

Master d'Informatique – 1ère année

Réseaux et protocoles

Couche transport

Bureau S3-354

[Mailto:Jean.Saquet@unicaen.fr](mailto:Jean.Saquet@unicaen.fr)

<http://saquet.users.greyc.fr/M1/rezopro>

Couche transport : fonctions

Le but des réseaux est de permettre la communication entre applications tournant sur deux (ou plus) machines reliées au réseau.

==> communication "de bout en bout" entre deux applications.

Les applications utilisent des services facilitant ces communications : couche transport.

Principaux services :

- adressage complémentaire des applications
- qualité du service de communication

Couche transport et X25

Lorsque la couche réseau assure déjà un mode connecté, une fiabilité des échanges, il reste peu de choses à faire au niveau transport.

Le niveau transport doit assurer une connexion de bout en bout (au niveau X25, ce n'est pas le cas à cause des nos de VL différents, et de la segmentation).

Le modèle OSI prévoit plusieurs classes de couche transport (TP0 à TP4), selon les caractéristiques et performances du réseau.

Couche transport et IP

IP est non fiable, non connecté. La couche réseau devra donc offrir des fonctions adéquates pour apporter des solutions aux problèmes liés.

Le niveau transport du modèle IETF :

- permet un adressage complémentaire pour faire communiquer entre elles des tâches tournant sur les machines connectées (numéros de port).
- offre le choix entre un mode non connecté et non fiable (UDP) pour certaines applications simples, et un mode connecté et fiable (TCP).

Numéros de port

C'est un adressage complémentaire permettant à un port d'une machine (ayant une adresse IP) de communiquer avec un port d'une autre machine, reliée au même inter-réseau IP.

Les applications désirant communiquer (clients ou serveurs) devront chacune être attachée à un numéro de port (au moins).

Chaque machine "possède" 65536 nos de ports pouvant servir pour cela.

Ces nos de port seront utilisés en UDP ou en TCP.

Gestion des ports

Les n° de port < 1024 sont réservés comme ports "bien connus" des diverses applications (ex : http=port 80). Les serveurs les utilisent, et sont donc "à l'écoute" sur ces ports. Un utilisateur ordinaire ne peut pas les utiliser. Une application cliente demande un n° de port disponible au système en fonction de ses besoins. Ce sont les ports "utilisateurs" de $n° \geq 1024$.

On peut bien entendu écrire un programme communiquant avec des nos de ports utilisateurs quelconques, qui devront être libres au lancement de ce programme.

UDP

Le datagramme UDP ajoute simplement les nos de port (émetteur, récepteur) aux données constituées du datagramme IP.

Un champ longueur et un champ checksum permettent toutefois de contrôler l'intégrité du datagramme.

UDP est utilisé par des applications simples, ne nécessitant en général qu'un seul échange requête / réponse, utilisant chacune un seul datagramme.

Exemples : Date et heure, écho UDP, DNS (cf. plus loin)

TCP (1)

TCP est le protocole de transport fiable du modèle IETF. Il utilise le mode connecté, ainsi que la numérotation des paquets (appelés segments) et des acquittements et timers.

Il est utilisé par la plupart des applications de l'Internet nécessitant un transfert d'information fiable (web, mail, transferts de fichiers, ...)

Toutefois certaines applications gèrent elles-mêmes les problèmes de fiabilité et utilisent donc essentiellement UDP (ex : système de partage de fichiers NFS).

TCP (2)

Phase de connexion :

L'appelant envoie un segment avec le bit SYN

L'appelé répond avec SYN + ACK

L'appelant renvoie un segment ACK.

Raison : ouverture dans les deux sens

Les nos de séquence sont initialisés à 0 et l'échange de données peut commencer.

Pour fermer la connexion, des segments FIN, FIN+ACK et ACK sont utilisés (fermer la connexion dans les deux sens).

TCP (3)

TCP doit assurer la remise en ordre des segments. Il y a donc un buffer de réception, avec pour paramètre la "largeur de fenêtre" (champ du segment TCP). Les techniques utilisées sont semblables à celles d'une couche 2, TCP utilise la technique des "fenêtres glissantes". Voir en TD le principe plus détaillé. Les acquittements portent le no du prochain octet attendu. Le réglage des timers est un point délicat, car on ne connaît pas les caractéristiques du chemin emprunté (qui peut de surcroît varier)

ICMP

C'est le protocole de gestion et de contrôle d'IP

Ce n'est pas un protocole de niveau transport, il comporte des fonctions relatives au fonctionnement d'IP sur les machines.

==> pas de nos de port, mais des codes de fonction.

Exemple : 8 = demande d'écho en v4

ICMP comporte :

- des commandes telles que demande/réponse d'écho
- des signalisations d'erreur, p.e. destination inaccessible
- des fonctions de dialogue avec les routeurs
- Des fonctions de gestion de contrôle de flux

En Ipv6, ICMP intègre le multicast (cf. cours sur le sujet)

ICMP (2)

- Demande et réponses :

Ne pas confondre la demande d'echo ICMP avec le service d'echo UDP.

La synchronisation d'horloge évite des pbs de confusion entre les systèmes.

- signalisation d'erreurs :

Les erreurs signalées portent sur les echecs de routage, le TTL dépassé, l'inaccessibilité de port, les interdictions dues au filtrage, ...

- demandes aux machines :

ICMP permet de demander une limitation d'envoi à la source en cas de congestion de routeur.

Il peut également demander des modifications de route

ICMP (3)

Dialogues entre machine et routeurs :

ICMP comporte des mécanismes tels que demande de masque de sous-réseau, découverte de routeurs qui permettent aux machines d'adapter automatiquement leurs tables de routage.

Les routeurs peuvent envoyer périodiquement des messages d'annonce, ou bien être sollicités par les machines.

Ipv6 utilise largement ce type de service.

Empilement des couches



Message de niveau application (ex http)

Appel couche transport, port dest 80

Demande port source utilisateur (exemple 3145)

Datagramme IP : adresses IP, TTL, **indication de la couche transport**

ARP et obtention de l'adresse dest. MAC, création de la trame Ethernet, **couche sup IP**

En réception, chaque couche vérifie les infos la concernant,
et transmet le reste à la couche supérieure (grâce au champ
"protocole" ou au no de port)