

S52 – Réseaux

Réseaux : Couches Transport et Application

E. Bruno – bruno@univ-tln.fr

USTV

Septembre 2012

- 1 La couche transport
 - Problématique
 - Deux protocoles de transport pour Internet : UDP et TCP
 - UDP – User Datagram Protocol
 - TCP - Transmission Control Protocol



La couche transport : problèmes

- Type de services :
 - De même que pour les services au niveau de la couche réseau il y a les services de transport :
 - sans connexion
 - orientés connexion (établissement d'une connexion,...)
- Pourquoi ?
 - Que se passe-t-il si le fournisseur de service offre un service non fiable ?
 - Que se passe-t-il si un des routeurs tombe en panne ?
 - Que se passe-t-il si des paquets sont perdus ? Que se passe-t-il si une connexion se termine de manière non prévisible ?
 - Les utilisateurs n'ont aucun contrôle sur le réseau

Fonctions de la couche transport

- Adressage : Accès à l'application pour l'établissement/fermeture d'une connexion
- Segmentation : Découper un message en paquets
- Contrôle de flux et tampons : pour éviter les débordements
- Multiplexage et démultiplexage
- Recouvrement d'erreurs (IP n'est pas fiable)

Transport et adressage

- Une application désireuse d'ouvrir une connexion avec une autre application distante, doit définir " l'adresse " de la couche transport où écouter les communication
- Différence avec IP : applications vs. machines
- Quelle est l'origine du message ?
- Quelle est la destination finale ?
- Il peut s'agir d'un processus au sens large :
 - Application
 - Tâche système



Établissement et fin d'une connexion (2)

- Obtenir une fiabilité pour les connexions asymétriques est théoriquement impossible.
- Cf. le problème des deux armées :
 - Les compagnie 1 et 2 de l'armée Bleue sont de part et d'autre de l'armée Rouge
 - L'armée Bleue gagne à coup sur si elle possède un protocole d'envoi de message pour se coordonner
 - Le message peut être éliminé par l'armée Rouge
 - Comme la compagnie 1 peut être sûre que la compagnie 2 sait ?
 - Par envoi d'une confirmation ? Mais alors comment la compagnie 2 sait que la compagnie 1 est sûre ?

Partie commune à TCP et UDP

- Orientation application vs orientation machine (cf.IP)
- Les sockets : identification d'une application
 - Adresse IP d'une machine
 - Numéro de port
- Un couple de sockets définit une connexion TCP ou un échange UDP
 - 10.1.73.26 :23 et 10.1.73.58 :1094
 - Connexion entre un processus client à partir du port 1094 de la machine 10.1.73.58 et un démon telnet sur la machine 10.1.73.26 (port 23).

Les ports réservés à TCP/UDP

- Comment un hôte peut-il savoir si un numéro de port est utilisé par une machine distante ?
- Deux approches :
 - Fixer des ports standards
 - Attribuer dynamiquement les ports
- Les protocoles Internet adoptent une approche mixte

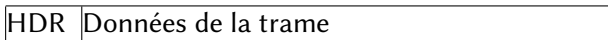
Les ports réservés à TCP/UDP (2)

- Exemples de ports réservés
 - 7 ECHO Ping d'une station (à quel niveau ?)
 - 21 FTP File Transfer Protocol
 - 53 DOMAIN Domain name server

UDP

- User Datagram Protocol Service simple sans connexion
- Simple en-tête ajouté aux paquets IP

Encapsulation UDP

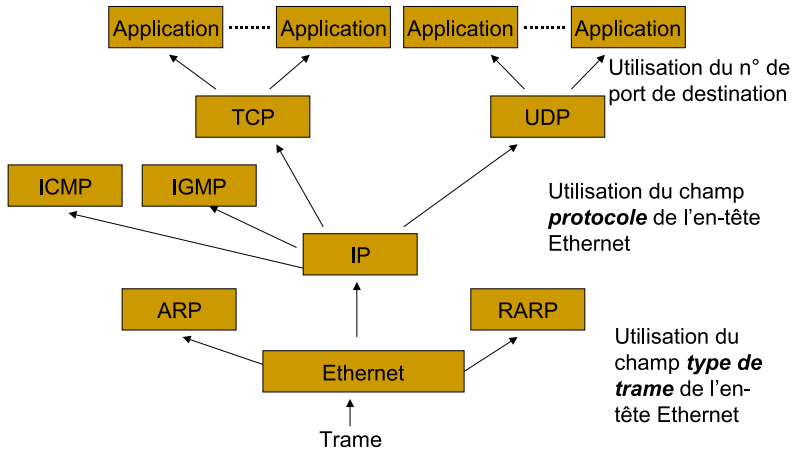


- La couche UDP différencie plusieurs sources et destinations sur un hôte
- IP gère le transfert entre les hôtes
- La couche physique transfère une trame sur le réseau physique

Pseudo en-tête et en tête UDP

IP Source		
IP Destination		
Zéro	Protocole	Longueur UDP
Port UDP source		Port UDP de destination
Longueur du message UDP		Somme de contrôle
Données		
Octet de bourrage		

Le multiplexage et le démultiplexage



Ports UDP

No. Port	Nom	Description
7 - 0x07	ECHO	Echo
11 - 0x0B	USERS	Active Users
13 - 0x0D	DAYTIME	Daytime
37 - 0x25	TIME	Time
42 - 0x2A	NAMESERVER	Host Name Server
53 - 0x35	DOMAIN	Domain Name Server
67 - 0x43	BOOTPS	Boot protocol server
68 - 0x44	BOOTPC	Boot protocol client
69 - 0x45	TFTP	Trivial File transfert protocol
123 - 0x7B	NTP	Network Time Protocol
161 - 0xA1	SNMP	Simple Network Management protocol

File d'attente et UDP

- L'O/S crée une file d'attente par port
- La taille de cette file est spécifiée et changée par l'application
- Quand UDP reçoit un datagramme, il contrôle le numéro de port de la destination avec la liste des ports actifs en cours d'utilisation
- En cas d'erreur un message ICMP “ port unreachable error ” est envoyé

UDP et la fragmentation IP

- IP négocie avec l'interface locale la MTU (Maximum transfert unit)
- Si le datagramme à une taille supérieure, IP fragmente, et seule la destination réassemble.
- En changeant de réseau un fragment de datagramme IP (paquet) peut être à nouveau fragmenté
- **Attention : Si un paquet IP est perdu tout le datagramme UDP doit être retransmis.**

UDP - Conclusion

- UDP est un protocole simple
- Il fournit (au dessus de IP)
 - Numéro de port
 - Somme de contrôle optionnelle
- IP vérifie uniquement les en-têtes, UDP vérifie aussi les données
- Compatible avec la fragmentation IP

Les caractéristiques de TCP

- Le protocole de transport de référence pour Internet
- TCP fournit aux applications
 - Une communication point à point
 - Un mode connecté
 - Une orientation “ flux de données ”
 - L’ouverture et la fermeture des connexions propre et fiable
 - Une communication en “ full duplex ”
 - Une interface applicative
- Ne supporte pas le multicast

TCP un service de transport fiable

- La fiabilité est parfois fondamentale même en dépit de la performance et de la simplicité mais ce n'est pas le seul cas
 - Écriture sur un système de fichier (NFS)
 - Diffusion multimédia “ temps réel ”
- Simplicité de la réception des des données
 - Dans le bon ordre
 - Sans perte, ni duplication

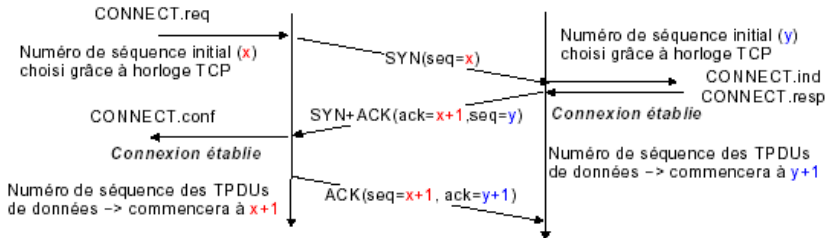
Les ports TCP

No. port	Nom	Description
20 - 0x14	FTP-DATA	File Transfer [Default Data]
21 - 0x15	FTP File	Transfer [Control]
23 - 0x17	TELNET	Telnet
25 - 0x19	SMTP	Simple Mail Transfer Protocol
37 - 0x25	TIME	Time
42 - 0x2A	NAMESERVER	Host Name Server
43 - 0x2B	NICNAME	Who Is
53 - 0x35	DOMAIN	Domain Name Server
79 - 0x4F	FINGER	Finger
80 - 0x50	HTTP	Hyper Text Transfert Protocol (WWW)
110 - 0x6E	POP3	Post Office Protocol - Version 3
111 - 0x6F	SUNRPC	SUN Remote Procedure Call

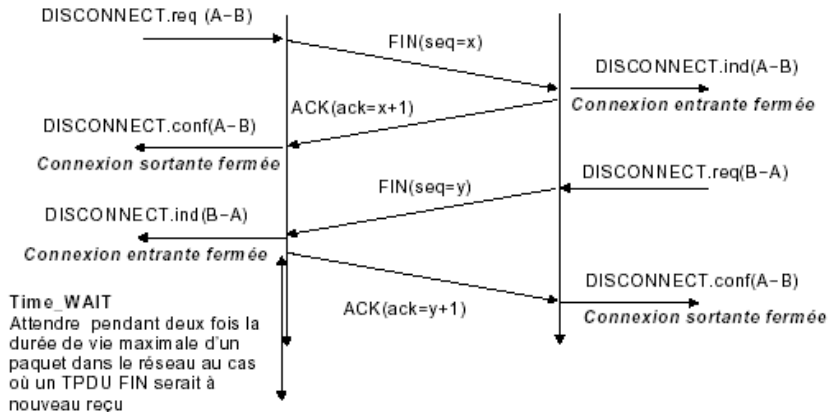
L'en-tête TCP

Port TCP de la source				Port TCP de la destination				
Numéro de séquence								
Numéro d'acquittement								
Longueur Du HDR 4 bit	Réservé 6 bit	SYN	FIN	RST	URG	ACK	PSH	Taille de fenêtre
Somme de contrôle TCP						Pointeur urgent		
Option et données ...								

Ouverture de connexion TCP



Fermeture de connexion TCP



Transfert fiable

- Chaque segment TCP contient
 - 16 bits checksum : utilisé pour détecter les erreurs de transmission sur l'en-tête et le contenu
- sequence number (un octet consomme un numéro séquence)
 - utilisé par l'émetteur pour délimiter les segments transmis
 - utilisé par le receveur pour réordonner les segments reçus
- acknowledgement number utilisé par le receveur (si ACK est vrai) pour annoncer à l'émetteur le numérode séquence du prochain octet attendu
- Comment faire face aux pertes de segments ?
 - protéger chaque segment par un temporisateur
 - si le temporisateur expire avant la réception de l'acquittement correspondant, retransmettre

Les acquittements

- Ils peuvent être transportés avec les données
- Ils acquittent un nombre d'octets de données reçus
 - Pas forcément un segment entier
 - L'acquiescement peut être retardé (avec un timeout)
- Problème des paquets arrivés en “ désordre ”
- Retransmission
 - A l'envoi d'un paquet un “ timer ” est déclenché
 - Adaptation automatique du délai d'acquiescement
 - “ Segment Round Trip Time ”
- TCP s'adapte sans paramétrage, à tous les débit et à tous les temps de réponse, et donc à tous les réseaux.

Retransmission adaptative

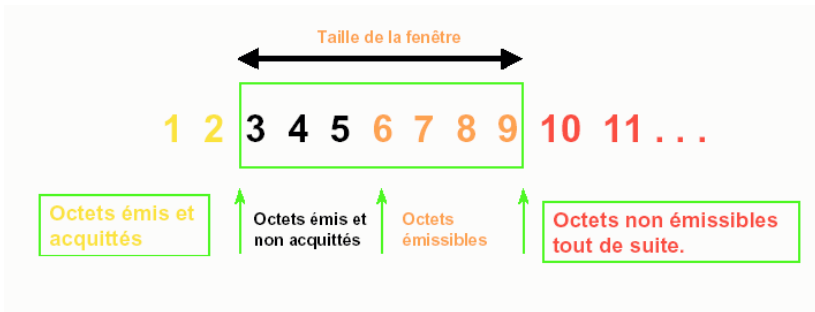
- Les paquets IP peuvent être perdus
- TCP attend des acquittements
- A l'envoi, un compteur est déclenché, en cas d'expiration avant l'acquittement les données sont retransmises
- Problème difficile : Quelle valeur pour le timer ?
 - RTT : temps d'aller retour, α coefficient de lissage

$$RTT_i = \alpha * RTT_{i-1} + (1 - \alpha)RTT_{mesuree}$$

$$Timeout = \beta * RTT, \beta > 1 (\beta = 2)$$

Le fenêtrage

- Mécanisme de fenêtre glissante
- L'émetteur peut envoyer plusieurs paquets avant de recevoir un acquittement
- La réception d'un acquittement décale la fenêtre



Contrôle de flux dynamique (1)

- Le destinataire dispose d'une place limitée (buffer)
- La quantité de buffers allouées à une connexion transport peut varier dynamiquement
- Le contrôle du flux et de la congestion est indispensable à Internet
 - Hétérogénéité des machines : Les réseaux et les routeurs ont des capacités différentes
 - Réseau dynamique
- La perte de segments est interprétées comme un signe de congestion
 - Utilisation du "slow start" : on réduit la fenêtre de congestion

Contrôle de flux dynamique (2)

- Le receveur doit annoncer la taille de buffer disponible à l'émetteur
- il indique la fenêtre de réception (rwin) dans chaque segment d'acquittement

Contrôle de flux dynamique (3)

- La fenêtre TCP est encodée dans un champ de 16 bits dans l'entête du segment TCP
- La taille maximale de la fenêtre TCP : 65535 octets
- Après avoir transmis toute une fenêtre de segments, une entité TCP doit s'arrêter et attendre le retour des acquittements

Fonctionnement résumé de TCP

- Établir une connexion
- n fois
 - Transférer des données
 - et/ou envoyer des acquittements
 - et/ou modifier la taille des fenêtres
- Fermer la connexion

Le contrôle de congestion (1)

- La retransmission avec mécanismes de timer dans TCP permet d'obtenir du contrôle de flux et ainsi d'améliorer la fiabilité des systèmes.
- Cependant TCP doit réagir aux problèmes de congestion
- Les congestions sont la résultante de délais importants causés par une surcharge de datagrammes dans un ou plusieurs nœuds du réseau (Les routeurs).
- Puisque les routeurs ont une capacité de stockage finie et que les datagrammes se disputent le stockage, un routeur peut épuiser sa capacité et commencer à perdre des paquets.

Le contrôle de congestion (2)

- Que fait TCP ?

- Les points terminaux des communications ne savent pas où sont les points de congestion ni la raison, car congestion = augmentation du délai.
- La plupart des protocoles de transport utilisent des mécanismes de timeout et de retransmission, ils réagissent ainsi aux extensions de délais.
- De telles retransmissions aggravent la situation et s'ils continuent à envoyer des paquets sans s'assurer de la cause, peuvent entraîner un écroulement du réseau.
- Pour éviter les écroulements du aux congestions, TCP a été conçu pour limiter automatiquement le nombre de segments qu'il transmet sur l'Internet.
- “ Slow start ” : (au démarrage augmentation progressive de la taille de fenêtre)
- “ décroissance ” :
 - perte de paquet : division de la fenetre par deux
 - pas de perte : accroissement uniquement additif $w_i = w_i - 1 + \beta$

Le contrôle de congestion (3)

- TCP manipule les fenêtres pour gérer le contrôle de flux.
- La taille de ces fenêtres permet de limiter la quantité de paquets émis sur le réseau.
- Sur les réseaux filaires, les pertes de paquets dues aux erreurs de transmissions sont rares, aussi TCP assimile les pertes de paquets à des problèmes de congestion et non à des problèmes de liens à faibles performance
- Quand TCP détecte une congestion au niveau du récepteur il ajuste la taille de la fenêtre récepteur (liée à la taille du tampon de réception) pour éviter les débordements.
- Pour gérer les congestion dans le réseau, TCP utilise une deuxième fenêtre appelée la fenêtre de congestion.
- Le nombre d'octets qui peut être transmis est le minimum entre les deux fenêtres.