

# Complément sur les Réseaux

## Plan : - Introduction

- Définitions
- Techniques de commutation.
- Mécanismes mis en œuvre dans un réseau.

## - Transmission des données

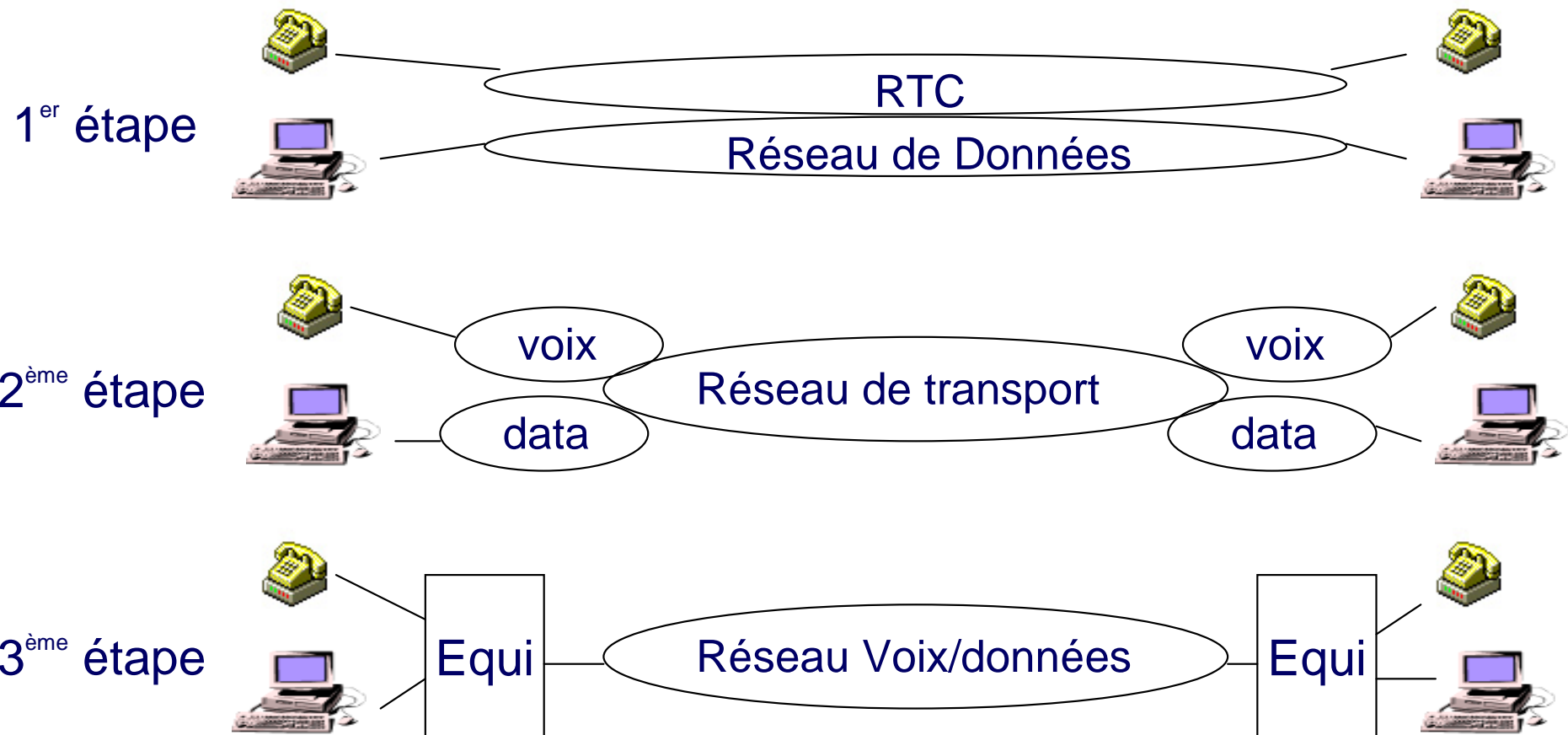
- Quantification du trafic
- Multiplexage et Optimisation

## - Quelques protocoles hauts débits

- IEEE 802.6 alias DQDB
- ISO 9314 alias FDDI
- Frame relay
- ATM ...

# Définition et Objectif

Un réseau est un ensemble de moyens informatiques géographiquement dispersés destinés à offrir un service ou assurer le transport des données.





# ***Critère pour un réseau***

- **Etendue géographique**
  - PAN, LAN, MAN , WAN
  - WPAN, WLAN, WMAN, WWAN
- **Type**
  - réseau privé
  - réseau public
- **Modes de diffusion**
  - réseau de diffusion  $1 \rightarrow n$
  - réseau de collecte  $n \rightarrow 1$
  - réseau de commutation  $1 \leftrightarrow 1$

# Topologie (1)

## ❏ Différence entre *topologie physique et logique*

## ❏ Topologie physique en bus.

- Un câble backbone unique qui est terminé aux deux extrémités.
- Bon rapport performance/prix
- Insertion d'une nouvelle station très simple sur le backbone
- Longueur du bus limité -> affaiblissement du signal
- Câble coupé -> plus de réseaux

## ❏ Topologie physique en étoile ou en étoile étendue.

- Tous les sites sont reliés à un nœud central
- Tous les messages transitent par ce nœud central
- Le nœud central agit comme un filtre
- Problème si panne du nœud central

# ***Topologie (2)***



## **Topologie physique en anneau.**

- Chaque hôte est connecté à son voisin
- L'information circule dans un seul sens
- Le message est lu et régénéré par chaque station
- Débit élevé et grande distance possible
- Pour augmenter la sécurité, double anneau



## **Topologie physique hiérarchique ou arborescente.**

- Dérivés des réseaux en étoile étendue et reliés entre eux par des ordinateurs ou routeurs



## **Topologie maillée**

- Utilisation de commutateurs
- Très performant au niveau des pannes



## **Topologie logique : broadcast ou passage de jeton**

# ***Equipement d'un réseau***

## **Couche Physique :**

- Modem, carte CSU/DSU (channel service Unit/ Data Service Unit)
- Répéteur
- Concentrateur ( HUB) ou répéteur multiport

## **Couche Liaison de données**

- NIC ( Network Interface Card) == carte réseau
- Pont
- Commutateur (switch) ou Pont multiport

## **Couche Réseau**

- Routeur, commutateur multi-couches (layer3)

## **Couche Application**

- Passerelle, Proxy

# Commutation (1)

## ❏ But :

- Relier les différents abonnés entre eux en minimisant les coûts
  - > Partage des ressources

## ❏ Commutation de circuits.

- Création d'un lien physique par juxtaposition de différents supports physiques afin de constituer une liaison de bout en bout entre le destinataire et la source
- Connexion réalisé avant l'échange d'information
- Libération à la fin de la connexion
  - ◆ Pas de stockage intermédiaire
  - ◆ Des ressources sont mobilisées
  - ◆ Régulation faite à la connexion

**Etablissement d'un circuit virtuel**

# Commutation (2)



## Commutation de messages.

- Pas de lien physique entre le destinataire et la source
- Chaque bloc d'informations (message) constitue une unité de transfert acheminée individuellement par le réseau
- Le message est mémorisé à chaque nœud
  - ◆ Utilisation d'une mémoire importante
  - ◆ Meilleure utilisation des lignes
  - ◆ Transfert même si le correspondant est occupé
  - ◆ Multicast possible



## Commutation de paquets.

- Les messages sont découpés en paquet de longueur maximale fixe
- Le séquençement des paquets n'est pas garanti
  - ◆ Réassemblage des paquets
  - ◆ Meilleure utilisation des lignes
  - ◆ ...



# Commutation (3)

## Comparaison circuits / paquets

Technique	Circuits	Paquets
Circuit dédié	Fixe	Dynamique
Gaspillage de BP	Oui	Non
Mémoire tampon	Non	Oui
Même chemin	Oui	Oui/Non
Congestion	Non	Oui
Facturation	Temps	Taille
Séquencement	Oui	Non

**But : *Emuler un circuit virtuel dans un réseau à commutation de paquets***

# ***Mode Non connecté***

- Les informations ( paquets) transitent dans le réseau indépendamment les uns des autres
- Les routes empruntées ne sont pas toujours les mêmes
- Le séquençement n'est pas garanti
- Un paquet peut se perdre

Le mécanisme de réception d'un paquet est assez compliqué  
( remise en ordre,...)



**Mode Datagramme**

# Mode connecté



## 3 phases distinctes

- Connexion ( discussion des paramètres,...)
- Transfert des données
- Déconnexion

	Paquets	Circuits
Mode non connecté Niveau 4	UDP au-dessus de IP	UDP au-dessus de ATM
Mode connecté Niveau 4	TCP au-dessus de IP	AAL1 (ATM) au-dessus de ATM

# *Mécanismes mis en œuvre dans un réseau (1)*

- L'adressage
  - Identification sans ambiguïté d'une machine
    - Adressage à plat ou global
    - Adressage hiérarchique (réseau, pt d'accès, identifiant)

Exemple : adresse IP, adresse ATM, n° téléphone,...
- Découpage des données
  - Fragmentation des données pour être en adéquation avec les protocoles transmettant les données

Chaque réseau à une taille maximale au niveau paquet:

**MTU : Maximum Transfert Unit**

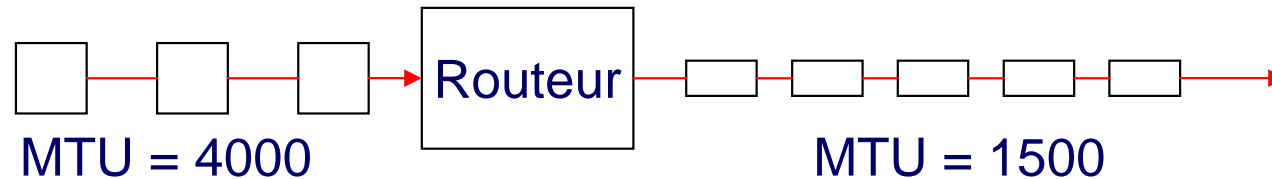
Exemple : Arpanet : 1000 octets  
Ethernet : 1500 octets  
FDDI : 4470 octets



**Travail du routeur**

# Mécanismes mis en œuvre dans un réseau (2)

Exemple découpage des données en IP (Fragmentation IP)



...	16 bits	16 bits	3 bits	13 bits	8 bits	...
	Longueur totale	Identification	Flag	Position du fragment	ttl	

Flag : bit 1 : non utilisé  
bit 2 : indique si le paquet peut être fragmenté  
bit 3 : dernier fragment (0), fragment autre (1)

Identification : n° attribué à chaque fragment afin de permettre leur réassemblage

Position fragment : position du début du fragment

Un paquet fragmenté n'est jamais réassemblé sauf par le destinataire



# ***Mécanismes mis en œuvre dans un réseau (3)***

- Le routage
  - Mise en œuvre de protocole de routage (RIP, OSPF, BGP,...)
  - Implémentation manuelle des routes
- La congestion
  - Mise en œuvre de méthode de contrôle de flux
    - exemple dans TCP, frame relay,...
- La sécurité
  - Politique de sécurité, cryptage, certificat ...

# Transmission des données

# Quantification du trafic (1)

- But :
  - Minimiser le nombre de lignes sans dégrader trop le service
  - m abonnés, n lignes  $\left\{ \begin{array}{l} \text{idéale : } m = n \\ \text{mais } n < m \text{ pour le coût} \end{array} \right.$

- Formule d'Erlang

1 erlang = 1 ligne occupée pendant 1 heure

- $V = \int_0^T n(t) dt \iff V = N * T$  avec **V : Volume du trafic écoulé**  
T : durée moyenne d'une session (en s)  
N : nb de connexions par heure
- $E = V/T \iff E = N * T / 3600$  avec **E : Intensité du trafic (en erlang)**
- Formule de erlang B : Etant donné une **intensité E** , comment choisir **m** pour que la probabilité **p** de refus d'appel soit inférieur à un nombre fixé :

$$p = (E^m / m!) / \left( \sum_{k=0}^{k=m} E^k / k ! \right)$$



# Quantification du trafic (2)

- Taux d'activité d'une ligne

$$A = t / T$$

avec A : taux d'activité

t : temps de transfert effectif

T : temps d'occupation de la ligne

- Or  $t = \text{Volume} / \text{Débit}$

$$\text{d'où } A = \text{Volume} / (\text{Débit} * T)$$

- Exemple

- Nb de session/ h : 15
- Durée moyenne d'une session : 3 mn
- Longueur moyen d'un message : 640 caractères
- Nb de messages transmis par session : 1
- Débit : 2400 b/s

$$E = 0,75$$

$$A = 1 \%$$

# Multiplexage (1)

**But** : permettre de faire passer plusieurs voies de communications sur un même support

- Multiplexage spatial ou fréquentiel

**FDMA : Frequency Division Multiple Access**

- Bande passante divisée en canaux
- Entre chaque canal, un espace de fréquence (bande de garde)

pour éviter l'intermodulation

- Chaque canal est affecté à une voie basse vitesse

→ { De moins en moins utilisé actuellement  
Non utilisé en informatique car analogique

- Multiplexage WDMA

**Wavelength Division Multiple Access**

- utilisation de longueurs d'ondes différentes dans les fibres optiques.

# ***Multiplexage (2)***

- Multiplexage temporel

## **TDMA : Time Division Multiple Access**

- utilisable par les réseaux numériques
- On divise le temps en intervalle de temps, chaque intervalle étant dédié à une communication.
- Nécessite de la mémoire

- Multiplexage de code

## **CDMA : Code Division Multiple Access**

- utilisé par les réseaux téléphoniques (UMTS)
- basé sur l'utilisation de vecteurs orthogonaux  
( multiplication = 0 → exemple: (1,0,1,1) et (1,-1,-1,0))
- permet de coder plusieurs signaux sur la même fréquence

# Multiplexage (3)

- Fonctionnement CDMA (sur un cas simple)

vecteur  $v = (1, -1)$  et  $x = (1, 1)$

$$v * x = 0$$

On associe à  $v$  le bit 1, et à  $-v$  le bit 0 (idem pour  $x$ )

On veut transmettre  $(1, 0, 1, 1)$  pour  $v \rightarrow ((1, -1), (-1, 1), (1, -1), (1, -1))$

On veut transmettre  $(0, 0, 1, 1)$  pour  $x \rightarrow ((-1, -1), (-1, -1), (1, 1), (1, 1))$

En transmettant ensemble, addition des vecteurs

➡ résultat transmis :  $(0, -2, -2, 0, 2, 0, 2, 0)$

On prend par couple (vecteurs sont constitués par un couple):

Pour déchiffrer le message de  $v$ ,

pour déchiffrer le message de  $x$

$$(0, -2) * (1, -1) = 2 > 0 \rightarrow 1$$

$$(-2, 0) * (1, -1) = -2 < 0 \rightarrow 0$$

$$(2, 0) * (1, -1) = 2 > 0 \rightarrow 1$$

$$(2, 0) * (1, -1) = 2 > 0 \rightarrow 1$$

$$(0, -2) * (1, 1) = -2 < 0 \rightarrow 0$$

$$(-2, 0) * (1, 1) = -2 < 0 \rightarrow 0$$

$$(2, 0) * (1, 1) = 2 > 0 \rightarrow 1$$

$$(2, 0) * (1, 1) = 2 > 0 \rightarrow 1$$

# Optimisation

- **But** : diminuer le trafic sur le réseau  
coût d'une liaison → Utilisation de la bande passante

## **Plusieurs solutions :**

- Compression des données
  1. Avec perte  
exemple : mpeg1, mpeg2, mpeg4,...
  2. Sans perte  
exemple : codage de Huffman
- Optimisation dans les échanges du protocole de communication  
un acquittement de temps en temps, etc...
- Diminution de la taille des entêtes

# Protocoles Hauts débits

( Autre que Ethernet ..... Cf. premier semestre)

# ***DQDB : IEEE 802.6 (1)***

**DQDB** : Distributed Queue Dual Bus

→ objectif : être un réseau MAN

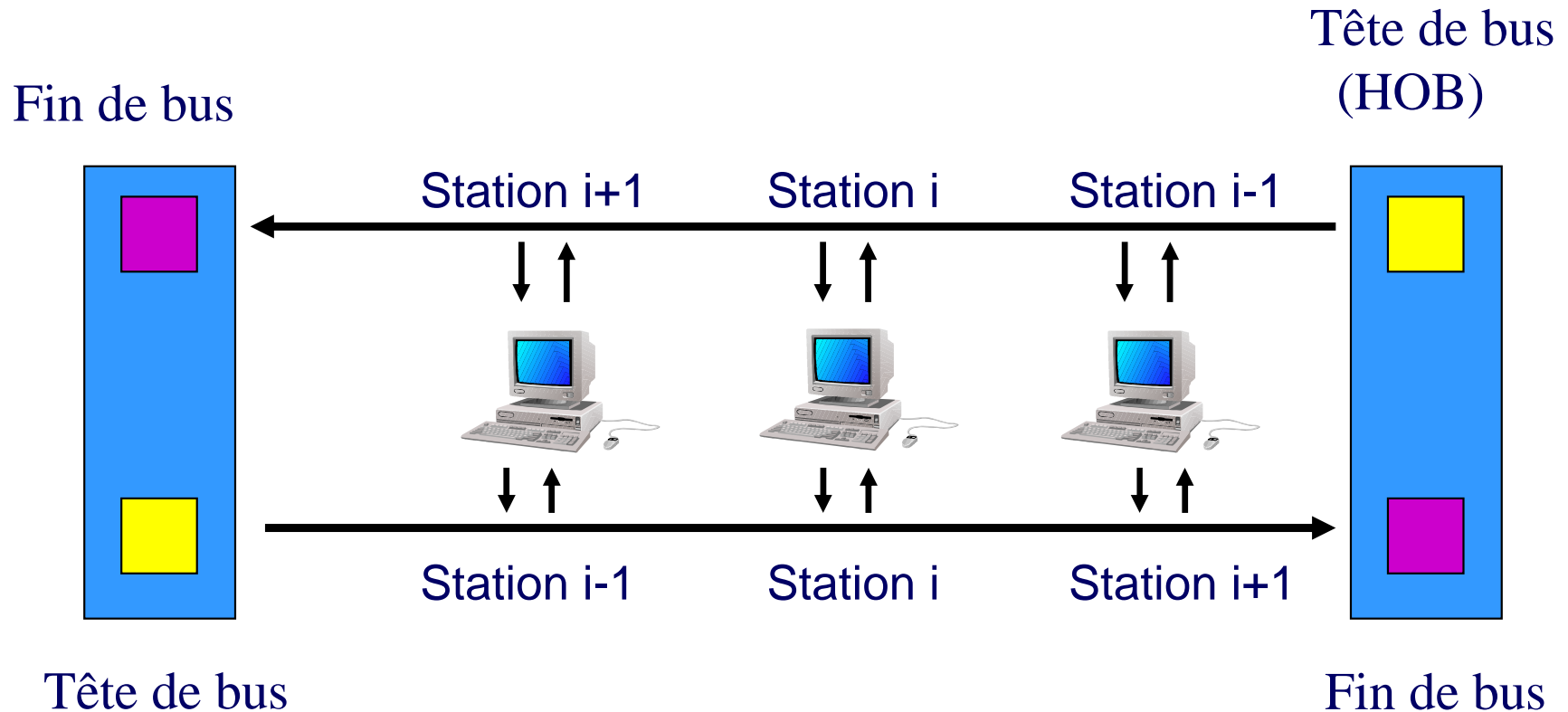
→ débit : 160 MB/s sur fibre optique

## **Caractéristiques :**

- Technologie en double bus et de sens inverse
- Chaque station est raccordée en parallèle sur chacun des deux bus
- A chaque extrémité d'un bus, une "tête de bus" qui émet régulièrement des trames/cellules vides
- Chaque trame est composée de 53 octets.
- Possibilité d'avoir des communications :
  - en mode connecté/ non connecté
  - en mode asynchrone/ isochrone

# ***DQDB : IEEE 802.6 (2)***

HOB : Head Of Bus



Chaque cellule comprend 53 octets, 5 octets pour l'entête et 48 octets pour les données.



# ***DQDB : Accès au médium (1)***

**But** : Réserver une trame par l'intermédiaire du bus B pour émettre sur le bus A.

Chaque station a 4 compteurs :

- 2 Request Counter (RC)
- 2 Countdown Counter (CD)

} Un pour chaque bus

RC : compte le nombre de requêtes venant des autres stations  
➡ nb de cellules vides à laisser passer

2 bits importants dans l'entête :

- Bit **Busy** : cellule vide ou pleine
- Bit **Rsvd** : cellule déjà réservé ou non

# ***DQDB : Accès au médium (2)***

Exemple : Station x veut émettre sur le bus A

- Sur le bus B,
  - première trame dont rsvd= 0, alors rsvd <- 1
  - recopie RC dans CD
  - remise à zéro de RC
- Sur le bus A,
  - Si CD=0, et trame vide, alors on émet  
sinon si trame vide,  $CD = CD - 1$

## ***Autre station :***

A chaque requête sur le bus B,  $RC = RC + 1$

A chaque cellule vide sur le bus A,  $RC = RC - 1$

# ***DQDB : Frames***

- Les cellules sont architecturées en frame qui sont émises toutes les 125  $\mu$ s
- Chaque frame contient en plus des cellules DQDB, un entête et une fin de frame utilisées pour la synchronisation des stations.

