).

3) Transfert de données

Des numéros de séquence et le numéro d'acquittement sont présents dans chaque segment envoyé, ce qui permet de classer chronologiquement les segments.

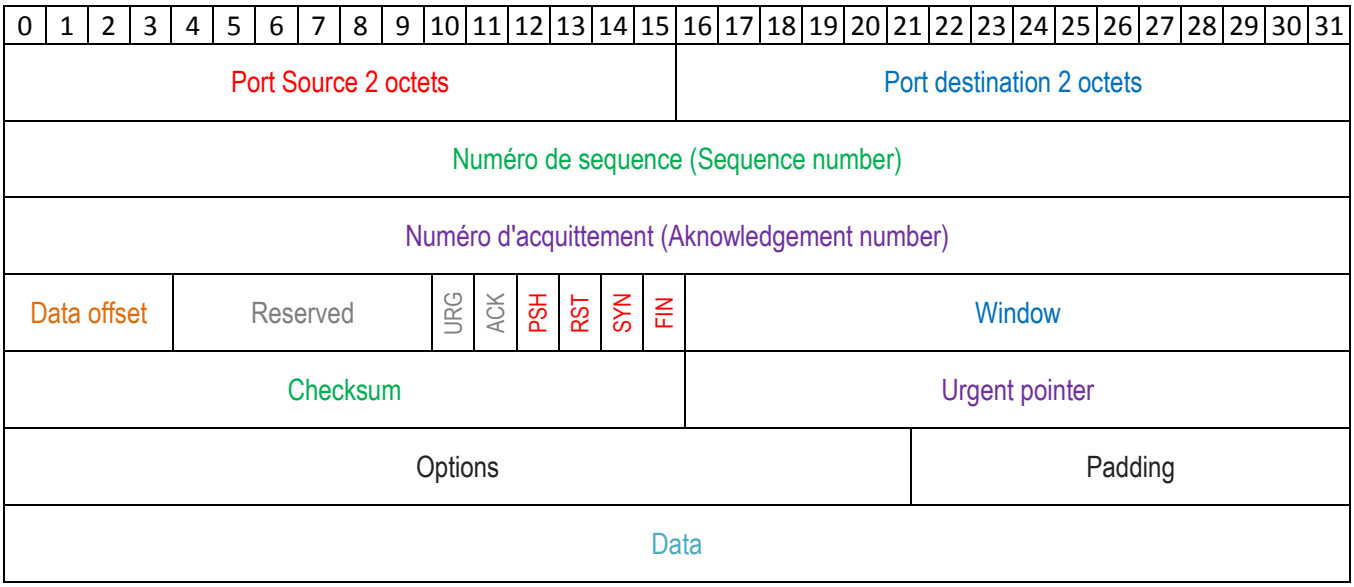
Un checksum sur 16 bits, est calculé par l'émetteur, et inclus dans chaque segment. Le destinataire contrôle le checksum, et s'il correspond, on considère que le segment a été reçu intact.

Si le segment n'a pas été reçu ou s'il est altéré, alors il est renvoyé.

4) Fermeture d'une connexion

La phase de terminaison d'une connexion utilise un handshaking en quatre temps.

5) Structure d'un segment TCP



Analyse d'un segment TCP :

EE 4F 00 50 43 21 00 00 AB CD AB CD 50 38 11 A0 2F 66 1A 1B 54 4F 54 4F

Port source : numéro du port source →EE 4F

Port destination : numéro du port destination → 00 50

Numéro de séquence : numéro de séquence du premier octet de ce segment → 43 21 00 00

Numéro d'acquittement : numéro de séquence du prochain octet attendu → AB CD AB CD

Data offset : nombre de mots de 32 bits de l'entête →5*

Reserved : →000000

URG : Signale la présence de données urgentes→1

ACK : signale que le paquet est un accusé de réception (acknowledgement)→1

PSH : données à envoyer tout de suite (push)→1

RST : rupture anormale de la connexion (reset)→0

SYN : demande de synchronisation ou établissement de connexion→0

FIN : demande la fin de la connexion→0

Window : indique la quantité de données acceptée sans exiger d'accusé de réception.→ 11 A0

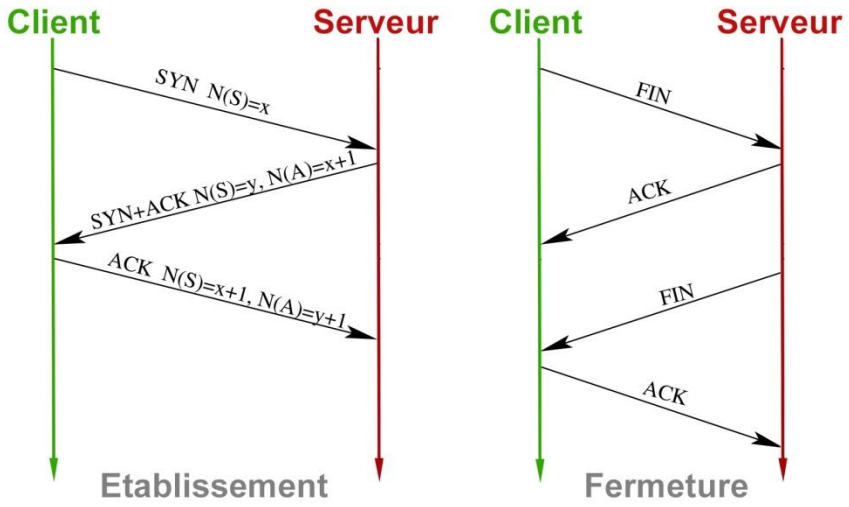
Checksum : →2F 66

Urgent pointer : quasiment jamais utilisé →1A 1B

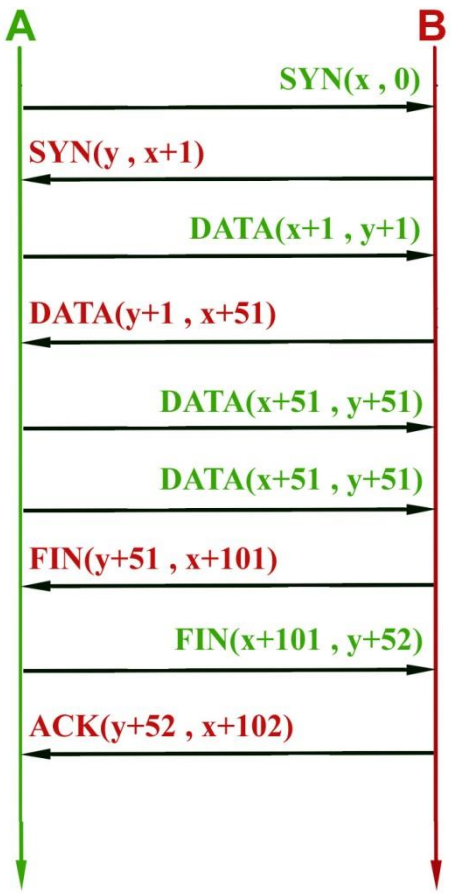
Option et padding : rien (*5 mots de 32 bits en en-tête)

Data : →54 4F 54 4F

6) Gestion de la connexion TCP (handshaking)



7) Echange TCP entre deux stations A et B



La figure ci-contre illustre une communication TCP complète entre deux stations A et B. Seuls les segments reçus et émis par la station A sont représentés. Les segments de données sont toujours de 50 octets.

Chaque segment est représenté par sa direction, DATA pour les segments de données ou l'un des flags SYN, ACK ou FIN.

Entre parenthèse, on indique le numéro de séquence et l'accusé de réception.

Les trois premiers messages établissent la connexion.

Le segment de de données dont le numéro de séquence est x+51 étant émis deux fois, on peut déduire que le premier envoi (5^{e} ligne) a été perdu.

Les trois derniers segments ferment la connexion.