Le protocole TCP

TCP (qui signifie Transmission ControlProtocol, soit en français: ProtocoledeContrôledeTransmission) est un des principaux protocoles de la couche transport du modèle TCP/IP. Il permet, au niveau des applications, de gérer les données en provenance (ou à destination) de la couche inférieure du modèle (c'estàdire le protocole IP). Lorsque les données sont fournies au protocole IP, celuici les encapsule dans des datagrammes IP, en fixant le champ protocole à 6 (Pour savoir que le protocole en amont est TCP...). TCP est un protocole orienté connexion, c'estàdire qu'il permet à deux machines qui communiquent de contrôler l'état de la transmission. Les caractéristiques principales du protocole TCP sont les suivantes :

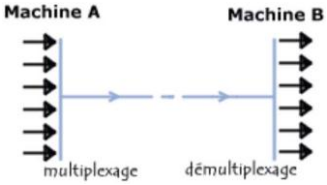
- TCP permet de remettre en ordre les datagrammes en provenance du protocole IP
- TCP permet de vérifier le flot de données afin d'éviter une saturation du réseau
- TCP permet de formater les données en segments de longueur variable afin de les "remettre" au protocole IP
- TCP permet de multiplexer les données, c'estàdire de faire circuler simultanément des informations provenant de sources (applications par exemple) distinctes sur une même ligne
- TCP permet enfin l'initialisation et la fin d'une communication de manière courtoise

Le but de TCP

Grâce au protocole TCP, les applications peuvent communiquer de façon sûre (grâce au système d'accusés de réception du protocole TCP), indépendamment des couches inférieures. Cela signifie que les routeurs (qui travaillent dans la couche Internet) ont pour seul rôle l'acheminement des données sous forme de datagrammes, sans se préoccuper du contrôle des données, car celuici est réalisé par la couche transport (plus particulièrement par le protocole TCP).
Lors d'une communication à travers le protocole TCP, les deux machines doivent établir une connexion. La machine émettrice (celle qui demande la connexion) est appelée client, tandis que la machine réceptrice est appelée serveur. On dit qu'on est alors dans un environnement ClientServeur. Les machines dans un tel environnement communiquent en mode connecté, c'est à dire que la communication se fait dans les deux sens.
Pour permettre le bon déroulement de la communication et de tous les contrôles qui l'accompagnent, les données sont encapsulées, c'est à dire qu'on ajoute aux paquets de données un entête qui va permettre de synchroniser les transmissions et d'assurer leur réception.
Une autre particularité de TCP est de pouvoir réguler le débit des données grâce à sa capacité à émettre des messages de taille variable, ces messages sont appelés segments.

La fonction de multiplexage

TCP permet d'effectuer une tâche importante: le multiplexage/démultiplexage, c'est à dire faire transiter sur une même ligne des données provenant d'applications diverses.



Ces opérations sont réalisées grâce au concept de **ports** (ou sockets), c'est à dire un numéro associé à un type d'application, qui, combiné à une **adresse IP**, permet de déterminer de façon unique une application qui tourne sur une machine donnée.

Le format des données sous TCP

Un segment TCP est constitué comme suit :

<table>
<tr><td colspan="1">0</td><td colspan="1">1</td><td colspan="1">2</td><td colspan="1">3</td><td colspan="1">4</td><td colspan="1">5</td><td colspan="1">6</td><td colspan="1">7</td><td colspan="1">8</td><td colspan="1">9</td><td colspan="1">10</td><td colspan="1">11</td><td colspan="1">12</td><td colspan="1">13</td><td colspan="1">14</td><td colspan="1">15</td><td colspan="1">16</td><td colspan="1">17</td><td colspan="1">18</td><td colspan="1">19</td><td colspan="1">20</td><td colspan="1">21</td><td colspan="1">22</td><td colspan="1">23</td><td colspan="1">24</td><td colspan="1">25</td><td colspan="1">26</td><td colspan="1">27</td><td colspan="1">28</td><td colspan="1">29</td><td colspan="1">30</td><td colspan="1">31</td></tr>
<tr><td colspan="16">Port Source</td><td colspan="16">Port destination</td></tr>
<tr><td colspan="32">Numéro d'ordre</td></tr>
<tr><td colspan="32">Numéro d'accusé de réception</td></tr>
<tr><td colspan="4">Décalagedonnées</td><td colspan="6">réservée</td><td colspan="1">URG</td><td colspan="1">ACK</td><td colspan="1">PSH</td><td colspan="1">RST</td><td colspan="1">SYN</td><td colspan="1">FIN</td><td colspan="16">Fenêtre</td></tr>
<tr><td colspan="16">Somme de contrôle</td><td colspan="16">Pointeur d'urgence</td></tr>
<tr><td colspan="22">Options</td><td colspan="10">Remplissage</td></tr>
<tr><td colspan="32">Données</td></tr>
</table>

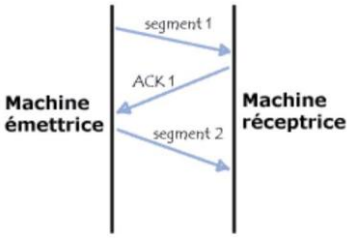
Signification des différents champs :
Port Source (16 bits): Port relatif à l'application en cours sur la machine source
Port Destination (16 bits):
Port relatif à l'application en cours sur la machine de destination

- Numéro d'ordre (32 bits): Lorsque le drapeau SYN est à 0, le numéro d'ordre est celui du premier mot du segment en cours. Lorsque SYN est à 1, le numéro d'ordre est égal au numéro d'ordre initial utilisé pour synchroniser les numéros de séquence (ISN)
- Numéro d'accusé de réception (32 bits): Le numéro d'accusé de réception également appelé numéro d'acquittement correspond au numéro (d'ordre) du prochain segment attendu, et non le numéro du dernier segment reçu.
- Décalage des données (4 bits): il permet de repérer le début des données dans le paquet. Le décalage est ici essentiel car le champ d'options est de taille variable
- Réservé (6 bits): Champ inutilisé actuellement mais prévu pour l'avenir
- Drapeaux (flags) (6x1 bit): Les drapeaux représentent des informations supplémentaires :
  - URG: si ce drapeau est à 1 le paquet doit être traité de façon urgente.
  - ACK: si ce drapeau est à 1 le paquet est un accusé de réception.
  - PSH (PUSH): si ce drapeau est à 1, le paquet fonctionne suivant la méthode PUSH.
  - RST: si ce drapeau est à 1, la connexion est réinitialisée.
  - SYN: Le Flag TCP SYN indique une demande d'établissement de connexion.
  - FIN: si ce drapeau est à 1 la connexion s'interrompt.
- Fenêtre (16 bits): Champ permettant de connaître le nombre d'octets que le récepteur souhaite recevoir sans accusé de réception
- Somme de contrôle (Checksum ou CRC): La somme de contrôle est réalisée en faisant la somme des champs de données de l'entête, afin de pouvoir vérifier l'intégrité de l'entête
- Pointeur d'urgence (16 bits): Indique le numéro d'ordre à partir duquel l'information devient urgente
- Options (Taille variable): Des options diverses

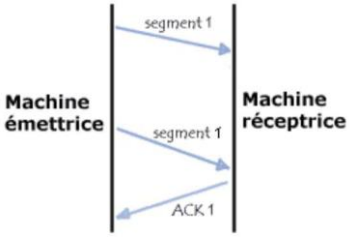
- Remplissage: On remplit l'espace restant après les options avec des zéros pour avoir une longueur multiple de 32 bits

Fiabilité des transferts

Le protocole TCP permet d'assurer le transfert des données de façon fiable, bien qu'il utilise le protocole IP, qui n'intègre aucun contrôle de livraison de datagramme.
En réalité, le protocole TCP possède un système d'accusé de réception permettant au client et au serveur de s'assurer de la bonne réception mutuelle des données. Lors de l'émission d'un segment, un numéro d'ordre (appelé aussi numéro deséquence) est associé. A réception d'un segment de donnée, la machine réceptrice va retourner un segment de donnée dont le drapeau ACK est à 1 (afin de signaler qu'il s'agit d'un accusé de réception) accompagné d'un numéro d'accusé de réception égal au numéro d'ordre précédent.



De plus, grâce à une minuterie déclenchée dès réception d'un segment au niveau de la machine émettrice, le segment est réexpédié dès que le temps imparti est écoulé, car dans ce cas la machine émettrice considère que le segment est perdu...



Toutefois, si le segment n'est pas perdu et qu'il arrive tout de même à destination, la machine réceptrice saura grâce au numéro d'ordre qu'il s'agit d'un doublon et ne conservera que le dernier segment arrivé à destination...

Etablissement d'une connexion

Etant donné que ce processus de communication, qui se fait grâce à une émission de données et d'un accusé de réception, est basé sur un numéro d'ordre (appelé généralement numéro de séquence), il faut que les machines émettrices et réceptrices (client et serveur) connaissent le numéro d'ordre initial de l'autre machine.
L'établissement de la connexion entre deux applications se fait souvent selon le schéma suivant:

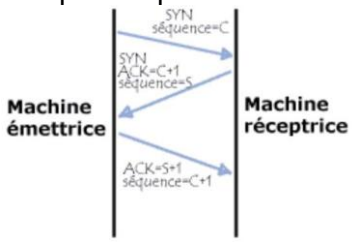
Les ports TCP doivent être ouverts
L'application sur le serveur est passive, c'est à dire que l'application est à l'écoute, en attente d'une connexion
L'application sur le client fait une requête de connexion sur le serveur dont l'application est en ouverture passive. L'application du client est dite "en ouverture active"

Les deux machines doivent donc synchroniser leurs séquences grâce à un mécanisme communément appelé threewayshandshake(poignée demain en trois temps), que l'on retrouve aussi lors de la clôture de session.
Ce dialogue permet d'initier la communication, il se déroule en trois temps, comme sa dénomination l'indique :

- Dans un premier temps la machine émettrice (le client) transmet un segment dont le drapeau SYN est à 1 (pour signaler qu'il s'agit d'un segment de synchronisation), avec un numéro d'ordre N, que l'on appelle numéro d'ordre initial du client
- Dans un second temps la machine réceptrice (le serveur) reçoit le segment initial provenant du client, puis lui envoie un accusé de réception, c'estàdire un segment dont le drapeau ACK est à 1 et le drapeau SYN est à 1 (car il s'agit là encore d'une synchronisation). Ce segment contient le numéro d'ordre de cette machine (du serveur) qui est le numéro d'ordre initial

du client. Le champ le plus important de ce segment est le champ accusé de réception qui contient le numéro d'ordre initial du client, incrémenté de 1

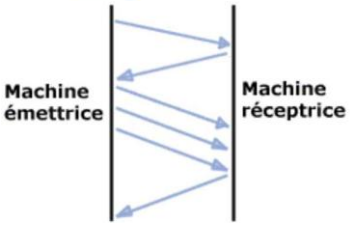
- Enfin, le client transmet au serveur un accusé de réception, c'est à dire un segment dont le drapeau ACK est à 1, dont le drapeau SYN est à zéro (il ne s'agit plus d'un segment de synchronisation). Son numéro d'ordre est incrémenté et le numéro d'accusé de réception représente le numéro d'ordre initial du serveur incrémenté de 1



Suite à cette séquence comportant trois échanges les deux machines sont synchronisées et la communication peut commencer!
Il existe une technique de piratage, appelée spoofing IP, permettant de corrompre cette relation d'approbation à des fins malicieuses

Méthode de la fenêtre glissante

Dans de nombreux cas, il est possible de limiter le nombre d'accusés de réception, afin de désengorger le réseau, en fixant un nombre de séquence au bout duquel un accusé de réception est nécessaire. Ce nombre est en fait stocké dans le champ fenêtre de l'entête TCP/IP.
On appelle effectivement cette méthode "méthodedela fenêtreglissante" car on définit en quelque sorte une fourchette de séquences n'ayant pas besoin d'accusé de réception, et celle ci se déplace au fur et à mesure que les accusés de réception sont reçus.



De plus, la taille de cette fenêtre n'est pas fixe. En effet, le serveur peut inclure dans ses accusés de réception en stockant dans le champ fenêtre la taille de la fenêtre qui lui semble la plus adaptée. Ainsi, lorsque l'accusé de réception indique une demande d'augmentation de la fenêtre, le client va déplacer le bord droit de la fenêtre.

Par contre, dans le cas d'une diminution, le client ne va pas déplacer le bord droit de la fenêtre vers la gauche mais attendre que le bord gauche avance (avec l'arrivée des accusés de réception).

Fin d'une connexion

Le client peut demander à mettre fin à une connexion au même titre que le serveur. La fin de la connexion se fait de la manière suivante :
Une des machines envoie un segment avec le drapeau FIN à 1, et l'application se met en état d'attente de fin, c'est à dire qu'elle finit de recevoir le segment en cours et ignore les suivants
Après réception de ce segment, l'autre machine envoie un accusé de réception avec le drapeau FIN à 1 et continue d'expédier les segments en cours. Suite à cela la machine informe

l'application qu'un segment FIN a été reçu, puis envoie un segment FIN à l'autre machine, ce qui clôture la connexion...