

2.4 – LA COUCHE TRANSPORT (COUCHE 4)

DESCRIPTION

- ▶ La couche transport
 - ▶ Elle traite des segments
 - ▶ Son rôle est de gérer la communication de bout en bout
- ▶ Exemple de mécanismes classiques
 - ▶ Client-serveur : paradigme de communication dans lequel un serveur attend que des clients se connecte
 - ▶ Multiplexage : différencier plusieurs communications d'un même client vers un même serveur
 - ▶ Contrôle de flux : un émetteur rapide ralentit quand il communique à un récepteur lent
 - ▶ Contrôle de congestion : l'émetteur ralentit si le réseau est surchargé

PLAN

- ▶ Le protocole UDP
- ▶ Le protocole TCP

2.4.1 – LE PROTOCOLE UDP

LE PROTOCOLE UDP

- ▶ UDP (*User Datagram Protocol*) est défini dans la RFC 768 (1980)
 - ▶ Il s'agit d'un protocole « non-connecté », dans le sens où il ne maintient pas d'information permanente sur une communication
- ▶ UDP ajoute par rapport à IP :
 - ▶ La notion de client et de serveur
 - ▶ Le multiplexage, grâce au mécanisme de ports

MULTIPLEXAGE PAR PORTS

- ▶ Le port est un entier utilisé pour identifier des flux au sein d'une communication entre deux entités
 - ▶ Port destination = permet de spécifier le service que le client souhaite accéder (ex : DNS port 53)
 - ▶ Port source = permet d'identifier l'application du client (quand plusieurs communications UDP partent du même client pour le même serveur) ; généralement un numéro aléatoire choisi par le système d'exploitation du client
 - ▶ Le numéro des ports source et de port destination sont stockés dans chaque paquet UDP

FORMAT D'UN SEGMENT UDP

- ▶ Entête UDP :
 - ▶ Port source (16 bits)
 - ▶ Port destination (16 bits)
 - ▶ Longueur (16 bits) : définit la longueur du segment UDP (entête et données)
 - ▶ Somme de contrôle (16 bits), optionnelle, et couvrant toute le segment UDP mais aussi une partie de l'entête IP

AVANTAGES ET INCONVÉNIENTS

- ▶ Avantages :
 - ▶ UDP est très simple (à implémenter et à exécuter)
 - ▶ UDP est rapide : il n'introduit pas de délai
- ▶ Inconvénients :
 - ▶ UDP n'est pas fiable : des segments (et donc des données) peuvent être perdus

EXEMPLES DE PROTOCOLES UTILISANT UDP

- ▶ TFTP (*Trivial FTP*) : port 69
- ▶ DHCP (*Dynamic Host Configuration Protocol*) : port 67
- ▶ SNMP (*Simple Network Management Protocol*) : ports 161 et 162
- ▶ DNS (*Domain Name System*) : port 53
- ▶ Jeux en ligne (ayant besoin d'être rapides)
- ▶ Système d'audio-conférences (ex : VoIP) et de visio-conférences

2.4.2 – LE PROTOCOLE TCP

LE PROTOCOLE TCP

- ▶ TCP (*Transmission Control Protocol*) est défini dans la RFC 9293
 - ▶ Il s'agit d'un protocole « connecté », dans le sens où il maintient des informations permanentes sur une communication
- ▶ TCP ajoute par rapport à IP :
 - ▶ La notion de client et de serveur
 - ▶ Le multiplexage, grâce au mécanisme de ports
 - ▶ La garantie de livraison = un segment envoyé sera reçu + les segments sont reçus dans l'ordre où ils sont envoyés + les segments dupliqués sont supprimés
 - ▶ Le contrôle de flux = l'émetteur adapte son débit au récepteur
 - ▶ Le contrôle de congestion = l'émetteur adapte son débit aux conditions réseaux

GARANTIE DE LIVRAISON

- ▶ TCP fournit une garantie de la livraison, qui se décline en trois propriétés
 - ▶ Les segments reçus sont transmis à la couche supérieure dans l'ordre d'envoi
 - ▶ Un segment reçu plusieurs fois n'est transmis qu'une fois à la couche supérieure
 - ▶ Un segment envoyé est reçu au moins une fois
- ▶ Pour les deux premières propriétés, on utilise un numéro (dit « numéro de séquence ») associé à chaque segment : on peut ainsi garantir l'ordonnancement et la non-duplication
- ▶ Pour la troisième propriété, on utilise des acquittements (avec un « numéro d'acquiescement ») et des retransmissions (avec des timers)
 - ▶ Remarque : les retransmissions peuvent causer des duplications, mais elles seront traitées par le récepteur

CONTRÔLE DE FLUX

- ▶ Chaque récepteur TCP dispose d'un buffer de réception, avec une taille maximale
- ▶ Chaque récepteur TCP informe en permanence l'émetteur correspondant de la taille restante dans son buffer
- ▶ Un émetteur TCP n'envoie jamais plus d'octets que la taille disponible dans le buffer du récepteur → ainsi, un émetteur rapide ralentit quand il parle à un récepteur lent

CONTRÔLE DE CONGESTION

- ▶ Chaque émetteur TCP maintient une valeur nommée « fenêtre de congestion » (CW = congestion window) qui lui indique combien de segments au maximum peuvent être envoyés
- ▶ A chaque acquittement reçu, la valeur de CW augmente « un peu » → ainsi, TCP essaye en permanence d'envoyer les données de plus en plus vite
- ▶ A chaque fois qu'un acquittement est perdu, la valeur de CW baisse « beaucoup » → l'hypothèse est qu'un acquittement perdu correspond à un réseau surchargé

MISE EN PLACE D'UNE CONNEXION

- ▶ Etablissement d'une connexion :
 - ▶ Les connexions TCP s'établissent selon le modèle client-serveur
 - ▶ Le client contacte le serveur (flag SYN), le serveur répond (flag SYN ACK), et le client confirme (flag ACK) : « *three-way handshake* »
 - ▶ Lorsque la connexion est établie, deux sens de communication sont ouvertes : du client vers le serveur, et du serveur vers le client

FONCTIONNALITÉS SUPPLÉMENTAIRES

- ▶ TCP agrège les acquittements
- ▶ TCP retarde légèrement les acquittements (de 200 ms)

FORMAT D'UN SEGMENT TCP

▶ Entête TCP :

- ▶ Port source (16 bits) et port destination (16 bits)
- ▶ Numéro de séquence (32 bits) et numéro d'acquittement (32 bits)
- ▶ Taille de l'entête (4 bits) en mots de 32 bits
- ▶ Réservé (4 bits)
- ▶ Drapeaux (8 bits) : CWR (*congestion window reduced*), ECE (*explicit congestion notification - echo*), URG (*urgent*), ACK (*acknowledgement*), PSH (*push*), RST (*reset*), SYN (*synchronization*), FIN (*finalization*)
- ▶ Taille du buffer de réception (16 bits)
- ▶ Somme de contrôle (16 bits) : obligatoire, et couvre une partie de l'entête IP et tout le segment TCP
- ▶ Pointeur urgent (16 bits) : index du dernier bit urgent
- ▶ Options (taille variable)

EXEMPLE D'OPTIONS TCP

- ▶ Pas d'opération : option utilisée pour le bourrage
- ▶ Fin de la liste d'options
- ▶ MSS (maximum segment size)
- ▶ Estampillage (*timestamp*) : utilisé pour mesurer le temps d'aller-retour
- ▶ SACK (*Selective acknowledgment*) : utilisé pour faire de l'acquittement sélectif (au lieu de l'acquittement cumulatif)
- ▶ etc.

AVANTAGES ET INCONVÉNIENTS

- ▶ Avantages

- ▶ TCP est fiable

- ▶ Inconvénients

- ▶ TCP est lent (exemple : acquittements retardés, bufferisation des données arrivées hors séquence si des données sont manquantes, etc.)

EXEMPLE DE PROTOCOLES UTILISANT TCP

- ▶ HTTP (*Hypertext Transfer Protocol*) : port 80
- ▶ FTP (*File Transfer Protocol*) : ports 21 (contrôle) et 20 (données)
- ▶ Telnet et SSH (*Secure Shell Protocol*) : ports 23 et 22
- ▶ DNS (*Domain Name System*) : port 53 (en TCP aussi)
- ▶ Transfert de fichiers ou de données volumineuses (dont vidéos)