

UDP et TCP sont dans le modèle OSI les deux plus gros représentants des protocoles de la couche Transport.

On trouve très rapidement en quelques recherches internet que l'un est fiable et l'autre non (en étant plus rapide quand même). Mais pour mieux comprendre leurs différences, on peut regarder comment les deux sont organisés.

Pour cela on peut comparer les deux entêtes des protocoles qui sont un peu au coeur de ses protocoles.

(fig entête protocole)

La première chose qui souhaite aux yeux c'est la longueur relative de l'entête du protocole TCP.

On comprend ainsi déjà un peu mieux pourquoi UDP est plus rapide, car pour chaque paquet TCP/UDP, l'ordinateur devra consommer des ressources pour décrypter cette entête et la traiter. Mais on peut aussi noter les points communs (tout l'entête d'UDP ...) :

- Un port d'émission, pour renvoyer les messages au bon port
- Un port de destination, le nom est explicite
- Une somme de contrôle qui permet de vérifier l'intégrité des paquets (une sorte de hachage? )
- La longueur du paquet transmis

Penchons nous un peu plus sur TCP pour comprendre ce qu'il fait en plus. Tout d'abord TCP utilise une ouverture active en 3 temps (3 way handshake) : le client envoie une demande de synchronisation SYN, le serveur lui répond par SYN (synchronisation) / ACK (acknowledgement) et enfin le client confirme avoir reçu ce retour ACK.

Et ce phénomène de demande, acquittement sera utilisé tout au cours de la séquence de transmission (même à fermeture), afin de s'assurer que chaque paquet est bien reçu. On a aussi dans l'entête le numéro de séquence et le numéro d'acquiescement. Ces deux numéros sont utilisés pour suivre l'ordre des paquets envoyés / reçus, et donc détecter les pertes dans le réseau de paquets, ce qui permet alors de les renvoyer.

La fermeture aussi se fait en 4 temps avec les séquences FIN et ACK.

Enfin le protocole TCP incorpore des algorithmes qui permettent d'éviter la congestion du réseau, comme le slow start qui augmente au fur et à mesure le débit jusqu'à trouver la capacité actuelle du réseau, ou bien un contrôle des capacités des buffer émetteurs / récepteurs afin de ne pas les dépasser.

Tout ensemble fait que TCP est un protocole très fiable, cad sans perte de paquets et avec un contrôle de l'entité avec laquelle on échange (via l'ouverture)

Cependant comme UDP n'implémente aucun de ces protocoles, ce qu'il gagne en rapidité de traitement, il le perd en fiabilité. Il n'y a aucun mécanisme pour récupérer la perte de paquets avec UDP. De plus, comme n'importe qui peut envoyer les paquets qu'il veut via UDP, on peut surcharger un serveur de request UDP, ce qui va le saturer.

(figure udp flood)

Avec tout ça en tête, on voit qu'il n'y a pas de meilleur protocole, juste des protocoles plus ou moins adaptés selon l'usage. Pour des applications capable de tolérer la perte de données et ayant

un besoin de vitesse, UDP est parfait. Cependant, si l'intégrité des données est vitale à votre application, mieux vaut utiliser TCP. Un autre gros avantage d'UDP est qu'il permet de faire du multicasting, alors que TCP est par son ouverture uniquement réservé à de l'unicasting.