Cours 30 : TCP&UDP

Dans ce cours nous allons parler du fonctionnement de TCP&UDP ainsi que de leurs différences. Nous commencerons par voir comment fonctionne la couche 4 du modèle OSI, puis nous verrons rapidement TCP (Transmission Control Protocol), puis UDP (User Datagram Protocol) en dernier temps nous comparerons les deux.

Commençons donc par rappeler quelques fonctions de la couche 4 qui correspond à la couche Transport du modèle OSI.
La couche 4 permet le transfert de données entre les hôtes. Il encapsule les données avec une entête de couche 4 puis utilise les services des couches les plus basses (les couches 3, 2 et 1) pour transférer le message intact jusqu'à l'hôte voulue. Les hôtes eux mêmes ne font pas attention aux détails des sous couche du réseau, le transfert des données est transparent pour eux.

Une autre fonction de la couche 4 est qu'il fournit des services variés aux applications, ces services peuvent être la sûreté de transfert des données. Un autre service est la réparation des erreurs, dans le cas ou une erreur survient le message est redistribué. Un autre service est le séquençage de donnée, dans le cas ou les données transmises arrive en dehors des délais, il y a une sûreté pour que les hôtes puissent séquencer les données dans le bonne ordre.
Un service supplémentaire est le contrôle de flux, qui permet le contrôle de la vitesse de transfert des données en s'assurant que l'hôte ne reçoit pas de données en excès.
Les services énuméré ci dessus sont fournit par TCP et non pas UDP.

La couche 4 permet également une chose importante qui est de pouvoir fournir les numéros de ports (On ne parle pas ici des ports physique d'une interface) Il identifie ainsi la couche Application du protocole utilisé. Ceci permet également le multiplexage de session, qui sert à pouvoir délivrer plusieurs sessions en même temps sur un même ordinateur pour plusieurs services ou serveurs différents.

Les classement des ports sont désignés par l'IANA (Internet Assigned Numbers Authority)
- les ports connus ont des numéros de ports qui vont de 0 à 1023
- Les ports enregistrés ont des numéros de ports qui vont de 1024 à 49151
- Les ports éphémère/privée/dynamique ont des numéros de ports allant de 49152 à 65535

Maintenant qu'à été expliqué le fonctionnement de la couche 4, expliquons comment fonctionne TCP (Transmission Control Protocol)
TCP est orienté pour pouvoir établir une connexion, avant même d'envoyer les données à l'hôte en question, les deux hôtes vont établir une connexion. Une fois la connexion établie, les données commencent l'échange de données.
TCP fournit une communication fiable, les destination des hôtes sont assurés d'avoir reçu chaque segment TCP. Si le segment ne reçoit pas de confirmation (ackowlegment en anglais) il renvois le segment.
TCP fournit également le séquençage qui assure que l'hôte de destination place bien les segments dans le bonne ordre même s'ils arrivent en dehors du temps.
TCP fournit le contrôle de flux qui permet à l'hôte source d'augmenter/ralentir le taux de réception de donnée.

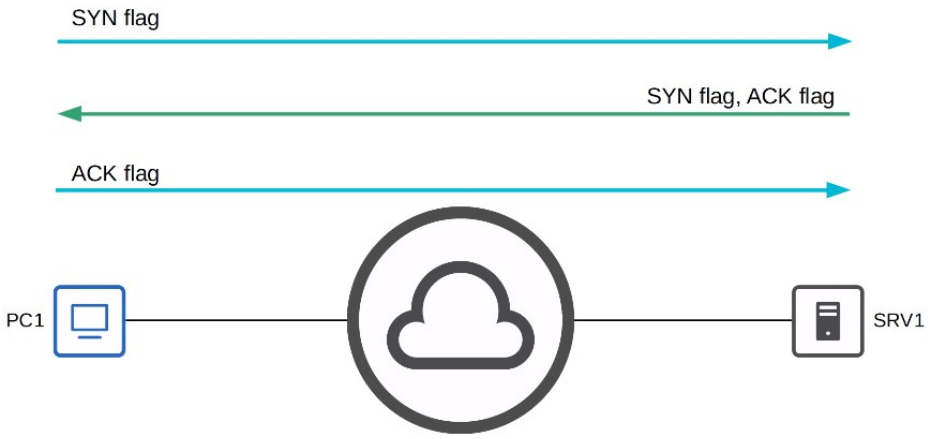
Voici une capture écran d'une entête TCP :

TCP segment header

Offsets	Octet	0								1								2								3							
Octet	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0				NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size														
16	128	Checksum																	Urgent pointer (if URG set)														
20	160	Options (if *data offset* > 5. Padded at the end with "0" bytes if necessary.)																															
...	...	...																															

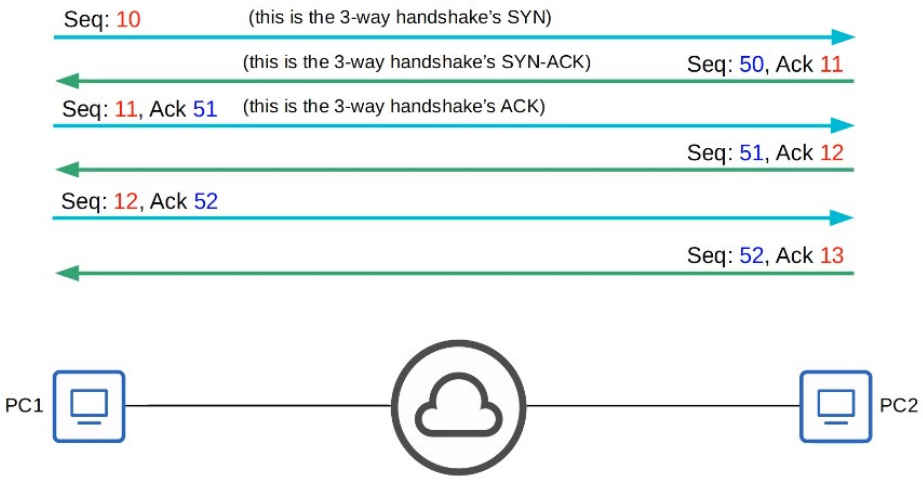
On peut voir tout d'abord les port source/destination utilisé chaque partie est d'une longueur de 16 bits.
On peut voir également la partie des flag. Les plus commun sont : ACK, SYN et FIN
Ces trois flags sont utilisés pour établir et terminer une connexion.

Expliquons à présent plus en détail comment la connexion est établie en utilisant le : « Three-Way Handshake » :
Sur ce schéma le PC1 veut établir une connexion avec le SRV1, pour cela il commence par envoyer un flag SYN, le SRV1 répond par un flag ACK la connexion est établie avec le ACK du PC1 envoyé.



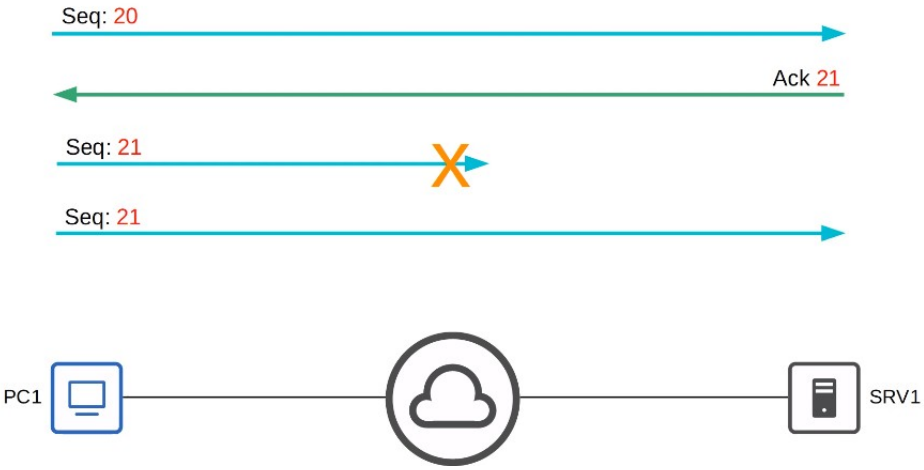
Une autre façon d'établir la connexion est par le moyen du « Four Way Handshake » ou ce sont cette fois 4 connexion nécessaires pour établir la connexion comme suit :

Voici la façon dont TCP permet le séquençage correct des données :



ce sont au départ des séquences de 10 (10 est pris pour exemple) puis au fur et à mesure des échanges le nombre de séquençage augmente de 1.

Le protocole TCP permet de s'assurer que la transmission est bien établie si ce n'est pas bien établie il renvoie le segment comme suit :



A présent que le fonctionnement de TCP est expliqué expliquons le fonctionnement de UDP.
UDP n'est pas orienté pour la connexion, les hôtes n'établissent pas de connexion avec l'hôte de destination avant d'envoyer les données. Les données sont juste envoyées.
UDP ne fournit pas une connexion fiable, Lorsque UDP est utilisé des confirmation (ou ackowledgement en Anglais) ne sont pas reçus. Si un segment est perdus UDP n'a pas de mécanisme pour le retransmettre. Les segment font de leur « mieux » pour être transmis.
UDP ne fournit pas de séquençage, il n'y a pas de nombre dans l'entête UDP. Si le segment arrive en dehors du délais, UDP n'a pas de mécanisme pour le renvoyer.
UDP ne fournit pas de contrôle de flux. UDP n'a pas de mécanisme de contrôle de flux des données comme TCP.
Voici l'entête utilisé pour UDP, image tiré de Wikipédia :

UDP datagram header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port														
4	32	Length																Checksum														

On peut voir qu'il ne contient que quatre partis.
Comparons à présent les deux mécanismes et dans quelles cas chacun est utilisé.
TCP fournit pour sa part beaucoup de fonctionnalités par rapport à UDP mais le coût de séquençage est surélevé.
Pour les applications qui requière une communication fiable (Par exemple télécharger un fichier), TCP est privilégié. (On veut s'assurer d'avoir le fichier en entier)
Pour les applications comme voix en temps réel/Vidéo en direct, UDP est privilégié.
Il y a certaines applications qui utilisent UDP mais qui fournissent de la fiabilité dans la connexion à l'application.
Certaine application utilisent TCP&UDP cela peut dépendre de la situation.

TCP	UDP
Orienté pour établir une connexion	Orienté transfert de donnée
Fiable	Non fiable
Séquençage	Pas de séquençage
Contrôle de flux	Pas de contrôle de flux
Utiliser pour télécharger etc..	Utilisé pour le direct (Streaming, VoIP etc..)

Voici une liste des numéro de ports protocoles les plus utilisés :

TCP
- FTP data (20)
- FTP control (21)
- SSH (22)
- Telnet (23)
- SMTP (25)
- HTTP (80)
- POP3 (110)
- HTTPS (443)

UDP
- DHCP server (67)
- DHCP client (68)
- TFTP (69)
- SNMP agent (161)
- SNMP manager (162)
- Syslog (514)

TCP&UDP
- DNS (53)