

Systèmes et Réseaux (ASR 2) - Notes de cours

Cours 10

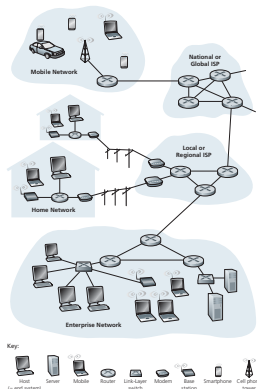
Anne Benoit
 April 8, 2015

PARTIE 1: Systèmes
PARTIE 2: Réseaux

1 Architecture des réseaux de communication

Réseaux: Internet, téléphonie, ... Tous organisés de la même façon (en couches). On s'intéresse à Internet. Concepts similaires dans les autres réseaux, avec une terminologie différente.

Approche concrète:



Approche fonctionnelle: accès à des applications distribuées, échange de données: Navigation Web, Mail, Messagerie instantanée, Téléphonie, Serveur ftp, Jeux, Pair-à-pair, ...

Les différents types de réseau

$\leq 1m$	circuit imprimé	} Multiprocesseur
	ordinateur	

10m	salle	} LAN (Local Area Network): réseau local
⋮	bâtiment	
1km	campus	
10km	ville	} MAN (Metropolitan): relie des LANs
100km	région	
1000km	continent	} WAN (Wide): relie des MANs et des LANs
10000km	terre entière	

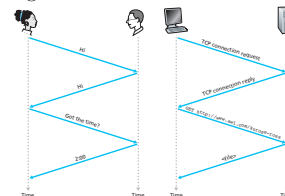
Internet

Deux types de service:

- Fiable, orienté, avec connexion
- Non fiable, sans connexion

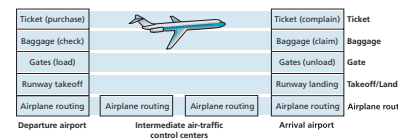
Pas de service garantissant la durée d'un transfert.

Protocole. Ce sont les protocoles qui gouvernent toutes les communications d'Internet. Analogie avec le contact humain: demander l'heure, poser une question. Protocoles pour gouverner toutes les communications d'Internet.



1.1 Modèle en couches d'Internet

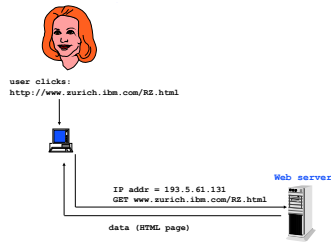
Exemple du réseau aérien:



Les couches d'un réseau TCP/IP:

5	Application	} Communication
4	Transport	
3	Réseau	} Interconnection
2	Liaison	
1	Physique	

- Couche 5-appli: aide les personnes et les machines à communiquer. Protocoles= ensemble de règles et de messages (Ex HTTP). Cas le plus simple avec 2 ordis:



• Couche 4-transport: Aide la couche application, avec une interface de programmation (évitte à l'utilisateur de répéter les mêmes tâches). Deux protocoles principaux dans TCP/IP:

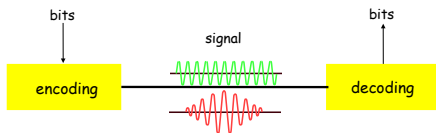
- UDP (téléphonie, visioconf...). Info transportée = message; non fiable (un message peut être perdu); pas de garantie de séquence.
- TCP: fiable: si une donnée est perdue, on la retransmet. Garantie de séquence. En plus, contrôle de flux/de congestion. Mais unité d'information = octet, données éventuellement groupées différemment à la destination qu'à la source.

Une application peut utiliser TCP ou UDP selon ses besoins.
Interface de programmation: *socket API*.

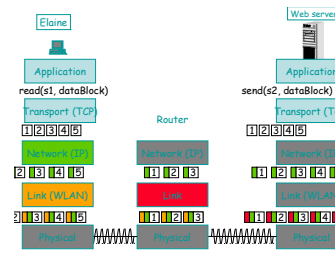
- Couche 3-réseau: Fournit la connectivité:
 - Utilisation de "routeurs": systèmes intermédiaires non visibles pour l'utilisateur.
 - Communication par paquets: l'information est coupée en morceaux (1500B), et des adresses IP sont utilisées pour communiquer.

Coeur du réseau (liens entre routeurs): communication par paquets ou commutation de circuits, voir Section 1.2.

- Couche 2-liaison: Adresses MAC en plus des adresses IP, et protocoles ALOHA, CSMA, ...
- Couche 1-physique: Encodage des bits en signal physique (signaux électromagnétiques).



Exemple d'utilisation des couches:

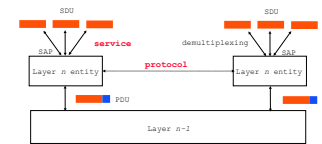


Protocoles et autres définitions. Couche n:

- communique avec les autres entités de couche n en utilisant des PDUs (*Protocol Data Unit*);
- utilise les services de la couche n - 1 et offre des services à la couche n + 1: SDUs (*Service Data Unit*).

SAP: *Service Access Point*: interface entre deux couches.

Protocole: règles entre entités de même couche.



Fonctions des couches.

- Détection d'erreurs (améliorer la fiabilité)
- Contrôle de flux (éviter la saturation par PDUs)
- Segmentation puis réassemblage (découpe en paquets)
- Multiplexage (partage d'une même connexion de niveau inférieur)
- Etablissement de connexion

Bilan sur les couches (et PDUs correspondant):

5-appli (Message): executer des applications (HTTP, SMTP, FTP).

4-transport (Paquet/segment): UDP et TCP.

3-réseau (Datagramme): IP, protocoles de routage.

2-liaison (Trame): Ethernet: transmet le paquet de noeud en noeud (routeurs). On véhicule des trames entières

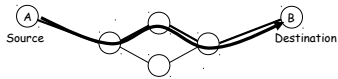
1-physique (Bit): Véhicule les bits de chaque trame, dépend du support physique (fil en cuivre à paire torsadée, câble coaxial, fibre optique, ...)

1.2 Techniques de base: commutation de circuits vs communication par paquets

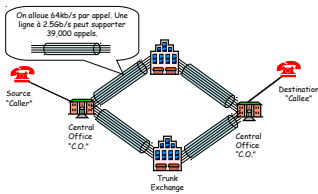
A relié à B par un réseau.

1.2.1 Commutation de circuits

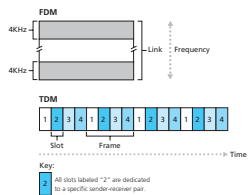
Si A veut appeler B, il faut établir un circuit, communiquer, puis fermer le circuit.



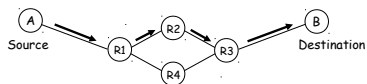
Réseau téléphonique commuté: 1 câble (physique ou virtuel) entre A et B, le débit est garanti de bout en bout.



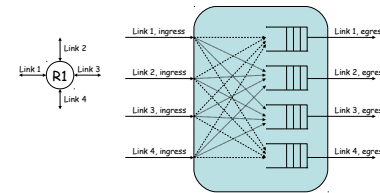
Multiplexage de fréquence vs multiplexage temporel.



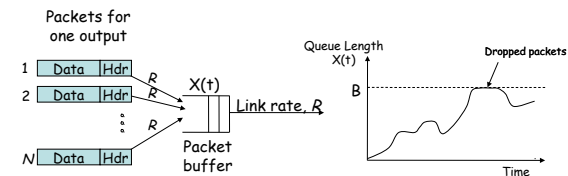
1.2.2 Communication par paquets



Pas de connexion, tables de routage pour aiguiller les paquets. Chaque paquet est indépendant: différents paquets peuvent prendre des routes différentes. Plusieurs paquets peuvent arriver en même temps et vouloir sortir sur le même lien: besoin de buffers au niveau des routeurs.



Multiplexage statistique: le buffer absorbe des rafales temporaires, le lien de sortie n'est pas obligé d'opérer au débit NR . Mais le buffer ayant une taille finie B , des pertes sont possibles.



Si A et B envoient avec un débit max x , le taux de A+B est $c < 2x$, d'où un gain de $2x/c$.

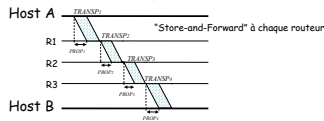
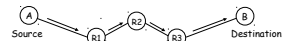


Internet emploie la communication par paquets: utilisation efficace de liens coûteux, et résistance aux pannes (de liens ou de routeurs).

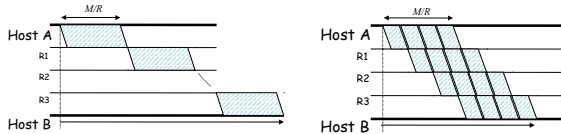
Performance de la communication par paquets. On note:

- M la taille d'un paquet, en bits;
- L la longueur d'un lien, en mètres;
- R le débit, en bits/secondes (b/s);
- D le délai de propagation, i.e., le temps de traversée d'un bit sur un lien de longueur L : $D = L/c$ (noté *PROP* sur les figs);
- $TRANS$ le temps de transmission, i.e., le temps pour transmettre un paquet de taille M : $TRANS = M/R$ (noté *TRANSP* sur les figs);
- La latence est le temps entre l'émission du premier bit et la réception du dernier bit. Sur un lien, $latence = D + TRANS$.

Ci dessous, la latence de bout en bout avec 3 routeurs pour aller de A à B est $\sum_i (L_i/c + M/R_i)$.



Pourquoi ne pas envoyer tout le message en un seul paquet?



Décomposer un message en paquets permet d'effectuer des envois en parallèle sur tous les liens, de réduire la latence de bout en bout, et d'éviter qu'un message s'accapare un lien.

- Calculer le temps de transmission dans les deux scénarios: message non découpé, ou bien message est découpé en k paquets de taille $S = \frac{M}{k}$, et chaque routeur attend d'avoir terminé la transmission d'un paquet avant d'envoyer le suivant (et liens identiques).

Dans le premier cas, le temps est $n(D + M/R)$.

Par paquets: Temps pour envoyer le premier paquet: $n \left(D + \frac{S}{R} \right)$.

Temps pour envoyer les $k - 1$ paquets restant: $(k - 1) \left(D + \frac{S}{R} \right)$,

soit un temps total de $T(S) = \left(n - 1 + \frac{M}{S} \right) \left(D + \frac{S}{R} \right)$.

- Quelle est la taille optimale des paquets?

On remarque que $T(S) = \frac{n-1}{R}S + MD\frac{1}{S} + \text{constante}$. Pour minimiser cette fonction, qui est de la forme $aS + b/S + c$, on utilise le théorème de la chèvre: un enclos de surface constante ($aS \times b/S = ab$) a un périmètre ($aS + b/S$) de taille minimum lorsque c'est un carré: $aS = b/S$, soit $S = \sqrt{b/a}$. (Vous pouvez aussi dériver la fonction pour vous en convaincre).

On obtient finalement $S_{opt} = \sqrt{\frac{MDR}{n-1}}$.

Todo: calculer $T(S_{opt})$ et observer son ordre de grandeur.

- Todo:** Quels sont les avantages et inconvénients de cette méthode par rapport à l'envoi en un seul paquet?