

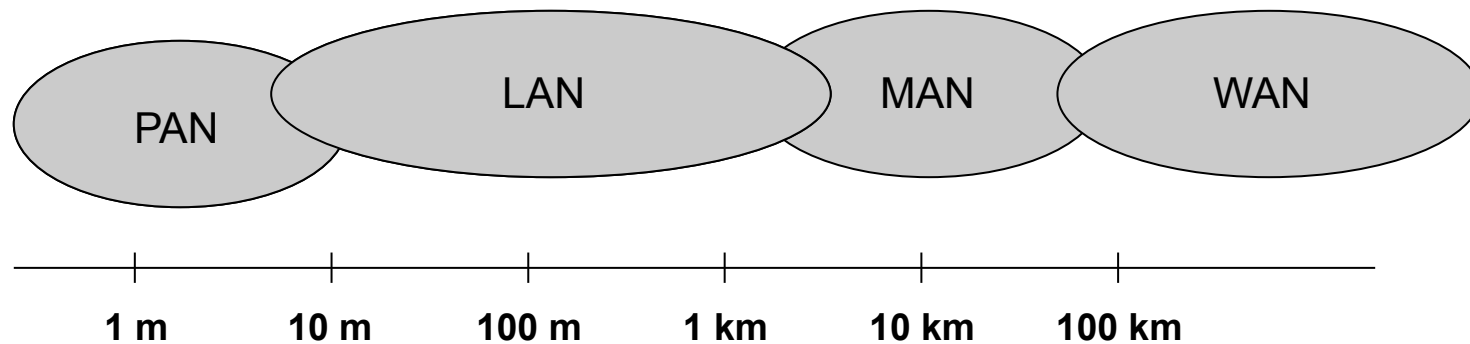
Partie 4

Réseaux grande distance

Réseaux grande distance - Plan

- ▣ **Réseaux grande distance**
- ▣ La commutation
- ▣ Les fonctions de réseau
- ▣ Acheminement par voie logique
- ▣ Acheminement par datagramme

Classification par taille



PAN - Personal Area Network

LAN – Local Area Network

MAN – Metropolitan Area Network

WAN – Wide Area Network

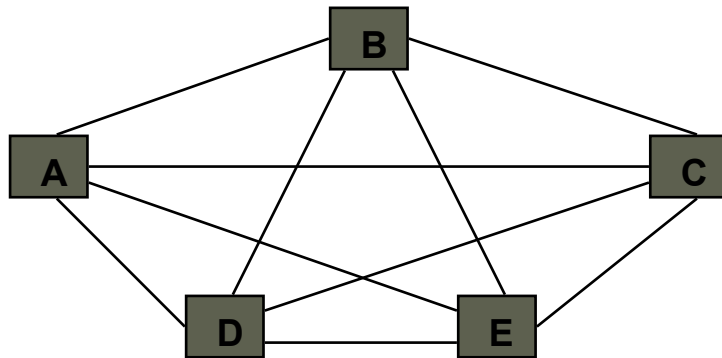
Caractéristiques des WAN

- réseaux étendus sur plusieurs centaines voire milliers de km (un pays, un continent, ...)
- réseaux publics ou privés
- composés de commutateurs, de routeurs et de liaisons entre eux
- des milliers d'ordinateurs y sont connectés

Problématique



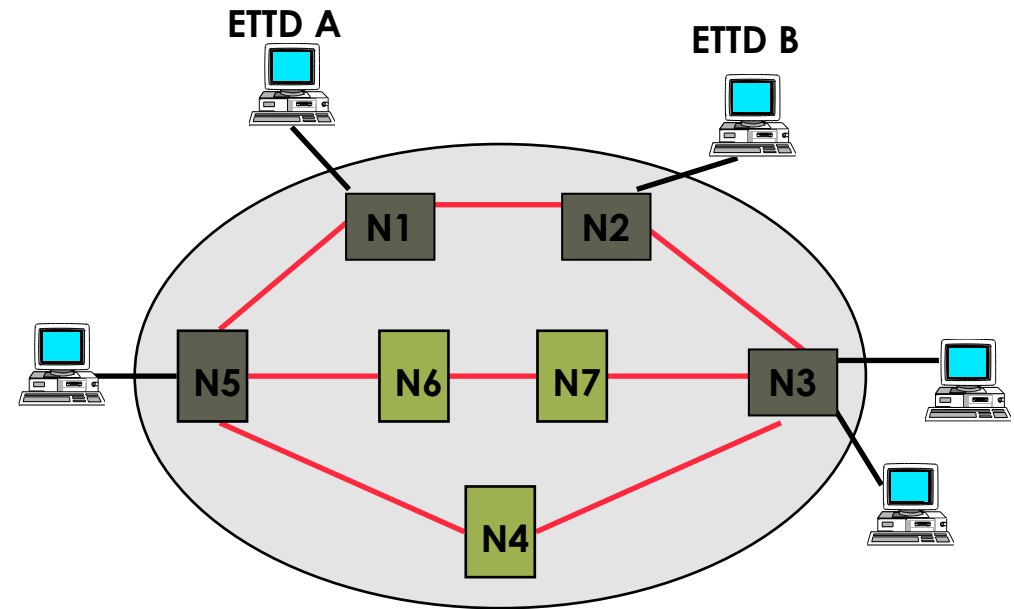
- on sait faire avec une liaison de données
- mais que se passe-t-il lorsqu'on a N ETTD ?



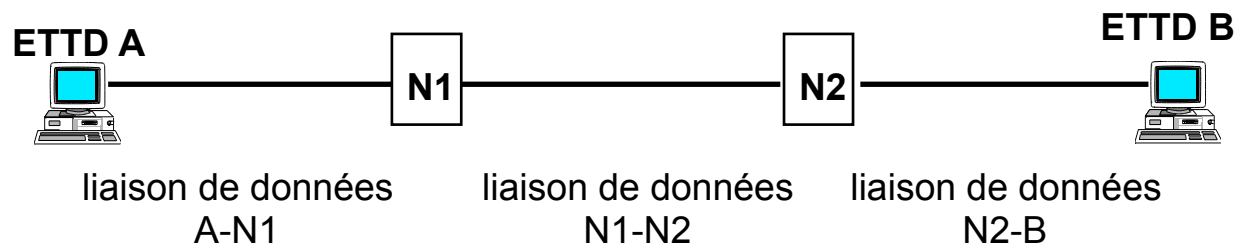
$$N \text{ ETTD} \Rightarrow \frac{N(N-1)}{2} \text{ liaisons !}$$

Un réseau...

- Des nœuds et des liaisons
 - topologie**
 - nœuds d'accès** et **nœuds internes**
- Chaque station est reliée à un nœud d'accès



↪ Une brique de plus !



Services de réseau

- assurer :
 - l'indépendance par rapport aux supports de transmission sous-jacents
 - le transfert de bout-en-bout (*entry-to-exit*) des données
 - la transparence des informations transférées
 - le choix d'une QoS
 - le service d'adressage aux points d'accès du service de réseau
- le transfert doit-il être fiable ?

2 réponses possibles...

Service en mode connecté

- service en mode **circuit virtuel**
- transfert fiable (approche «opérateur public»)
- communication en 3 phases
 - établissement de la connexion
 - transfert de données
 - libération de la connexion

Service en mode non connecté

- service en mode **datagramme**
- fiabilité non assurée **best effort** (approche Internet)
- communication
 - par échange de datagrammes indépendants
 - sans notion de connexion

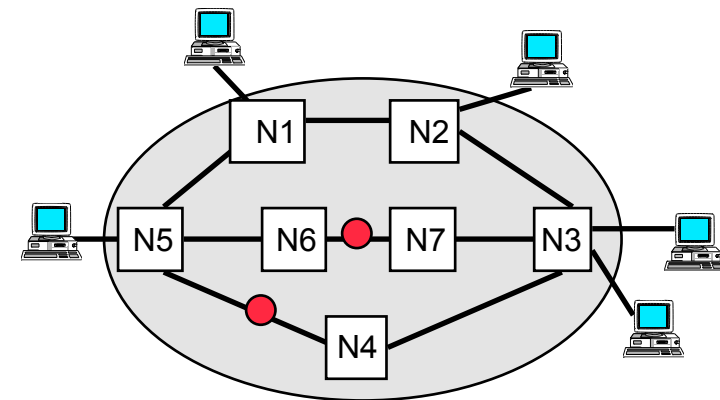
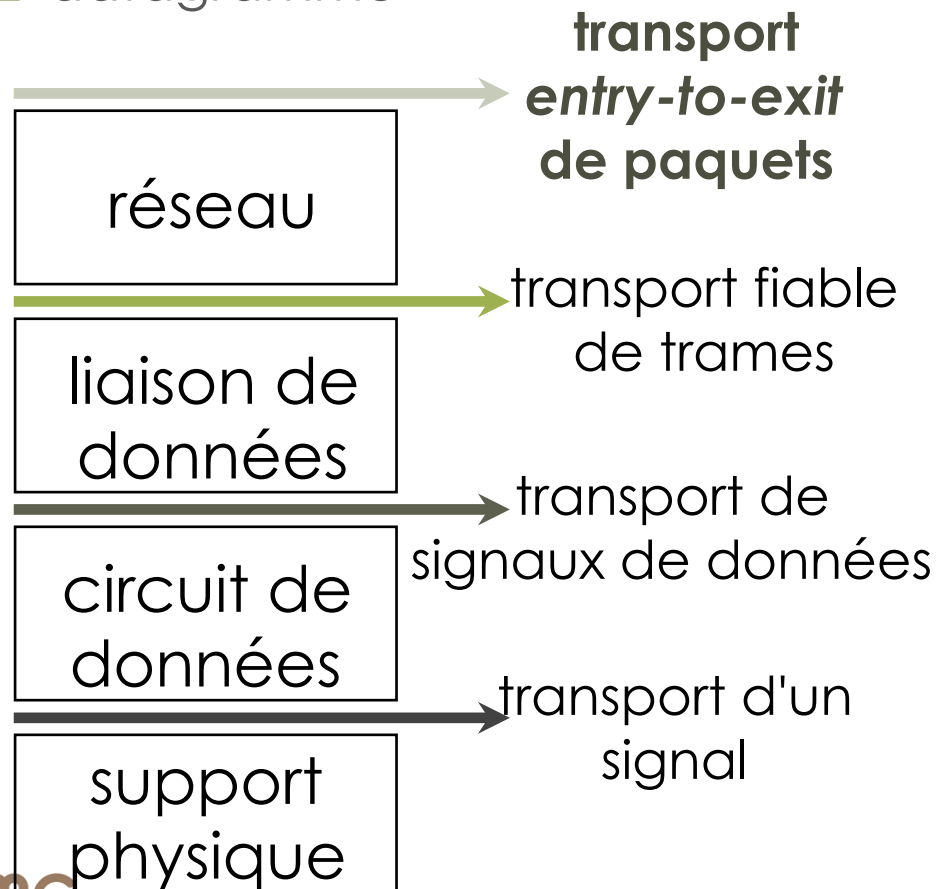
Protocole de réseau

■ 2 types de service

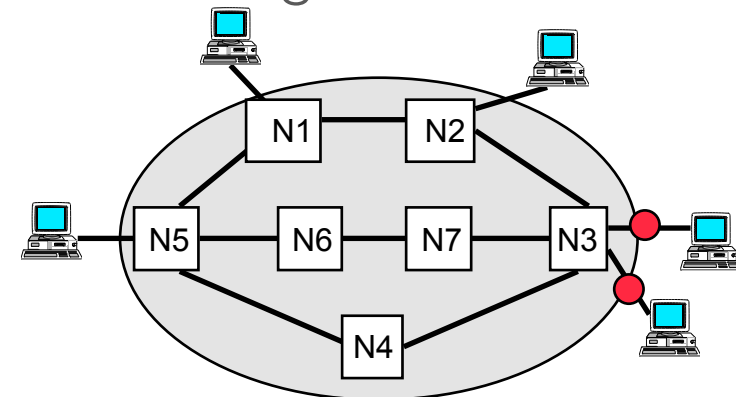
- circuit virtuel
- datagramme

■ 2 points de vue

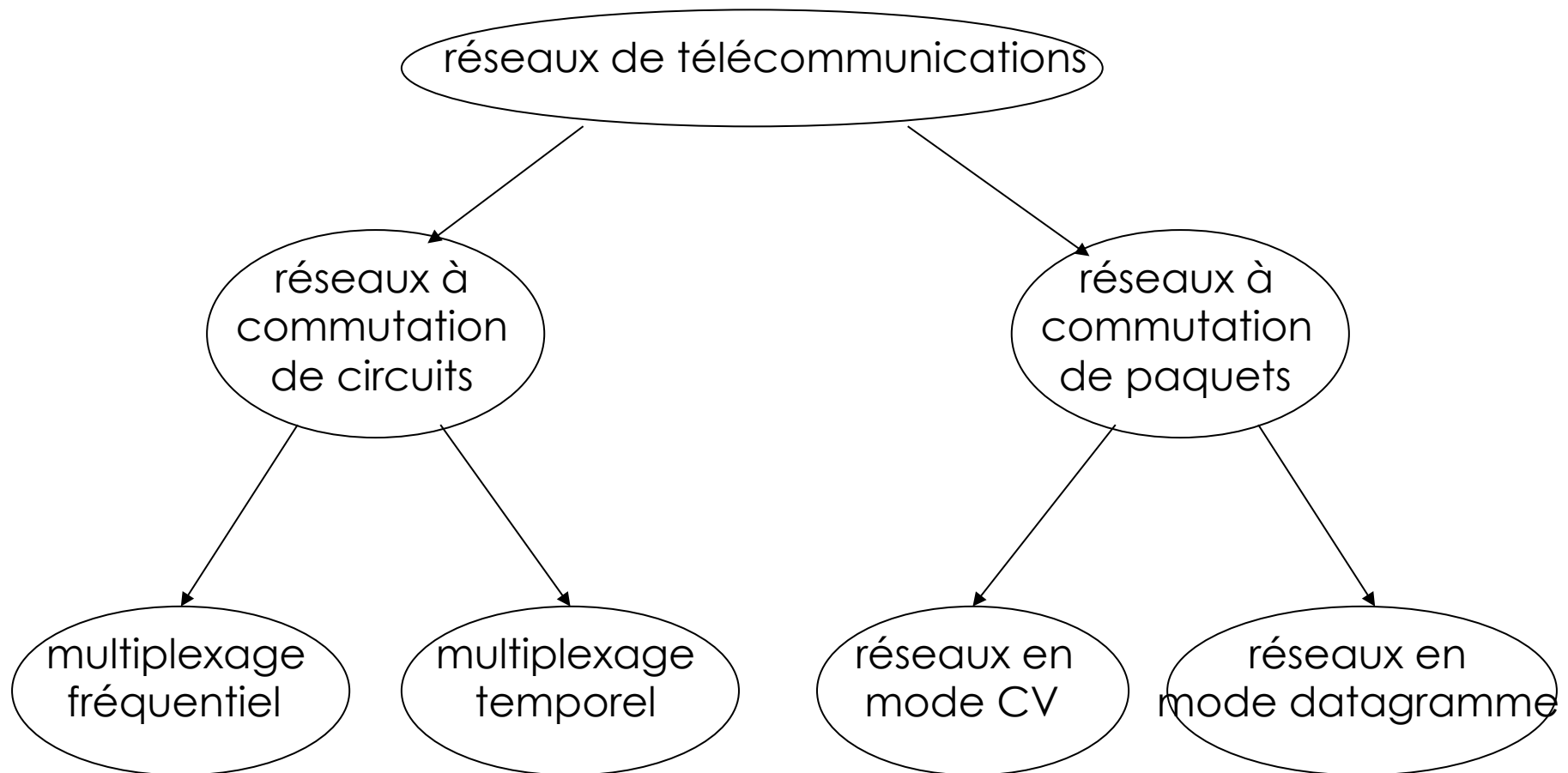
- entre deux nœuds du réseau



- entre l'utilisateur et le réseau



Taxonomie

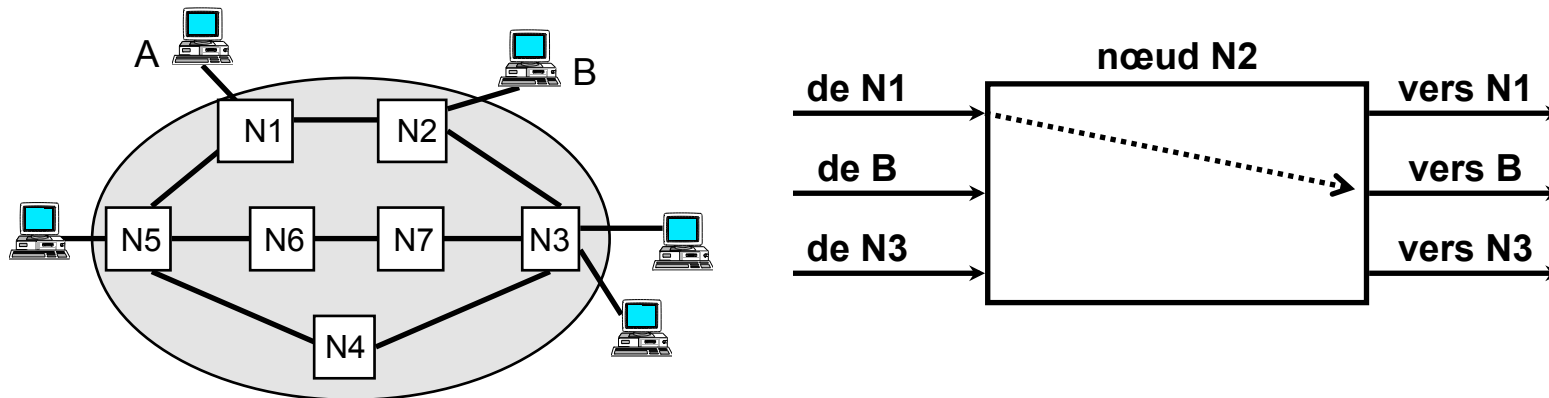


Réseaux grande distance - Plan

- Réseaux grande distance
- **La commutation**
 - définition
 - les types de commutation
- Les fonctions de réseau
- Acheminement par voie logique
- Acheminement par datagramme

La commutation

- **aiguillage** d'une communication provenant d'un canal en entrée vers un canal de sortie



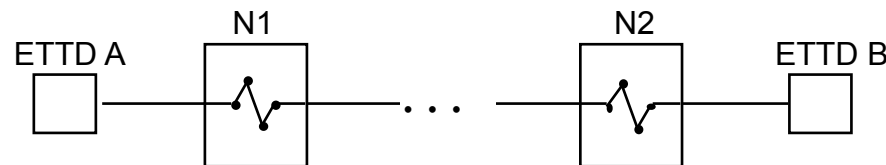
- différents types de commutation

- Circuits
- Messages
- Paquets
- Cellules

Commutation de circuits

■ Principe

- Les commutateurs établissent un itinéraire physique permanent pour chaque canal de communication
- Cet itinéraire est un circuit qui n'appartient qu'aux deux entités qui communiquent :
 - le circuit doit être établi avant que des informations ne transitent
 - le circuit dure jusqu'à ce que l'une des entités décide d'interrompre la communication
 - les ressources de communication sont allouées pour toute la durée de vie du circuit



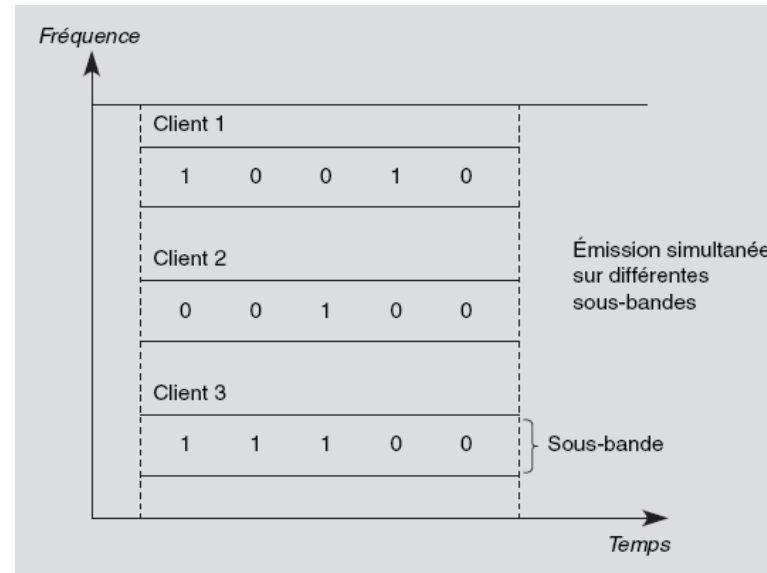
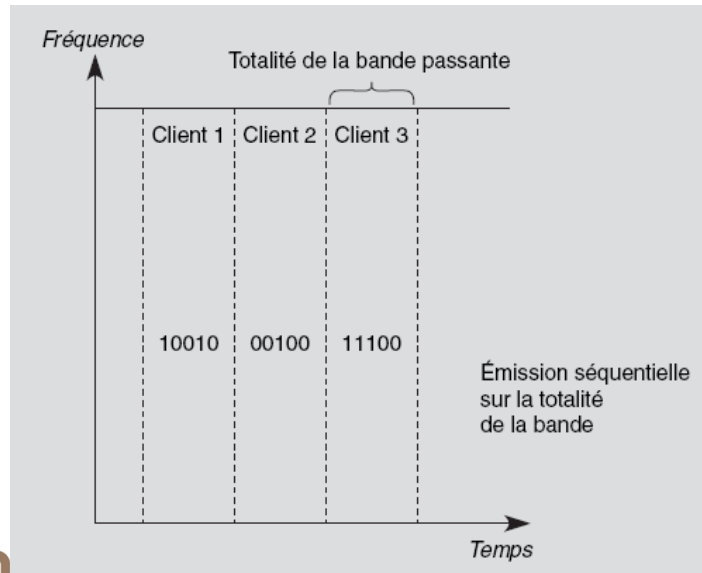
Commutation de circuits

■ Multiplexage

- Partager le support physique
- Avoir les voies haut débit

■ Deux méthodes

- Multiplexage temporel
- Multiplexage fréquentiel



Commutation de circuits

■ Avantages

- 😊 délai de transfert constant
- 😊 pas de risque de congestion du réseau

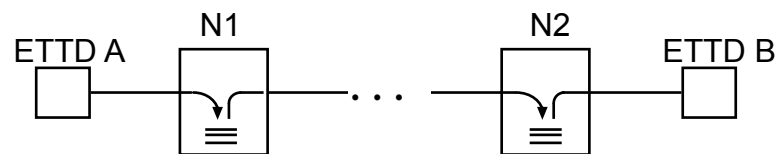
■ Inconvénients

- 😞 mauvaise utilisation des ressources
- 😞 risque de rejet à l'établissement
- 😞 délai d'établissement

Commutation de messages

■ Principe

- un message est une information formant logiquement un tout et pour la source et pour le destinataire
- chaque message est envoyé indépendamment des autres
- fonctionnement de type Store-and-Forward



Commutation de messages

■ Avantages

- 😊 les ressources ne sont utilisées que lorsque nécessaire
- 😊 pas de délai d'établissement

■ Inconvénients

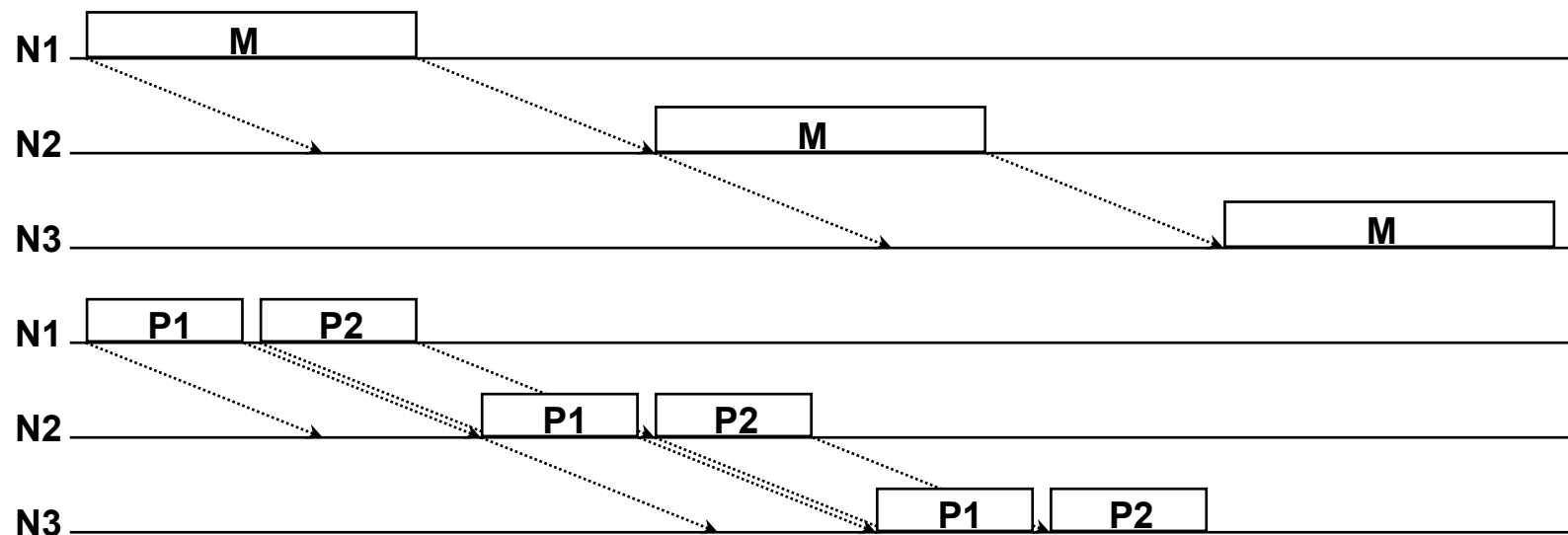
- 😞 ressources de stockage importantes
- 😞 temps de transfert importants et variables
- 😞 risques de congestion
- 😞 taux d'erreurs message importants

Commutation de paquets

■ Principe

- idem commutation de messages
- découpage des messages en paquets de taille limitée

■ commutation de paquets vs. messages



Commutation de paquets

■ Avantages

- diminution du temps de transfert

■ Inconvénients

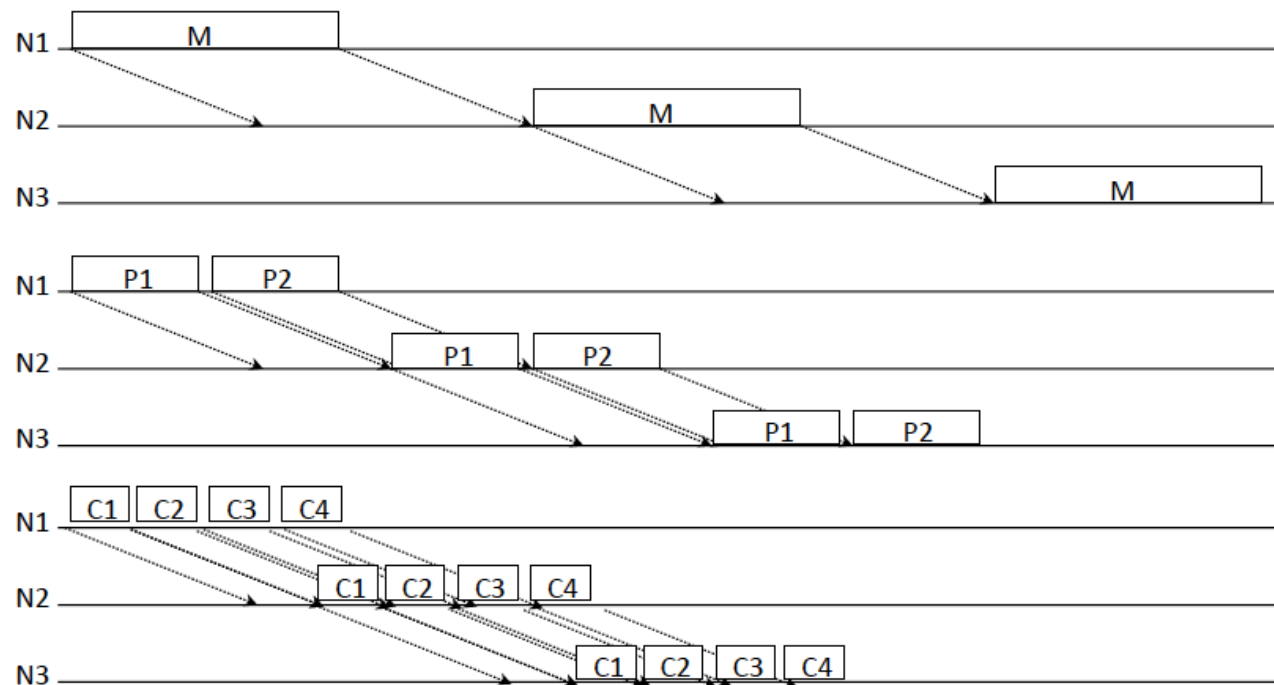
- déséquencements possibles
- temps de transfert variables
- risques de congestion

Commutation de cellules

■ Principe

- idem commutation de paquets en mode connecté
- cellule = paquet de taille fixe et courte (53 octets)

■ Commutation de cellules vs. commutation de paquets



Commutation de cellules

petite taille

- 😊 réduction du temps de constitution des paquets
- 😊 réduction du délai d'acheminement
- 😊 réduction du nombre de pertes (dus à des dépassements de files d'attente)
- 😊 réduction de la taille des tampons des nœuds
- 😊 meilleur entrelacement des messages : puisque les grands flux de données sont découpés en petites cellules, le trafic isochrone peut s'intercaler sans subir de retard significatif

- 😊 gigue faible

taille fixe

- 😊 augmentation de la capacité des nœuds (traitement // dans les nœuds)
- 😊 meilleure performance (utilisation de technologies à très haute intégration *hardware*)
- 😊 gestion mémoire des commutateurs plus simple
- 😞 mauvaise utilisation de la bande passante (le cadrage des cellules nécessite des octets de bourrage)

Réseaux grande distance - Plan

- ▣ Réseaux grande distance
- ▣ La commutation
- ▣ **Les fonctions de réseau**
 - ▣ adressage
 - ▣ routage
 - ▣ contrôle de congestion
- ▣ Acheminement par voie logique
- ▣ Acheminement par datagramme

L'adressage

■ Problématique

- lorsqu'un système de transmission est utilisé par plus de 2 équipements, pour délivrer une unité de données à son destinataire, il faut connaître l'adresse de ce dernier
- de manière unique et non ambiguë

■ Techniques d'adressage

- plat / absolu : adresse identique partout
ex : Ethernet
- hiérarchique : facilite le routage
ex : téléphone, @ postale, IP

■ Types d'adresses

- unicast
- broadcast
- multicast

Le routage

■ Problématique

- trouver la **meilleure route** pour aller d'une source vers une destination
- la meilleure → **critère de coût**
 - distance, nombre de nœuds traversés
 - temps de réponse
 - débit
 - fiabilité
 - coût financier

Le routage

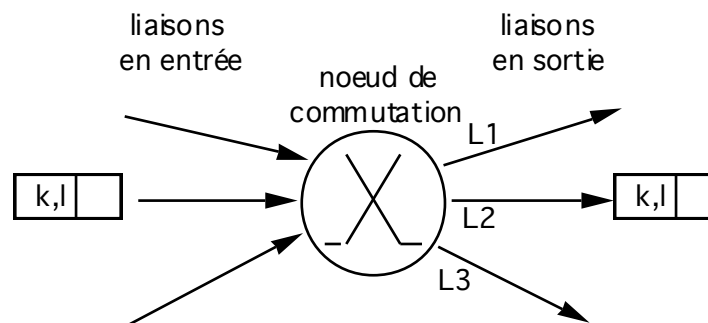
- routage : terme général, utilisé pour 2 fonctions
 - acheminement (**forwarding**)
 - trouver un chemin au paquet vers sa destination par consultation de tables
 - fonction simple, se déroulant localement à un nœud
 - adaptation des chemins (**routing**)
 - construire et mettre à jour les tables servant à l'acheminement
 - fonction complexe, mettant en œuvre des algorithmes distribués qui s'appuient sur des protocoles de routage

La fonction d'acheminement

■ Principe

- chaque nœud dispose d'une **table de routage** qui donne pour chaque destination le port (la ligne) de sortie sur lequel le nœud doit commuter le paquet
- par sauts successifs, de nœud en nœud, le paquet parvient au destinataire

■ exemple



destination	liaison de sortie
a	L1
l	L2
N	L3

- un paquet d'origine k et à destination de l traverse un nœud
- le nœud consulte sa table de routage qui donne L2 comme liaison de sortie

La fonction d'acheminement

- notion d'**étiquette**

- information de contrôle contenue dans l'en-tête du paquet, utilisée pour acheminer le paquet

- 2 modes d'acheminement

- acheminement par voie logique
 - acheminement par datagramme

Prise de décision du routage

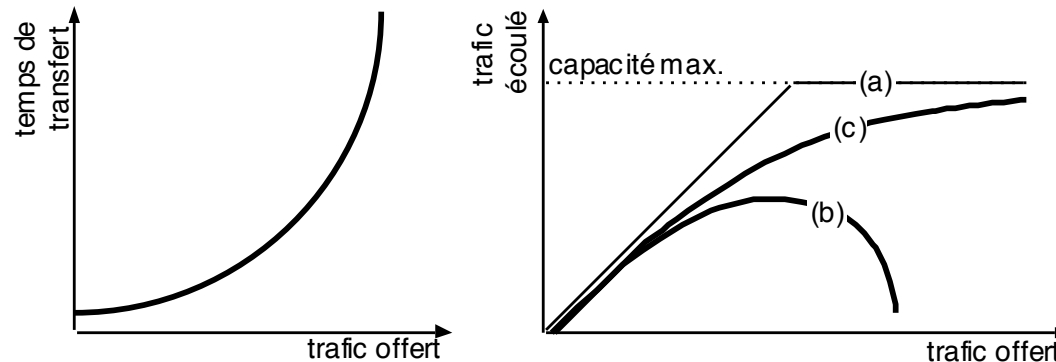
- Le moment de prise de décision du routage (consultation de la table de routage) dépend du mode d'acheminement
 - acheminement par voie logique
 - décision de routage prise une seule fois lors du passage du paquet d'établissement
 - tous les autres paquets de la connexion suivent la même route, après consultation de la table de translation
 - par datagramme
 - une décision de routage est prise pour chaque paquet
 - la route peut être identique ou non pour deux paquets ayant le même destinataire

Le contrôle de congestion

■ Problématique

- ressources limitées dans le réseau
 - capacité de transmission des liaisons
 - capacité de traitement des nœuds
 - capacité de stockage (en émission et en réception) des nœuds
- lorsque le trafic soumis augmente, le réseau subit des phénomènes de congestion → effondrement des performances
 - trafic utile écoulé
 - temps de transfert moyen
 - pertes de paquets

Le contrôle de congestion



- a) réseau idéal : capable d'acheminer un trafic utile directement proportionnel à la charge soumise
 - ✓ ce jusqu'à ce que sa capacité maximum de transport soit atteinte
 - ✓ à ce point-là, le réseau devrait être capable de fonctionner à sa capacité maximum, quelle que soit la charge soumise
- b) en pratique, le fonctionnement du réseau s'écarte de l'idéal, à cause d'une allocation de ressources inefficace en cas de surcharge (retransmissions en cascade)
- c) pouvoir éviter et guérir la congestion !!!

Le contrôle de congestion

- contrôle en boucle ouverte
 - résoudre le problème par une bonne conception
 - utilisation de politiques pour déterminer s'il faut accepter de nouveaux paquets ou les supprimer
 - prendre les décisions quel que soit l'état du réseau.
- contrôle en boucle fermée
 - basé sur le concept de mécanisme de feedback.
 - 3 phases
 - surveiller les congestions : % de paquets détruits, longueur moyenne des files d'attente, nombre de timers déclenchés, déviation standard du délai du paquet
 - informer : les routeurs envoient un paquet à la source, ou un paquet de données informe les routeurs, ou les routeurs échangent de la signalisation
 - corriger le problème : augmentation des ressources ou réduction de la charge

Le contrôle de congestion : solutions

- surdimensionnement des équipements afin de se placer dans une zone de fonctionnement éloignée de la congestion
 - problème d'efficacité et de coût
- contrôle d'admission
- régulation de trafic et lissage
- contrôle de flux par fenêtre
 - ne s'applique pas au mode non connecté
 - # total de paquets dans le réseau : fonction du # instantané de paires actives et du type de communication (uni ou bidirectionnelle)
 - débit maximum sur la connexion limité par la vitesse de progression de la fenêtre
 - risque de congestions locales si le réseau comporte des points de passage obligatoires ou fort fréquentés,
 - le réseau est utilisé très en dessous de sa capacité maximale ; on a une utilisation peu efficace de ses ressources

Le contrôle de congestion : solutions

- préallocation des ressources
 - le paquet d'appel se charge de réserver dans chaque nœud qu'il traverse les ressources qui seront allouées à cette communication
 - approche "commutation de circuits"
 - utilisation peu efficace des ressources, d'autant plus si la charge soumise est faible
- régulation isarithmique
 - limiter le nombre de paquets en transit dans le réseau par un système de jetons
 - mise en œuvre difficile : répartition, circulation et intégrité des jetons
- paquets d'engorgement
 - le nœud mesure l'occupation de ses buffers ; lorsqu'elle dépasse un seuil, il envoie des notifications de congestion

Réseaux grande distance - Plan

- ▣ Réseaux grande distance
- ▣ La commutation
- ▣ Le multiplexage
- ▣ Les fonctions de réseau
- ▣ **Acheminement par voie logique**
 - ▣ Principes
 - ▣ L'exemple de X.25
 - ▣ L'exemple du Frame Relay
 - ▣ L'exemple d'ATM
- ▣ Acheminement par datagramme

Acheminement par voie logique

■ Principe

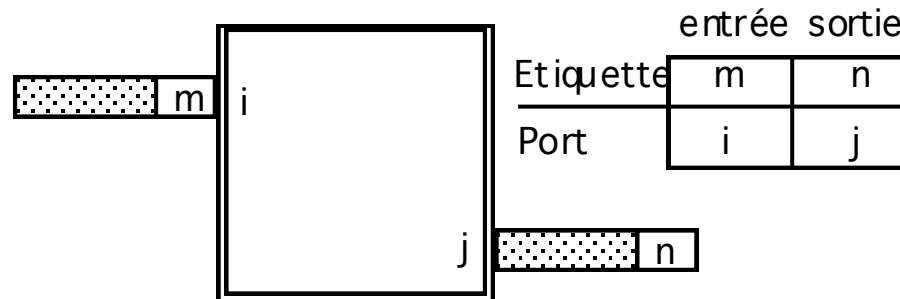
- étiquette = N° VL

- signification locale au multiplex (i.e. à un tronçon de connexion)

- permet de distinguer plusieurs communications empruntant ce multiplex

↪ translation d'étiquette par le nœud

- les informations nécessaires à cette translation sont stockées dans une table du nœud appelée *table de translation* (*lookup*)



Acheminement par voie logique

- préalablement à tout transfert de données, un itinéraire doit être marqué dans les nœuds via une correspondance temporaire entre une voie logique entrante sur un multiplex entrant et une voie logique sortante sur un multiplex sortant

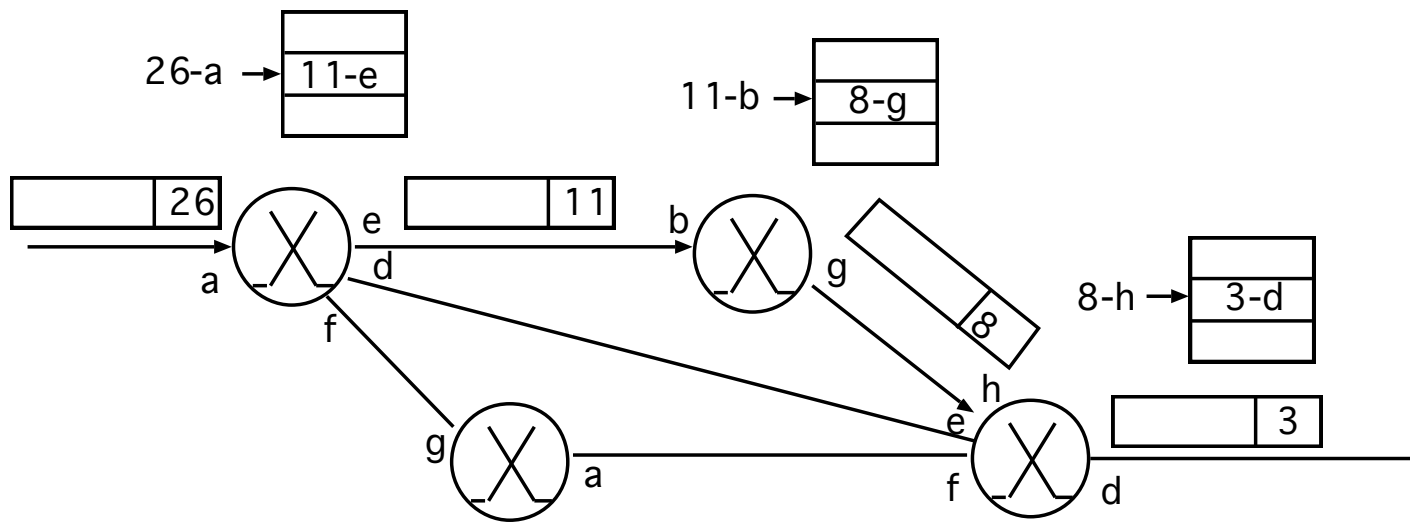
↳ mode connecté

- tables de translation mises à jour lors de l'établissement et de la libération des connexions
- les paquets d'une même connexion suivent le même chemin physique → *circuit virtuel*

Acheminement par voie logique

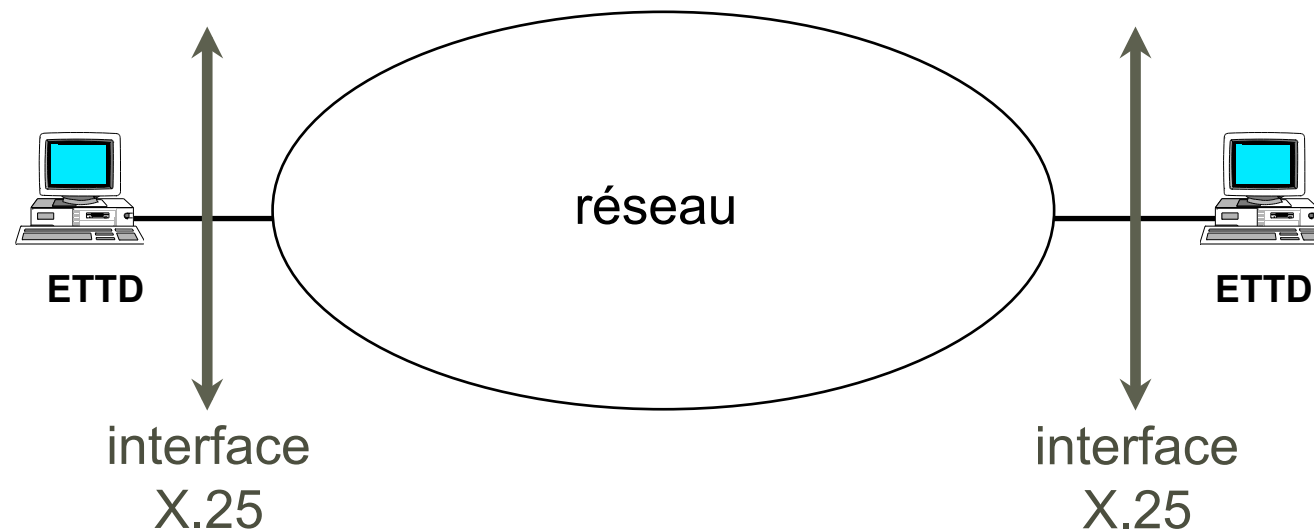
■ exemple

- la succession des numéros logiques peut être vue comme un circuit : c'est le *circuit virtuel*



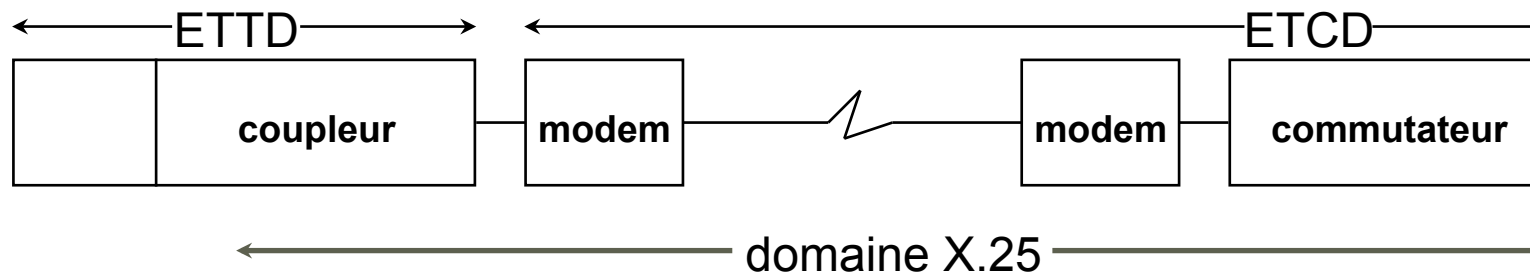
Un exemple d'interface d'accès

- X.25 : interface d'accès à un réseau à commutation de paquets
- adopté par le CCITT en 1976
- offre un service de réseau en mode connecté
- supporté par Transpac (France), EPSS (Grande-Bretagne), Datapac (Canada), Telenet (USA), ...

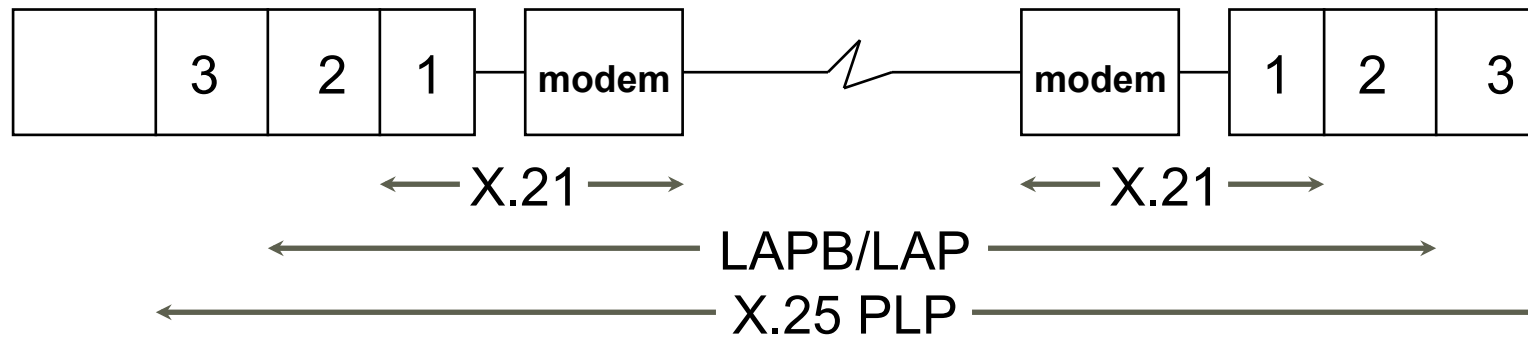


X.25 : une interface

- X.25 s'applique entre un ETTD et un ETCD



- 3 niveaux de protocole



X.25 : le niveau PLP

- 2 types de circuits virtuels

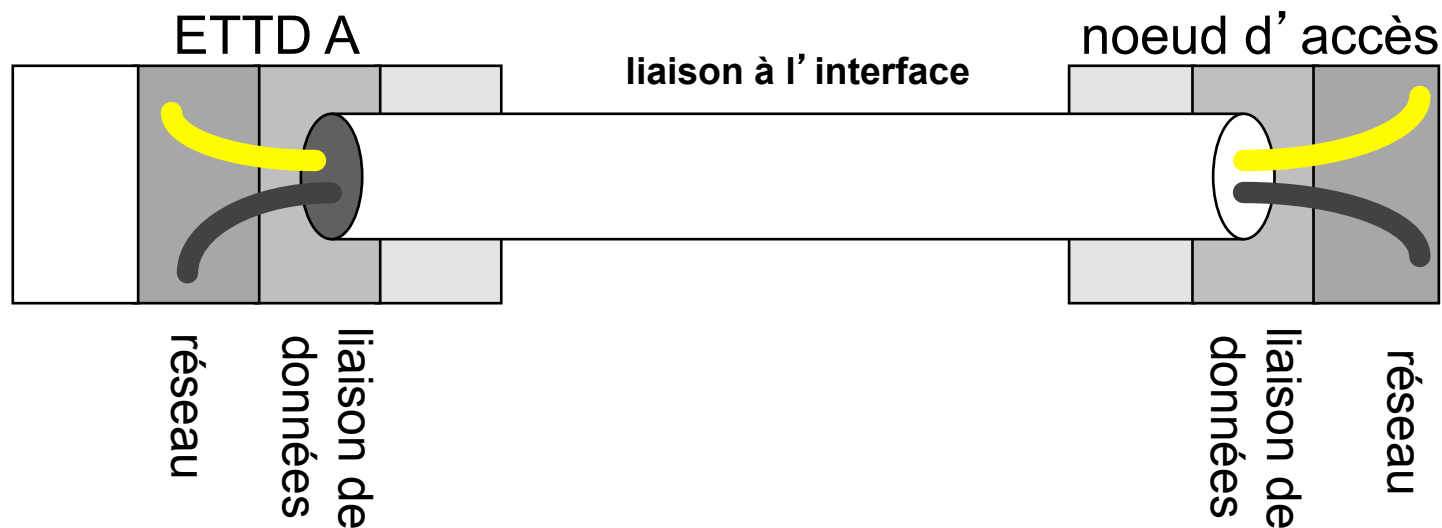
- CVP : permanent, il relie en permanence 2 abonnés
- CVC : commuté, il permet à un abonné d'atteindre tout autre abonné

- **voie logique**

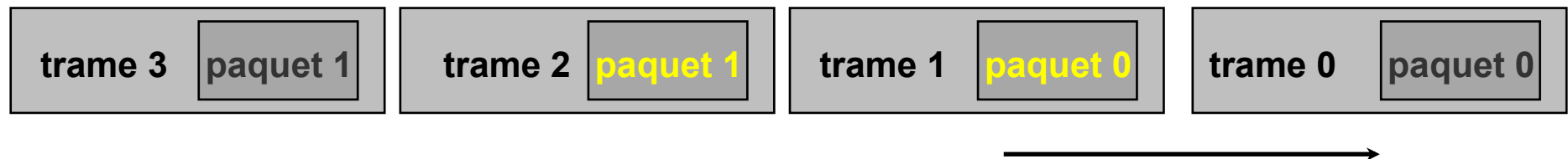
- notion permettant la coexistence de tronçons de plusieurs CV sur une même liaison
- moyen de transmission bidirectionnelle simultanée sur une liaison de données
- identifiée par un N° de GVL (£ 15) et un N° de VL (£ 255)
 - signification purement locale à l'interface
 - valeurs attribuées à l'abonnement (CVP), à l'établissement (CVC)

X.25 : voie logique et liaison

- multiplexage de voies logiques sur la liaison



- entrelacement de paquets de CV différents dans le flot de trames

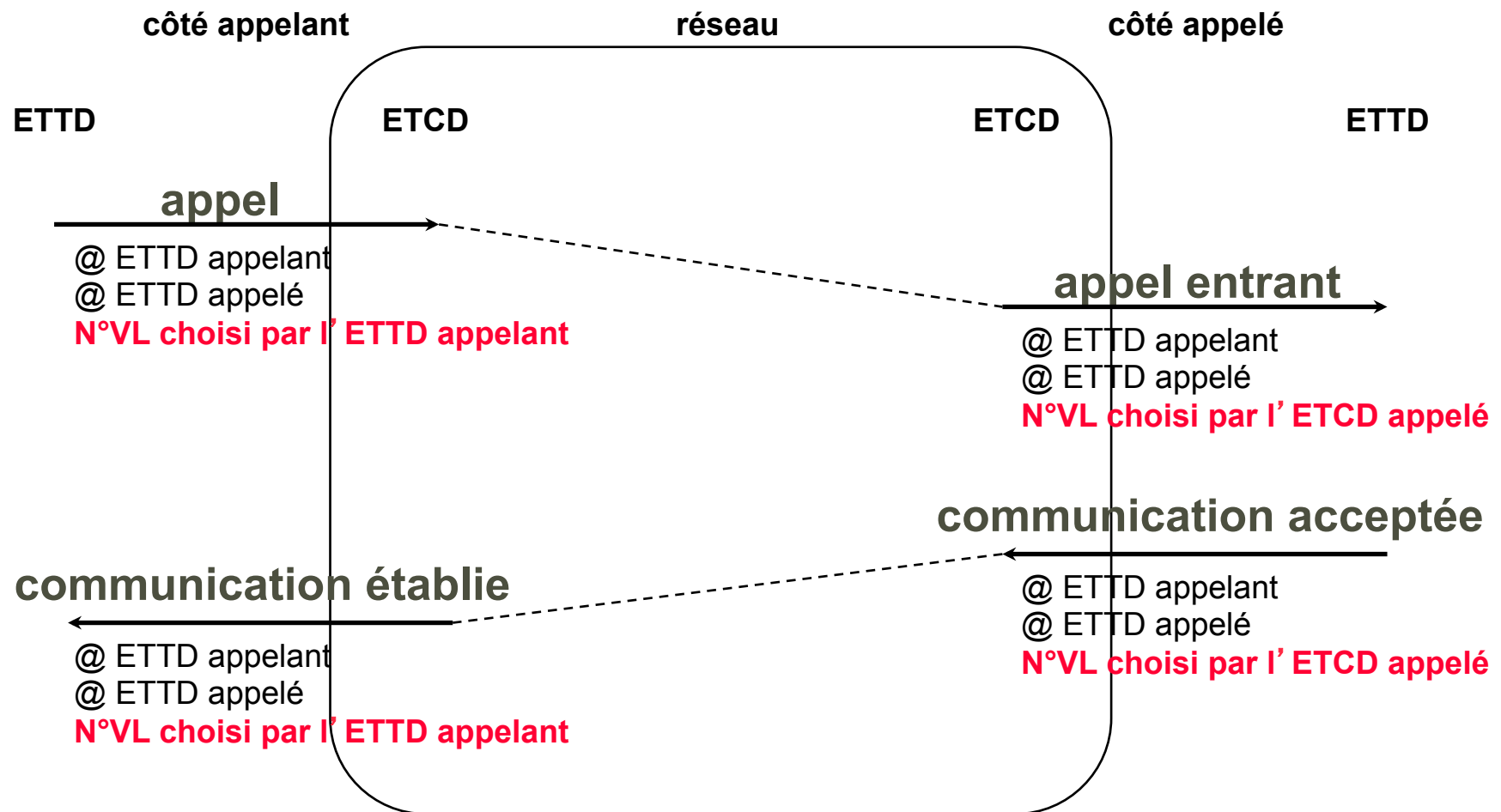


X.25 : les paquets

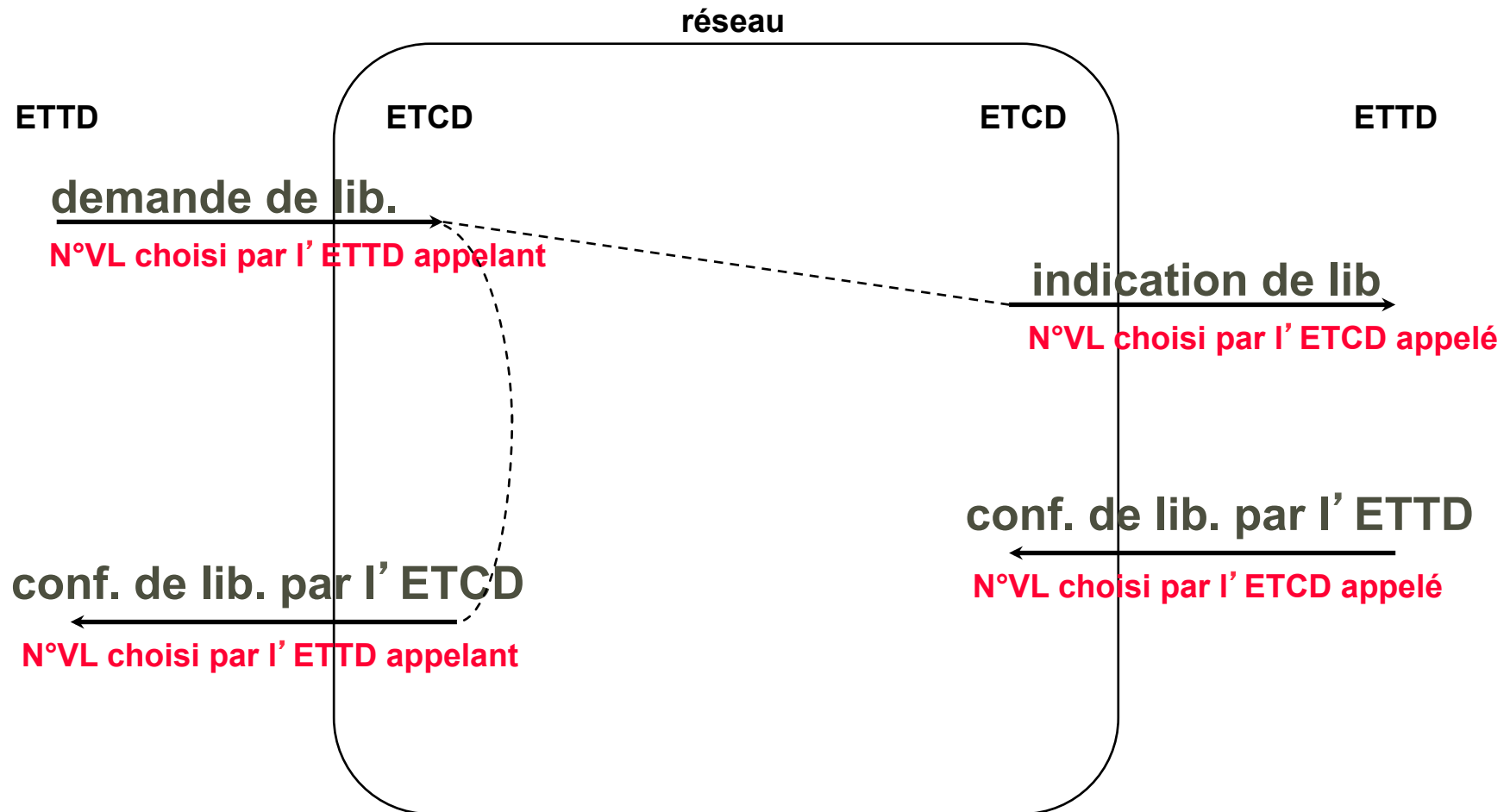
- structure de base avec un en-tête d'au moins 3 octets :
 - identification générale de format (dont le modulo utilisé)
 - identification de VL
 - identification de type de paquet

- des paquets pour :
 - établir un CV
 - transférer des données (avec contrôle de flux et interruptions)
 - libérer un CV
 - réinitialiser un CV

X.25 : établissement d'un CVC



X.25 : libération d'un CVC



X.25 : le transfert de données

- les (principaux) paquets utilisés

Q	D	0	1	N°GVL	
N°VL					
P(R)		M	P(S)		0
données utilisateur					

données

0	0	0	1	N°GVL			
N°VL							
0	0	1	0	0	0	1	1
données utilisateur (de 1 à 32 octets)							

interruption

0	0	0	1	N°GVL	
N°VL					
P(R)	0	0	0	0	1

RR

0	0	0	1	N°GVL	
N°VL					
P(R)	0	0	1	0	1

RNR

0	0	0	1	N°GVL			
N°VL							
0	0	1	0	0	1	1	1

confirmation
d' interruption

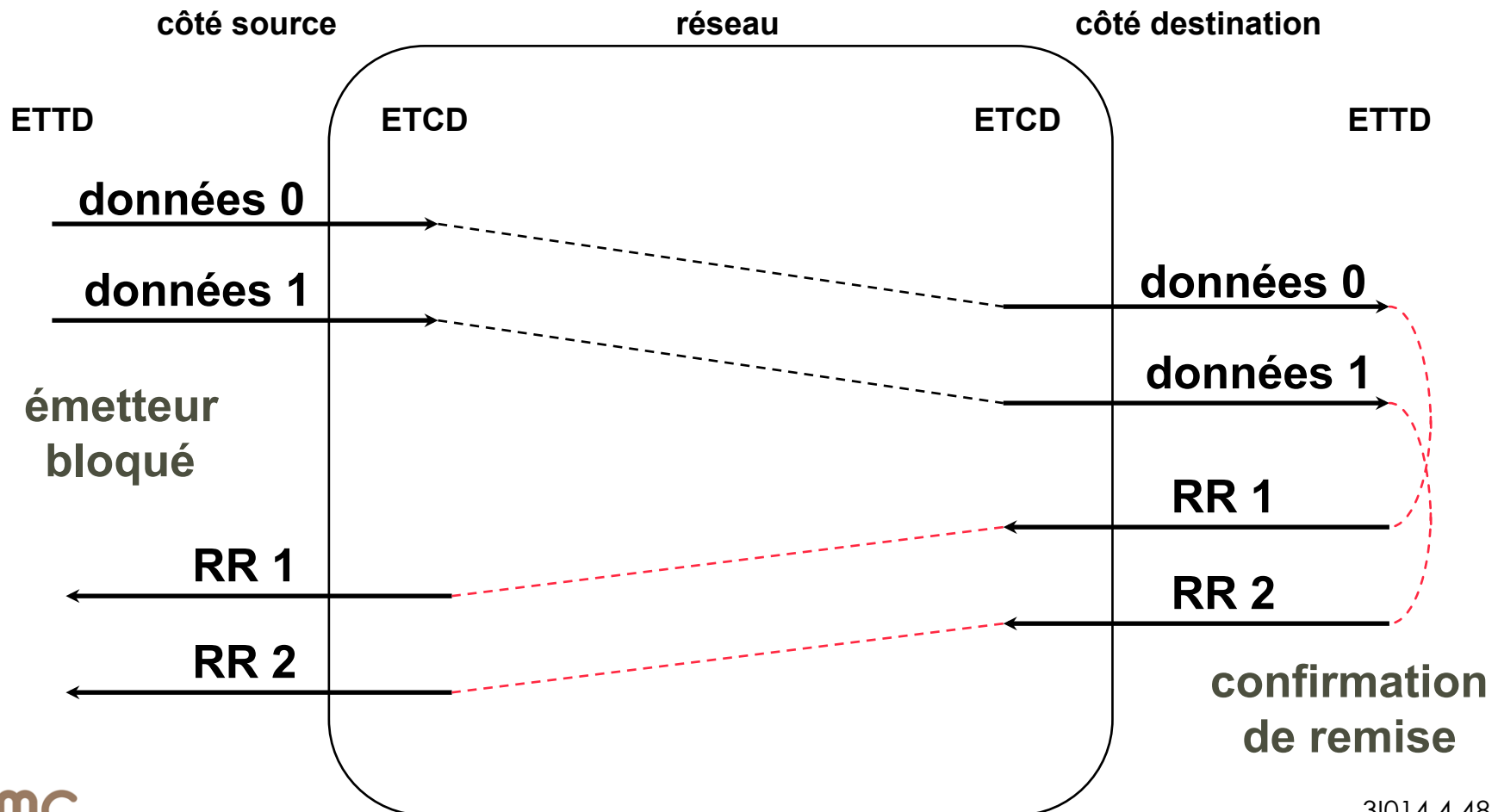
X.25 : le transfert de données

□ contrôle de flux

- à l'interface, une fenêtre est utilisée pour chaque sens de transmission
- $W = 2$ (par défaut)
- pour être émis, un paquet de données doit avoir son $P(S)$:
 $\text{dernier } P(R) \text{ reçu} \leq P(S) \leq \text{dernier } P(R) + W - 1$
- le contrôle peut se faire
 - en local ($D = 0$)
 - de bout-en-bout ($D = 1$)

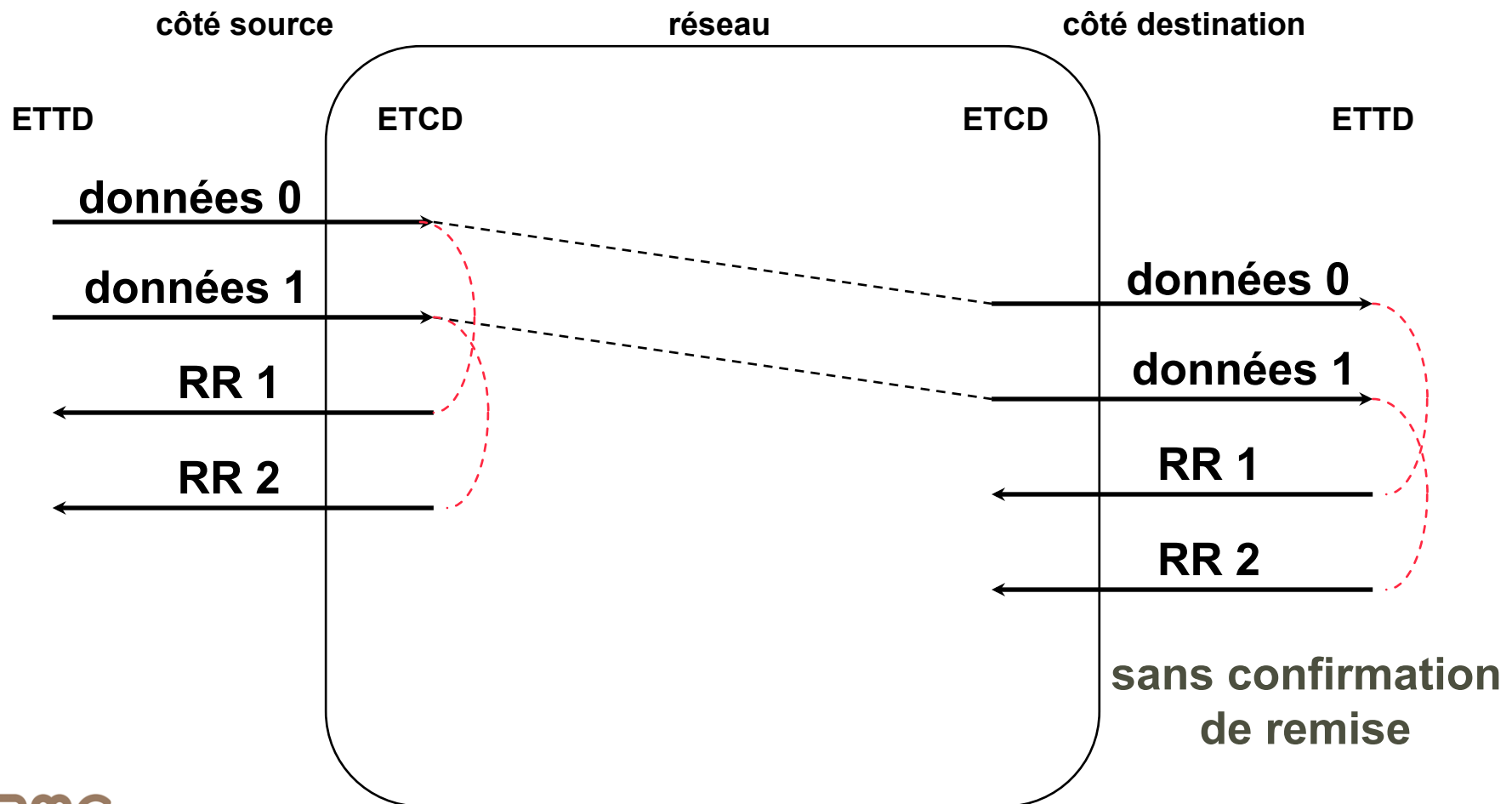
X.25 : le transfert de données

□ contrôle de flux de bout-en-bout

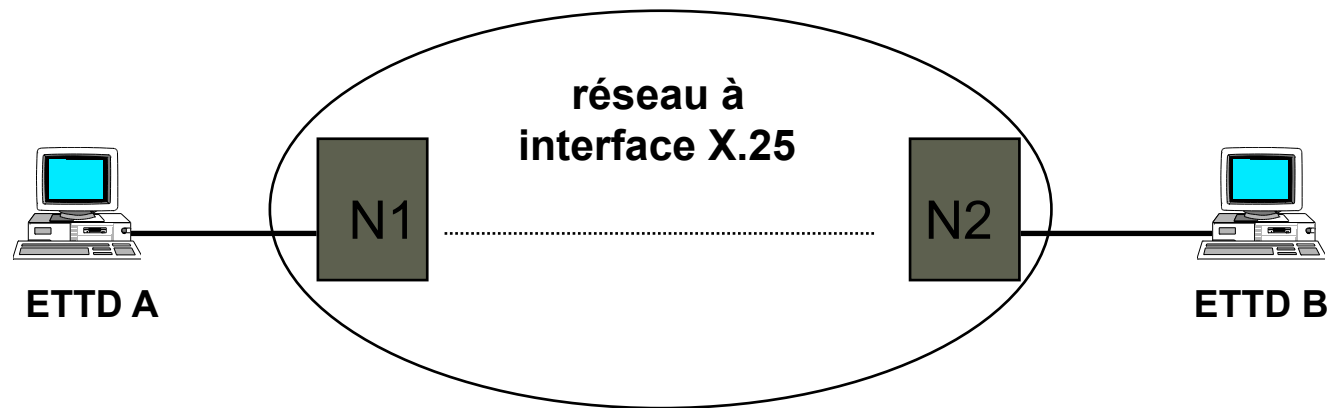


X.25 : le transfert de données

□ contrôle de flux en local

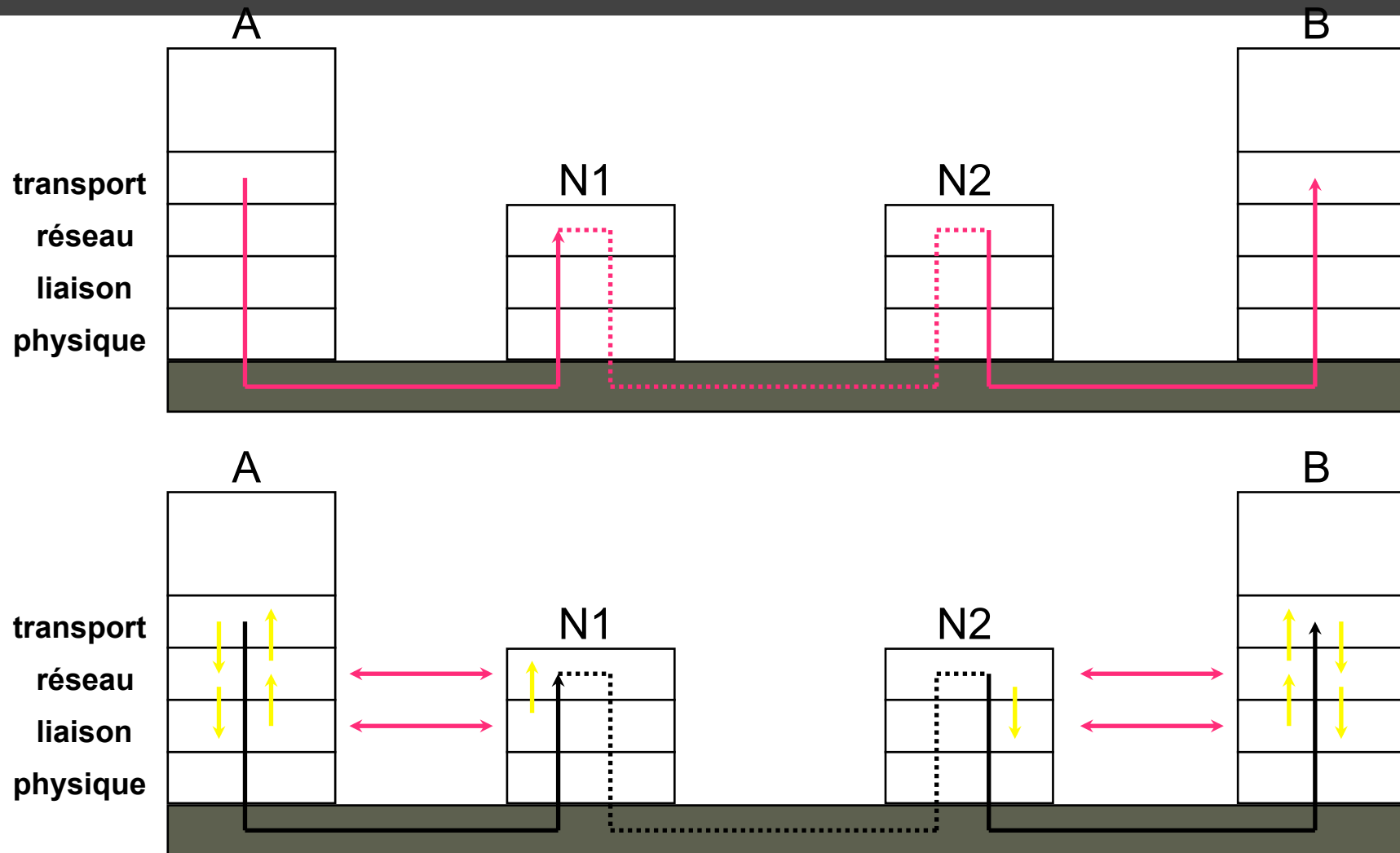


Illustration

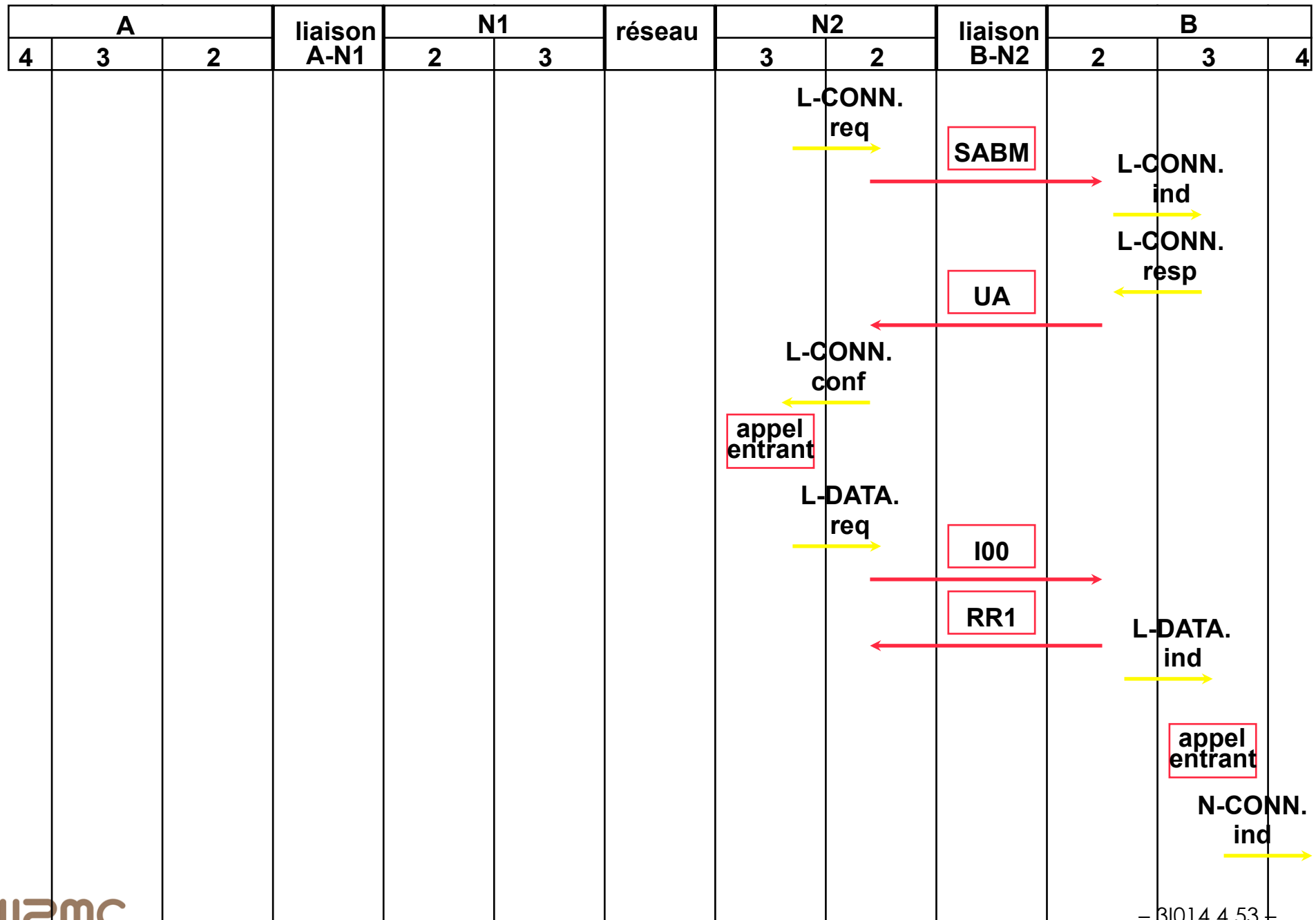


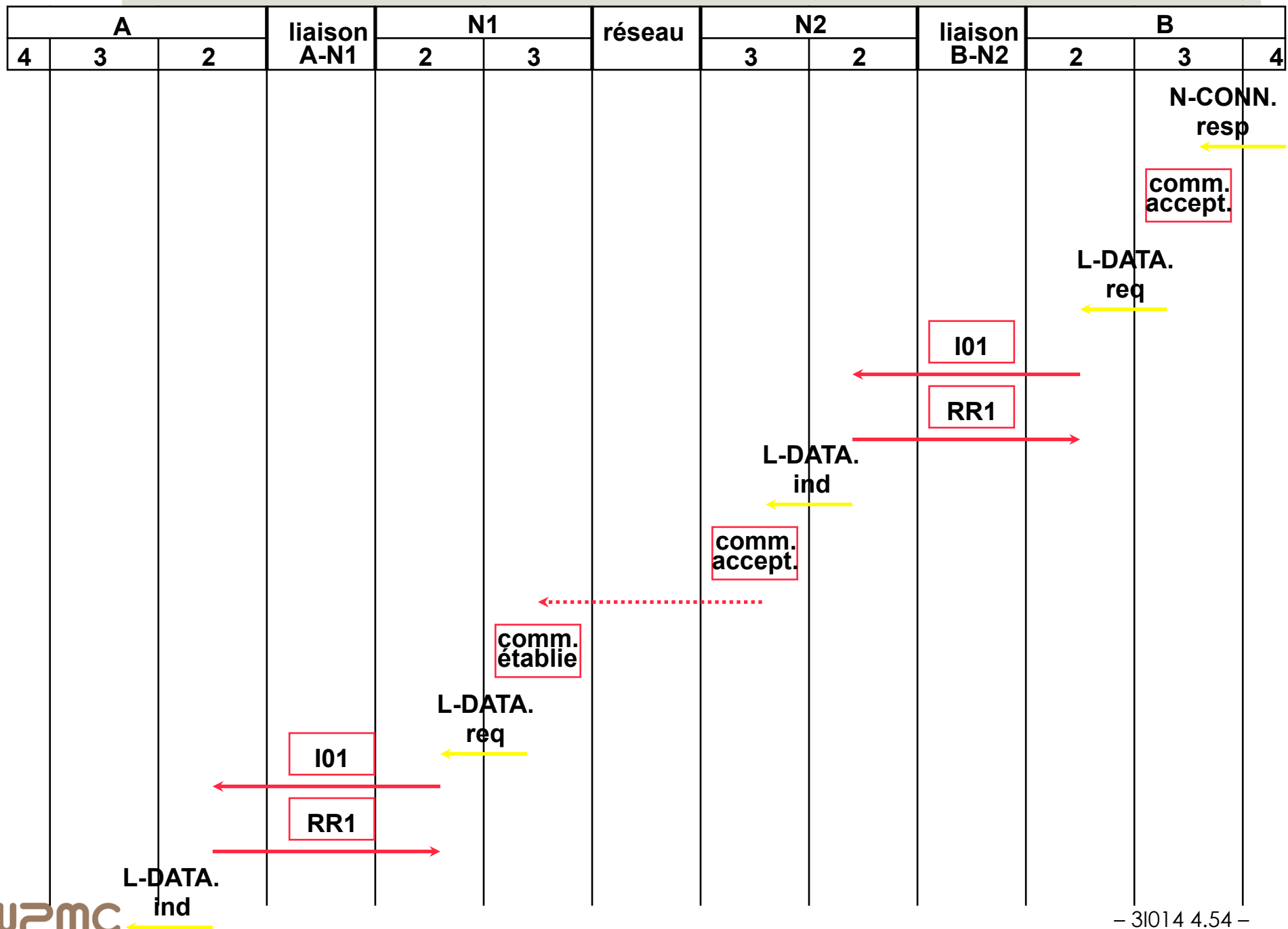
- Un utilisateur du service de réseau sur une machine A souhaite envoyer 256 octets de données à un utilisateur du service de réseau sur une machine B
- ↪ Primitives invoquées aux niveaux liaison et réseau ?
- ↪ Paquets et trames échangés ?

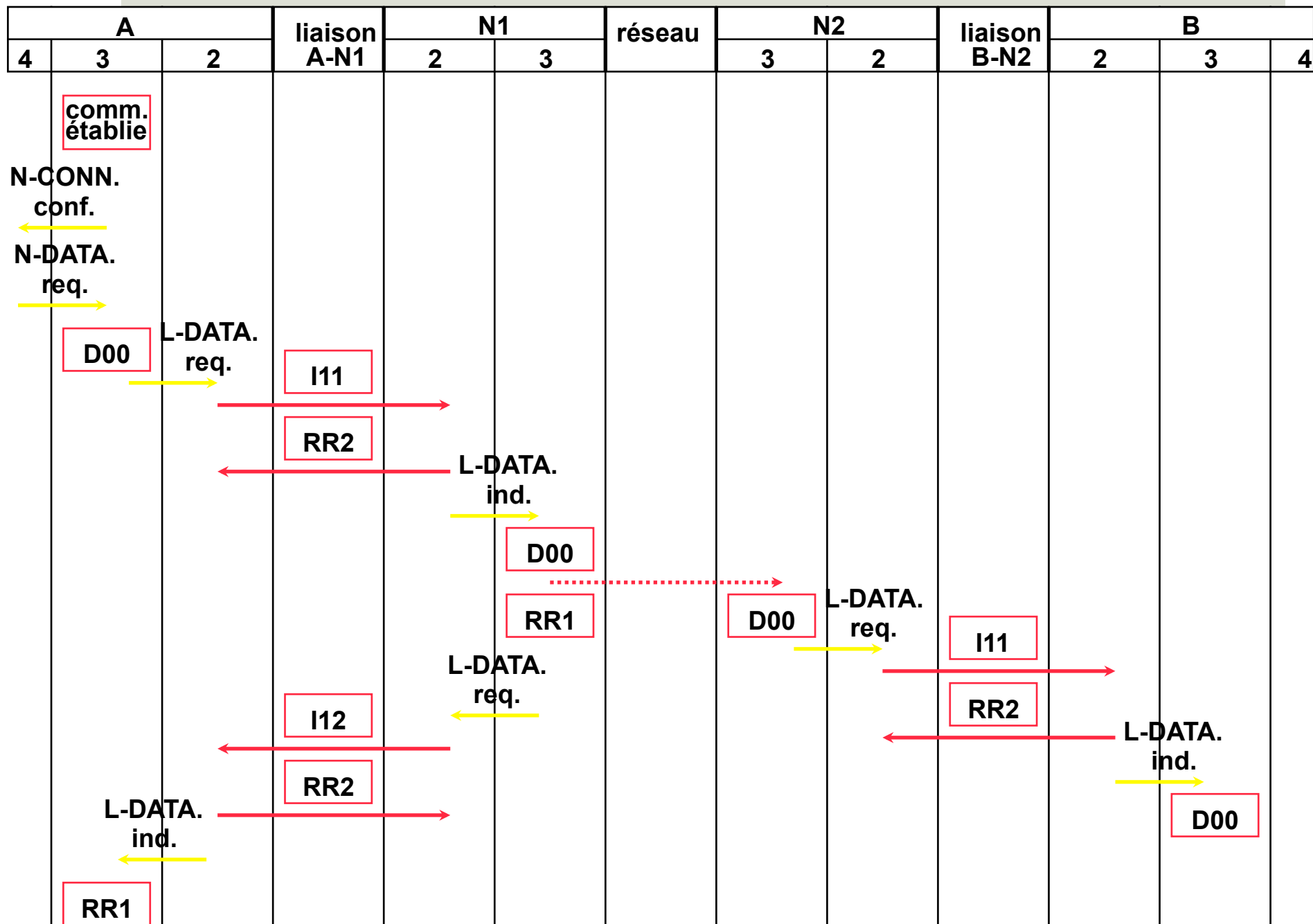
Illustration

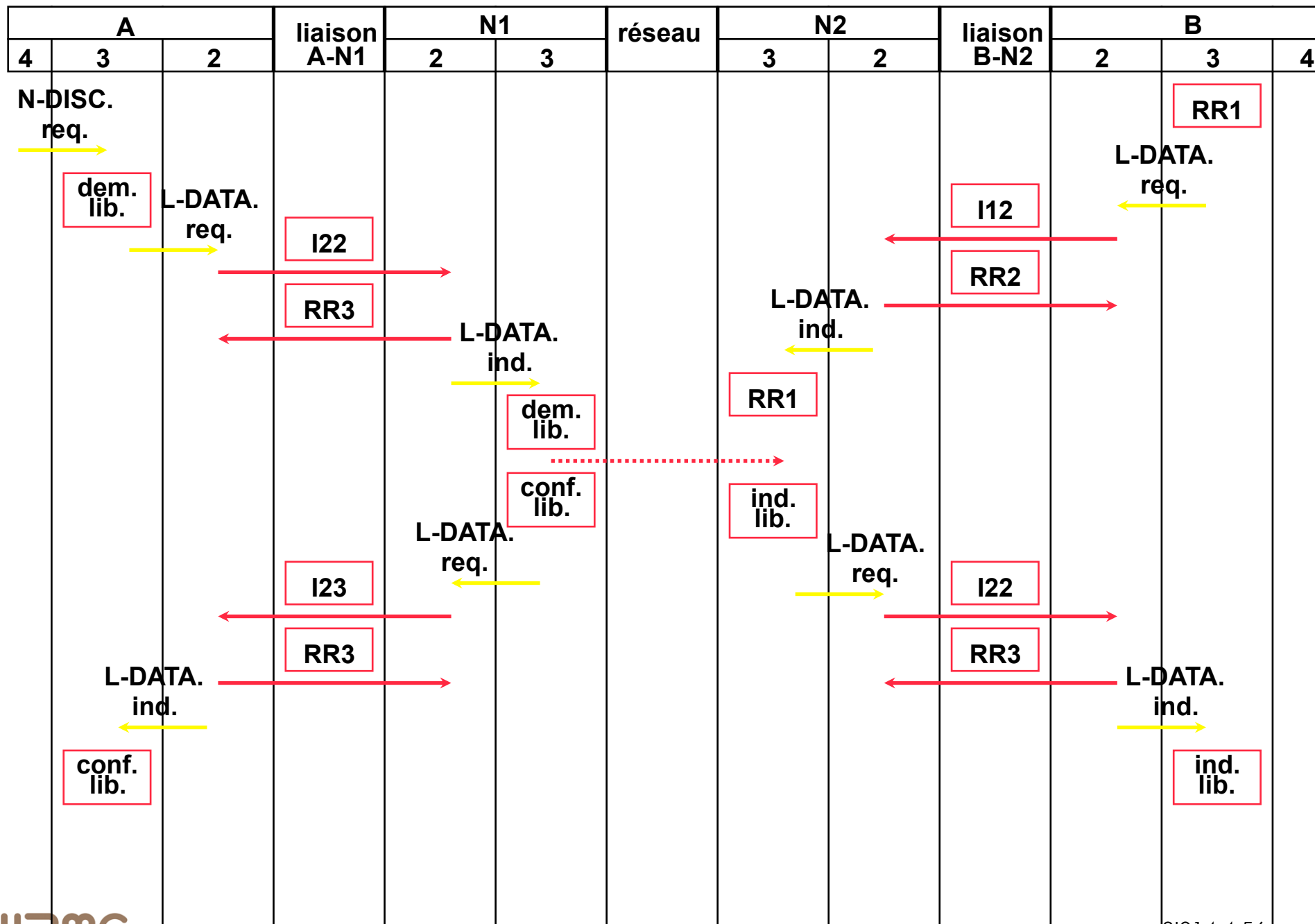












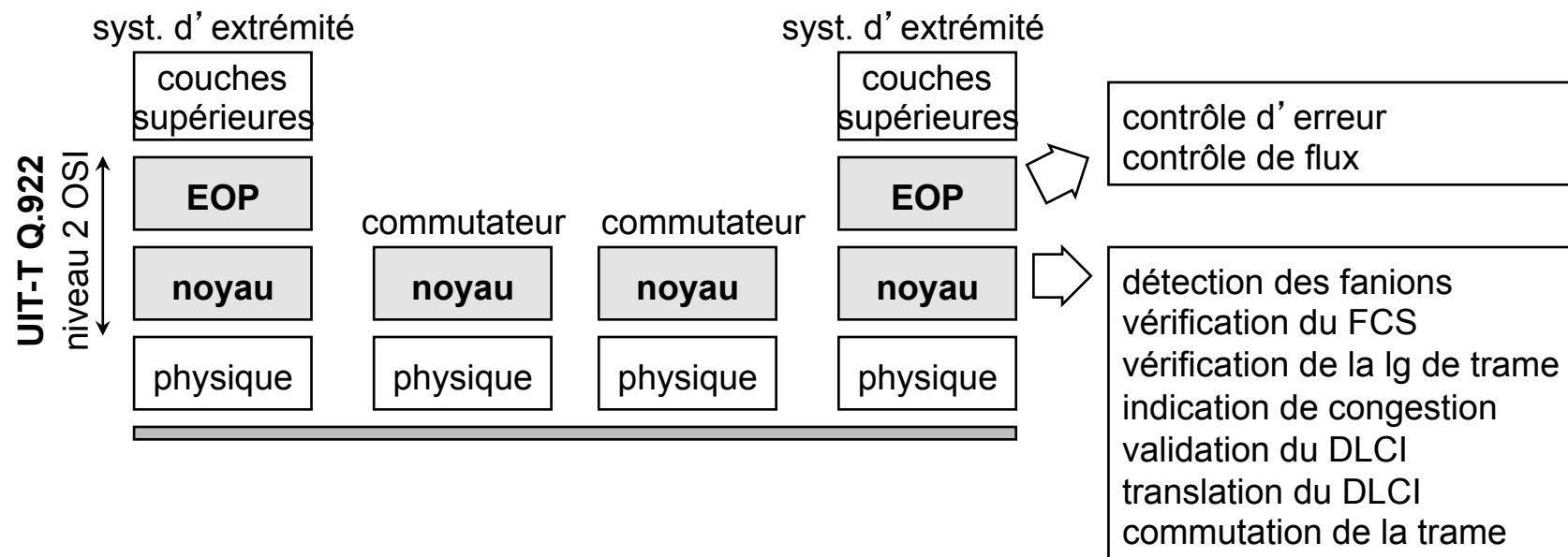


Relais de frames

- constat : évolution des technologies
 - liaisons
 - ordinateurs
- objectif : performances élevées à un coût réduit
- idée : utiliser des protocoles aussi simples que possible entre commutateurs
 - commuter au niveau trame
 - reporter le contrôle d'erreur et le contrôle de flux sur les systèmes d'extrémité
 - introduire une signalisation séparée

Relais de frames

- service orienté-connexion
 - analogie avec des liaisons louées virtuelles
 - analogie avec des CVP X.25
- services réduit à sa plus simple expression



Relais de frames

□ format de frame

0	1	1	1	1	1	1	0
DLCI						C/R	EA
DLCI			FE CN	BE CN	DE	EA	
user data							
FCS							
FCS							
0	1	1	1	1	1	1	0

DLCI - Data Link Connection Identifier

C/R - Command/Response

EA - Extended Address

FECN - Forward Explicit Congestion Notification

BECN - Backward Explicit Congestion Notification

DE - Discard Eligibility

□ détection des erreurs

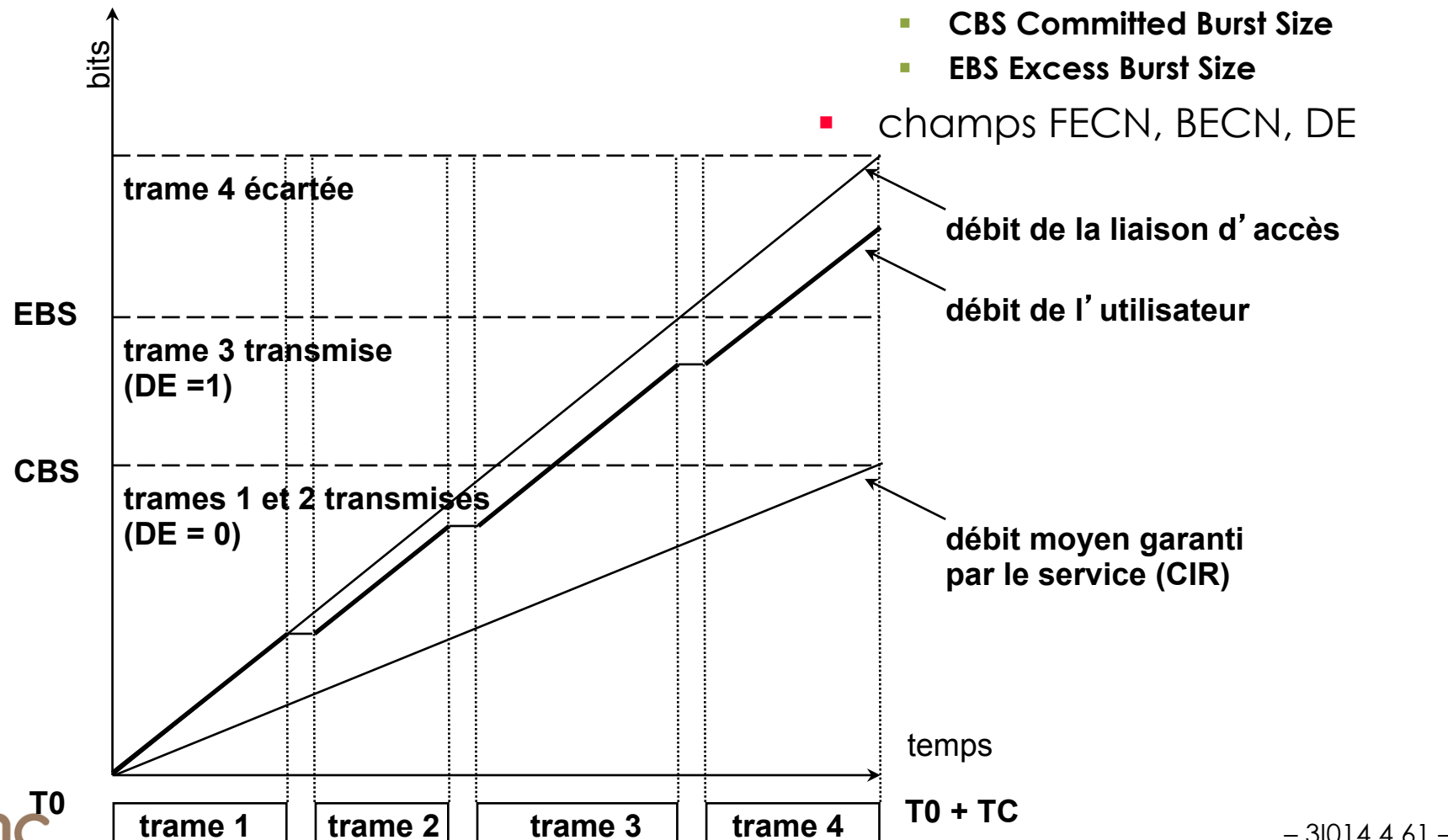
□ champ FCS

□ routage

□ champ DLCI

Relais de frames

□ contrôle de congestion



B-ISDN et ATM

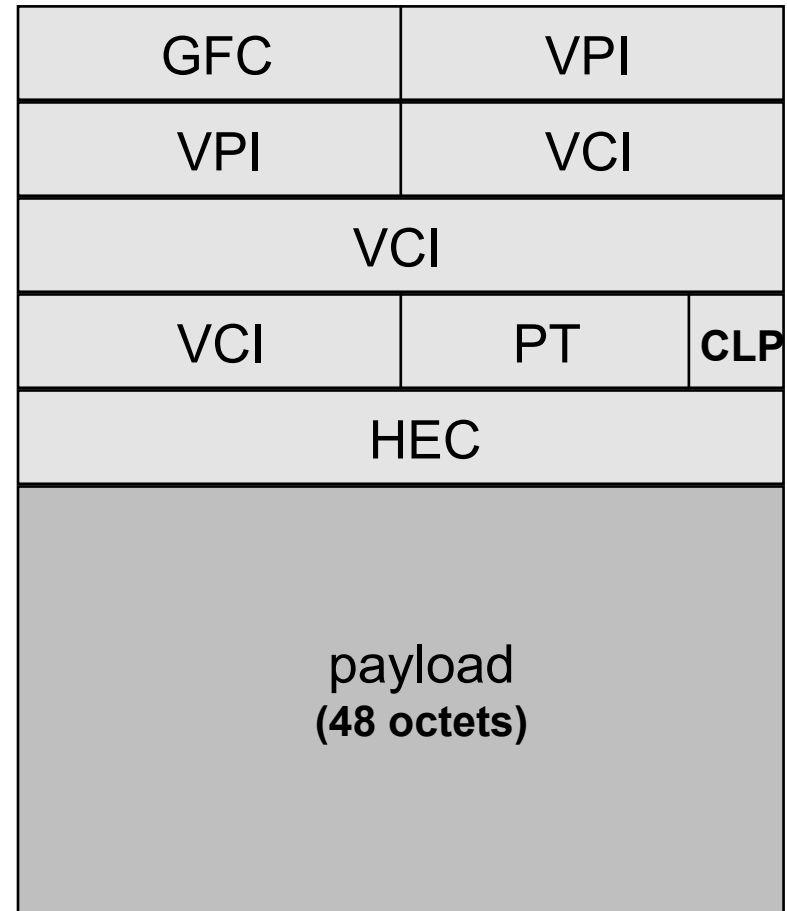
- Broadband Integrated Services Digital Network (RNIS-LB)
- objectif : créer un réseau universel unique
- moyens
 - fibre optique
 - ATM (*Asynchronous Transfer Mode*)
- principe de base : commutation de *cellules* - paquets de taille fixe et réduite (53 octets dont 5 d'en-tête)
 - flexibilité
 - efficacité
 - rapidité
 - multi-débits
 - diffusion

B-ISDN et ATM

- applications cible
 - téléphone
 - vidéo à la demande
 - TV câblée
 - audio
 - courrier électronique multimédia
 - interconnexion de LAN
 - calcul scientifique
 - etc.

ATM : la cellule

- transport des cellules
 - traitement de l'en-tête
 - translation VPI/VCI
 - (dé)multiplexage de cellules
- deux interfaces
 - UNI (User-Network Interface)
 - NNI (Network-Network Interface)
- format de la cellule (UNI)
 - Generic Flow Control
 - Virtual Path Identifier
 - Virtual Channel Identifier
 - Payload Type
 - Cell Loss Priority
 - Header Error Control
- mode connecté



ATM : vie d'une connexion

■ établissement

- la source déclare ses paramètres au réseau (@ dest., débit)
- reste-t-il des étiquettes et du débit entre les 2 extrémités
 - oui : réservation de ressources, mise à jour des tables
 - non : appel rejeté

■ transfert de données

■ libération de la connexion

ATM : les connexions

- VCC (*Virtual Channel Connection*)
 - de bout-en-bout entre la source et le destinataire
 - concaténation de VC (*Virtual Channel*)
 - identification sur chaque VC par un VCI

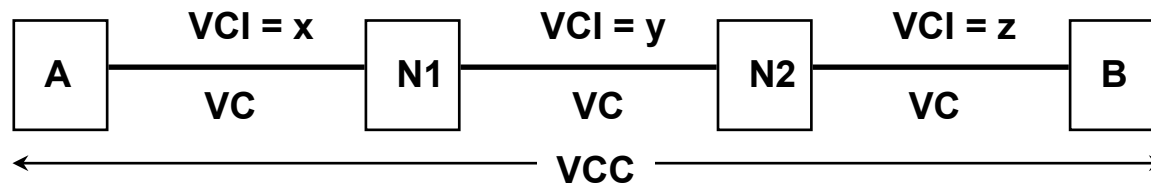
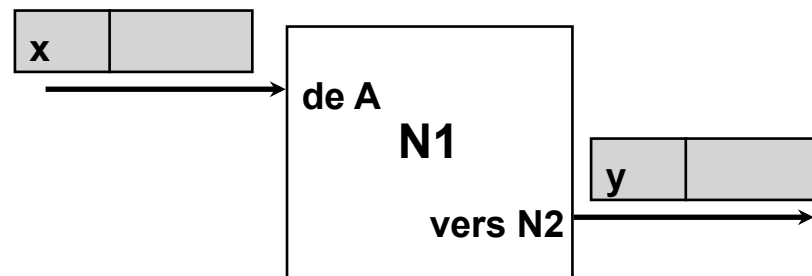


table de lookup de N1

	entrée	sortie
VCI	x	y
port	A	N2

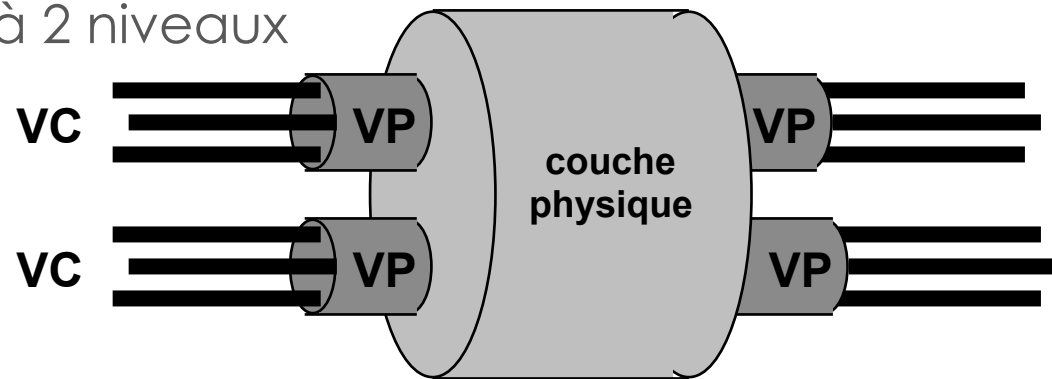


ATM : VC et VP

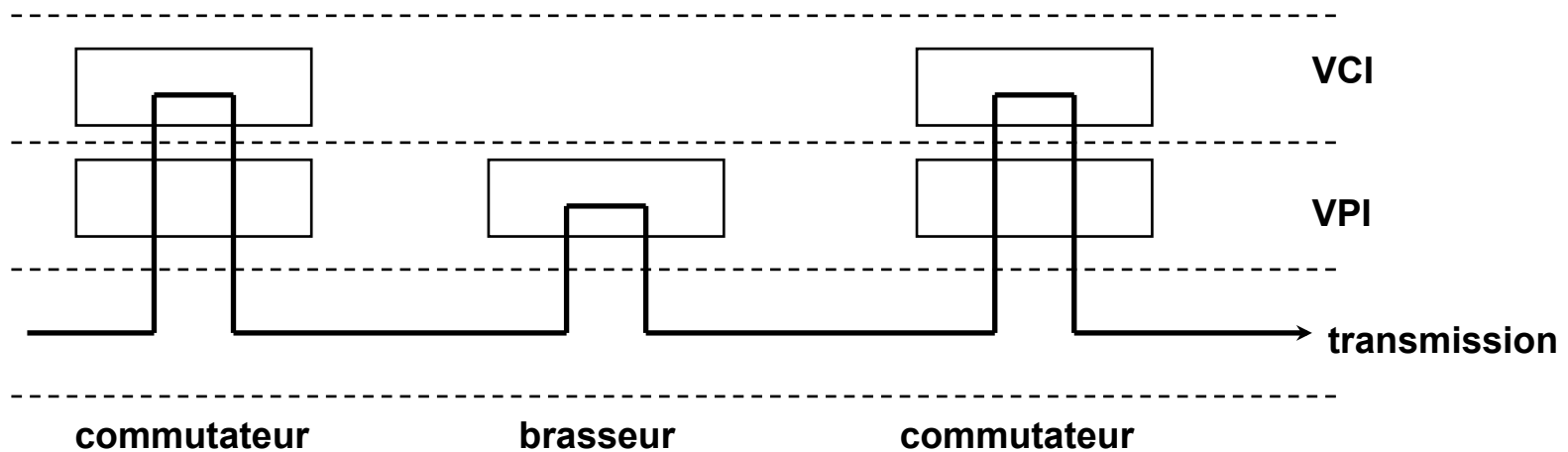
■ hiérarchie des connexions à 2 niveaux

■ VC (Virtual Channel)

■ VP (Virtual Path)



↪ commutation à 2 niveaux



Réseaux grande distance - Plan

- ▣ Réseaux grande distance
- ▣ La commutation
- ▣ Le multiplexage
- ▣ Les fonctions de réseau
- ▣ Acheminement par voie logique
- ▣ **Acheminement par datagramme**
 - ▣ Principes
 - ▣ L'exemple de RIP
 - ▣ L'exemple d'OSPF

Acheminement par datagramme

■ Principe

- étiquette = adresse du destinataire
- les paquets sont acheminés individuellement

■ Avantages et inconvénients

- 😊 l'expéditeur peut émettre sans accord préalable explicite du réseau
- 😞 fiabilité non garantie
 - 😞 déséquilibrage
 - 😞 pertes de paquets
- 😞 overhead (étiquette + longue)

La fonction d'adaptation des chemins

■ Principe

- réalisée grâce aux **protocoles de routage** qui maintiennent des **tables de routage** dans les nœuds du réseau
- une table de routage comporte au moins deux colonnes
 - la première pour la destination (ou pour le réseau de destination)
 - la seconde pour l'adresse du nœud correspondant au « saut » suivant sur le « meilleur » chemin vers la destination souhaitée.
- lorsqu'un datagramme arrive sur un routeur (lorsqu'un paquet d'appel arrive sur un commutateur), le routeur (le commutateur) consulte sa table de routage pour décider du prochain nœud pour ce paquet

Protocoles de routage : objectifs

- tout protocole de routage doit communiquer des informations sur la **topologie globale** du réseau à chaque routeur afin que celui-ci puisse prendre une **décision locale** de routage
- information globale
 - difficile à collecter
 - sujette à des modifications fréquentes
 - volumineuse
- ↘ minimisation des messages de contrôle échangés
- ↘ minimisation de l'espace des tables de routage
- ↘ robustesse : éviter les trous noirs, les boucles et les oscillations
- ↘ utilisation du chemin optimal

Protocoles de routage : classification

- centralisé vs. distribué
 - en routage centralisé, un nœud central se charge de collecter les informations sur chaque lien (on/off, utilisation, capacité), et de calculer la table de routage pour chaque nœud du réseau
 - en routage décentralisé, les routeurs coopèrent selon un protocole de routage distribué de façon à construire des tables de routage consistantes
- à la source vs. saut par saut
 - en routage à la source, un paquet peut transporter toute sa route
 - en routage saut par saut, un paquet ne véhicule que l'adresse de la destination
- déterministe vs. stochastique
 - en routage déterministe, tous les paquets vers une même destination seront retransmis au même nœud suivant
 - en routage stochastique, chaque routeur maintient plusieurs nœuds aval pour une même destination
- à chemin unique vs. à chemin multiple
 - en routage à chemin multiple, chaque routeur maintient une route principale et des routes alternatives qu'il peut utiliser en cas d'indisponibilités de la route principale
- statique vs. dynamique
 - en routage statique, le choix de la route ne dépend pas de l'actuelle mesure de l'état du réseau
 - en routage dynamique, le choix de la route dépend de l'actuelle mesure de l'état du réseau

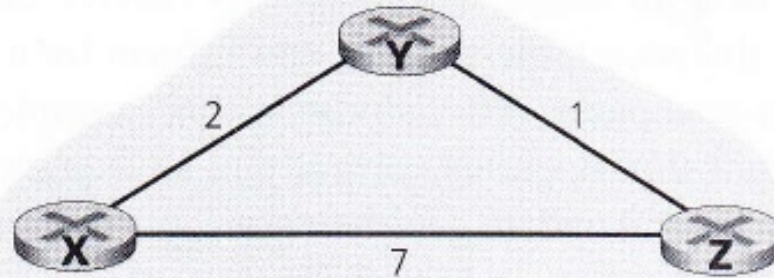
Algorithmes de routage

- Le routage est, par essence, un problème de la théorie des graphes
 - trouver le chemin de coût minimum entre deux nœuds quelconques, sachant que le coût d'un chemin est la somme des coûts des liens qui le composent
- Les algorithmes utilisés :
 - supposent que chaque routeur connaît l'adresse de chacun de ses voisins, ainsi que le coût pour l'atteindre ;
 - permettent à chaque routeur de déterminer l'information de routage globale, i.e. le prochain nœud pour atteindre chaque destination possible sur la route la plus courte, en échangeant de l'information de routage seulement avec ses voisins
- Deux classes d'algorithmes de routage
 - les algorithmes à vecteurs de distance : Bellman-Ford
 - les algorithmes à états des liens : Dijkstra

Routage à vecteur de distance

- Algorithme de routage de Bellman-Ford ou de Ford-Fulkerson
- Principe
 - Chaque nœud connaît la distance (i.e. le coût) vers chacun de ses voisins
 - Chaque nœud maintient une table dont chaque ligne correspond à une destination qu'il peut atteindre, le coût associé et le voisin via lequel qu'il puisse joindre cette destination
 - Chaque nœud forme un vecteur qui est la liste des destinations qu'il peut atteindre ainsi que les coûts associés et l'échanger avec les nœuds voisins
 - A la réception d'un vecteur du nœud voisin, chaque nœud essaie de voir s'il peut atteindre une nouvelle destination ou trouver un meilleur chemin en passant par ce voisin
- Exemples des protocoles de routage à vecteur de distance
 - RIP (Routing Information Protocol)
 - AODV (Ad hoc On-Demand Distance Vector)

Algorithme de Bellman-Ford



■ Au démarrage

Destination	Distance	Via
Y	2	direct
Z	7	direct

Table de X

Destination	Distance	Via
X	2	direct
Z	1	direct

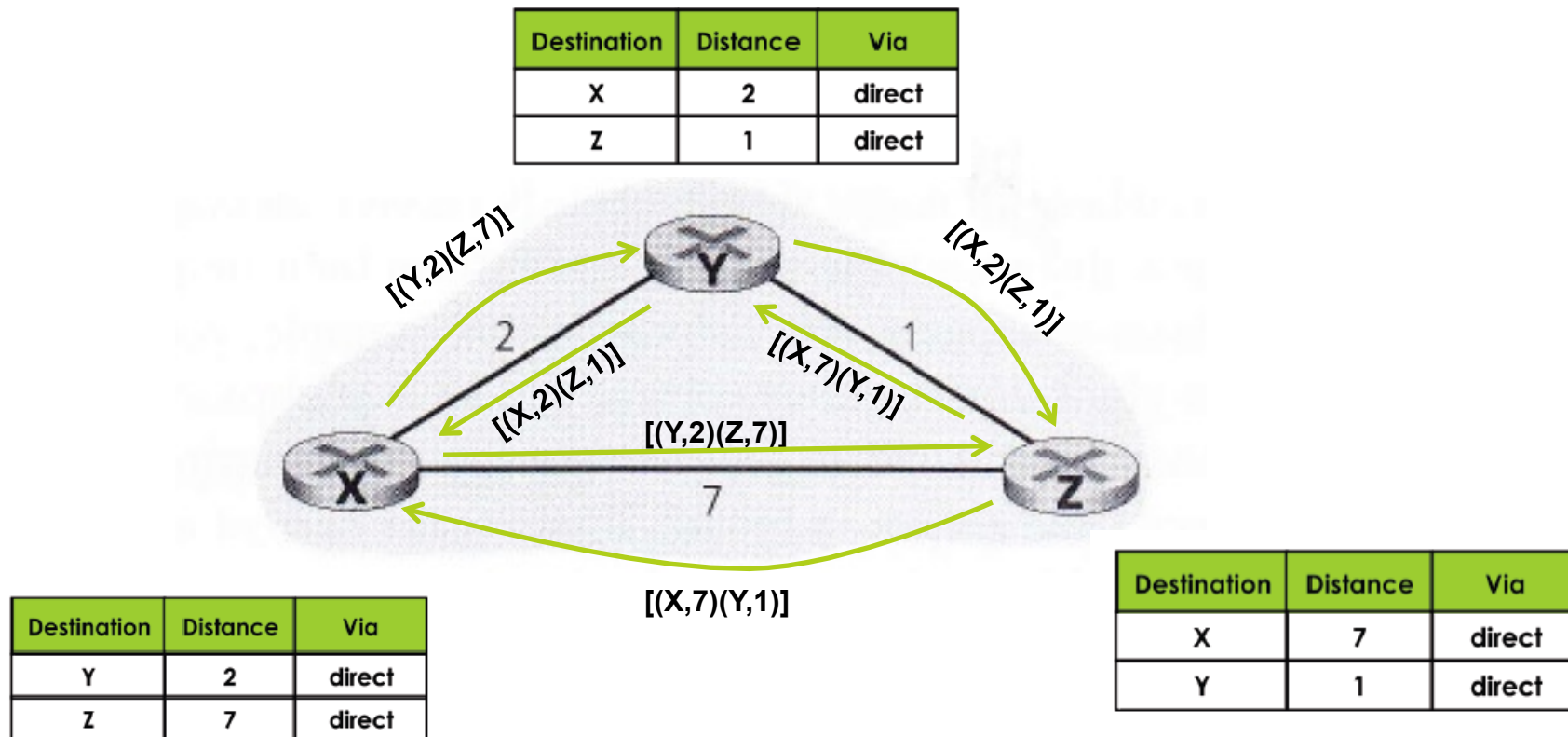
Table de Y

Destination	Distance	Via
X	7	direct
Y	1	direct

Table de Z

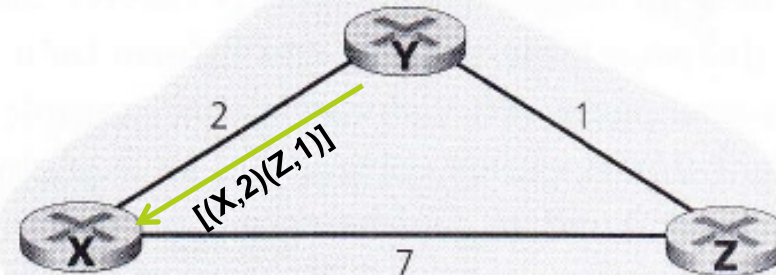
Algorithme de Bellman-Ford

- Les routeurs échangent des vecteurs de distance



Algorithme de Bellman-Ford

- Exemple de mettre à jour la table de routage du nœud X lors de la réception du vecteur de distance de Y



Destination	Distance	Via
Y	2	direct
Z	7	direct

Table de X originale

Calcul:
 $X \rightarrow Z = (X \rightarrow Y) + (Y \rightarrow Z)$
 $= 2 + 1$
 $= 3$

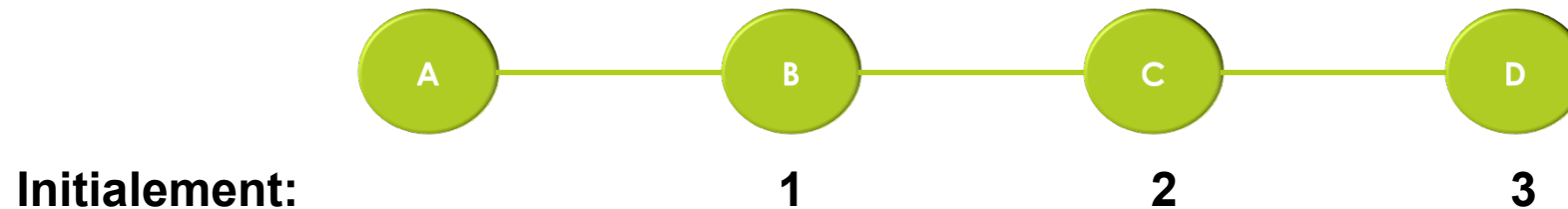
Conclusion:
 $(X \rightarrow Z \text{ via } Y) < (X \rightarrow Z \text{ en direct})$

Destination	Distance	Via
Y	2	direct
Z	3	Y

Table de X mise à jour

Le problème de comptage à l'infini

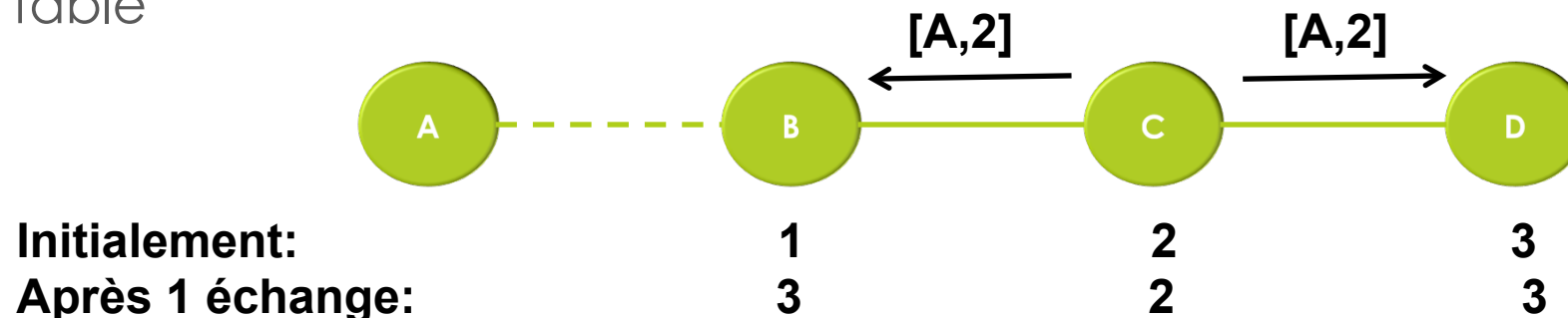
- Considérer la topologie suivante



- Les distances des nœuds B, C et D vers A sont respectivement 1, 2 et 3

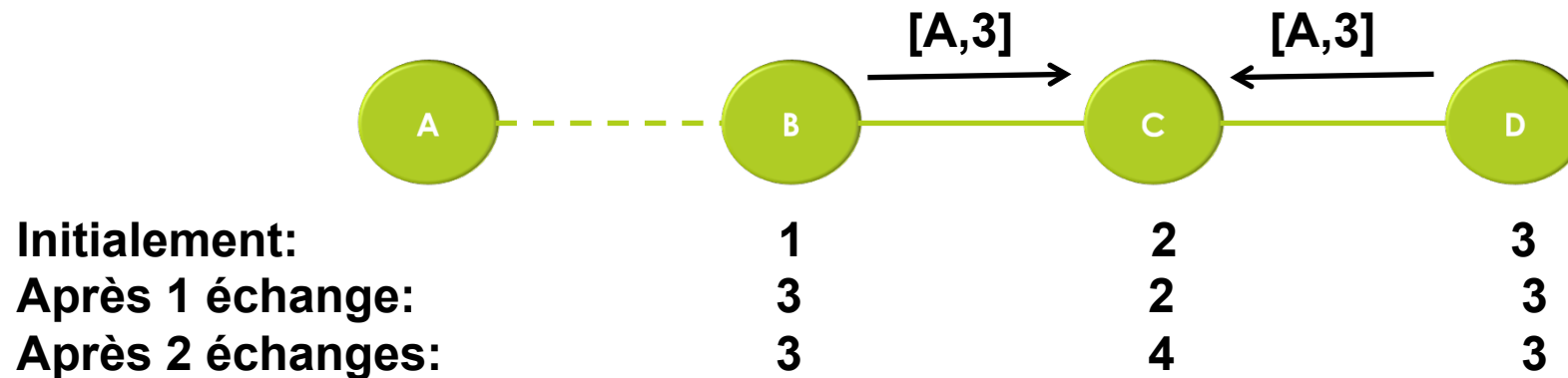
Le problème de comptage à l'infini

- La ligne A-B est interrompue
- B ne reçoit plus de nouvelles venant de A mais C annonce toujours qu'il peut joindre A avec une distance égale à 2
- B ne sait pas que le chemin annoncé par C passe par B lui-même car C peut très bien avoir plusieurs lignes distinctes vers A
- B pense qu'il puisse atteindre A en passant par C et met à jour sa table




Le problème de comptage à l'infini

- Après 2 échanges, C remarque que chacun de ses voisins annonce une route vers A d'une distance de 3
- C met à jour sa table avec une distance de 4 pour aller à A



Le problème de comptage à l'infini

- L'évolution des coûts du chemin vers A en fonction du nombre d'échange des vecteurs de distance est comme la suivante



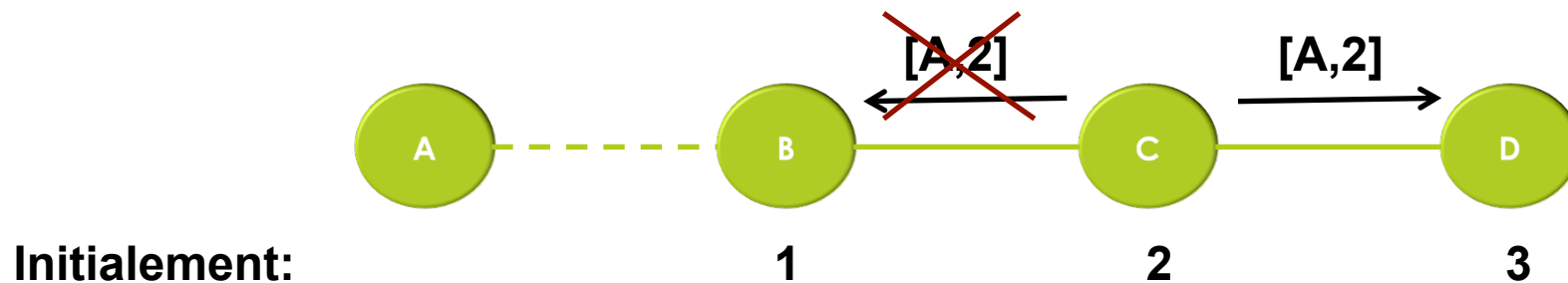
	A	B	C	D
Initialement:		1	2	3
Après 1 échange:		3	2	3
Après 2 échanges:		3	4	3
Après 3 échanges:		5	4	5
Après 4 échanges:		5	6	5
Après 5 échanges:		7	6	7
Après 6 échanges:		7	8	7

.....

- Les routeurs convergent lentement vers une réponse correcte pour détecter que A n'est plus joignable → C'est le problème de comptage à l'infini

Le problème de comptage à l'infini

- Une solution proposée: la technique de l'Horizon partagé (split horizon)
- Un routeur ne communique jamais le coût vers une destination à son voisin si ce voisin est le prochain nœud vers cette destination



Routage à états des liens

■ Principe

- Chaque nœud connaît l'état des liens entre lui et les voisins
- Chaque nœud diffuse cette liste (i.e. les états des liens) à tous les nœuds dans le réseau
- Avec les informations sur les états des liens venant des autres nœuds dans le réseau, chaque nœud possède d'une topologie complète du réseau
- Chaque nœud applique l'algorithme de Dijkstra sur cette topologie pour trouver les chemins les plus courts vers chacune des destinations

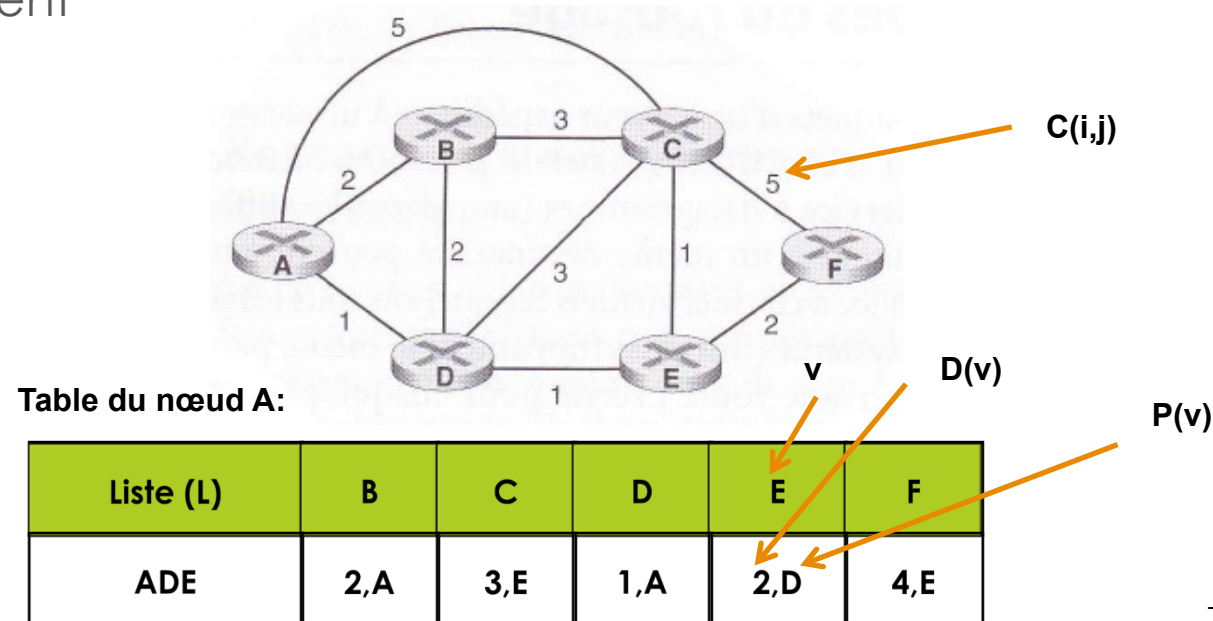
■ Exemples des protocoles de routage à états des liens

- OSPF (Open Shortest Path First)
- OLSR (Optimized Link State Routing Protocol)

Algorithme de Dijkstra

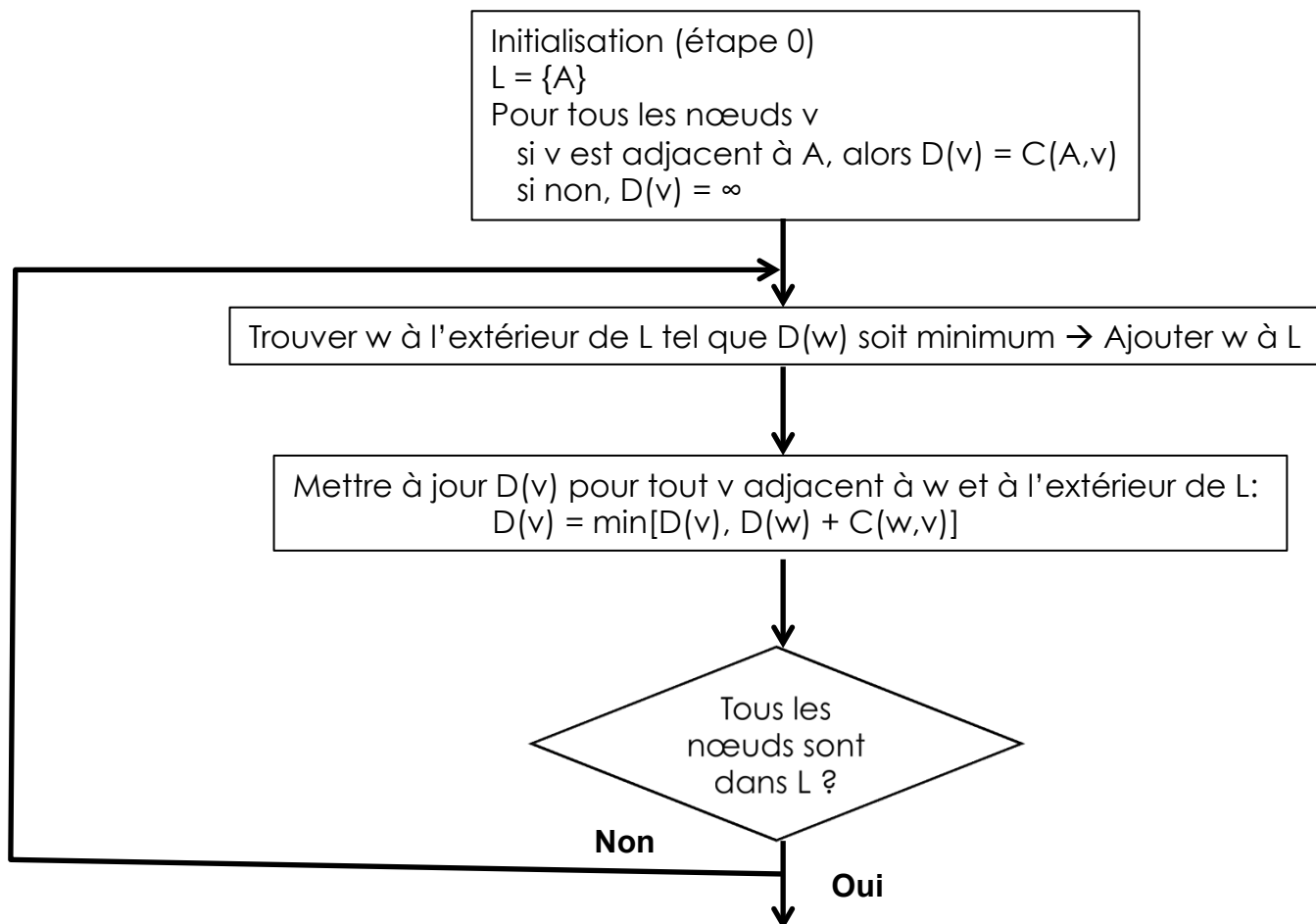
- Chaque nœud maintient une table en utilisant les paramètres suivants:
 - $C(i,j)$: le coût du lien direct (i,j)
 - $D(v)$: le coût du chemin du nœud source à la destination v
 - $P(v)$: le nœud précédent de la destination v
 - L est la liste des destinations donc les chemins les plus courts sont trouvés définitivement

■ Exemple:



Algorithme de Dijkstra

- Pour trouver les chemins les plus courts vers toutes les destinations



Algorithme de Dijkstra - Exemple

■ Etape 0

- A est le nœud source
- La liste L ne contient qu'A
- De A, nous pouvons aller à B (2), C(5) et D(1) en 1 saut
- D sera rajouté dans la liste (L) car le chemin A→D a le coût le plus faible

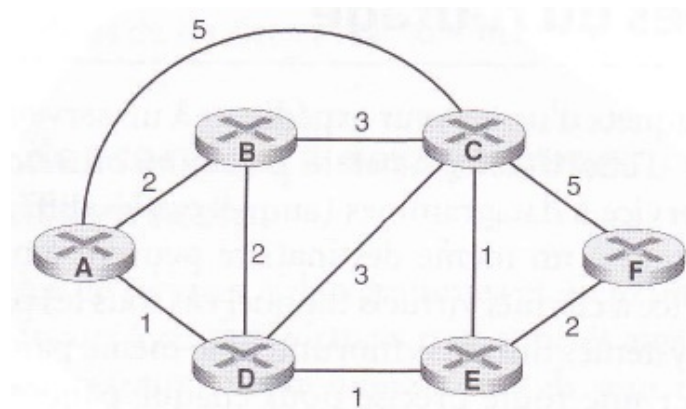


Table du nœud A:

Étape	Liste (L)	B	C	D	E	F
0	A	2,A	5,A	1,A	∞	∞

Algorithme de Dijkstra - Exemple

- Etape 1
 - D est rajouté dans la liste L
 - En essayant d'aller 1 saut plus loin à partir de D, nous découvrons la nouvelle destination E et trouvons un meilleur chemin pour joindre C
 - Les destinations ayant le coût le plus petit sont B et E. Dans cet exemple, E sera rajouté dans la liste L

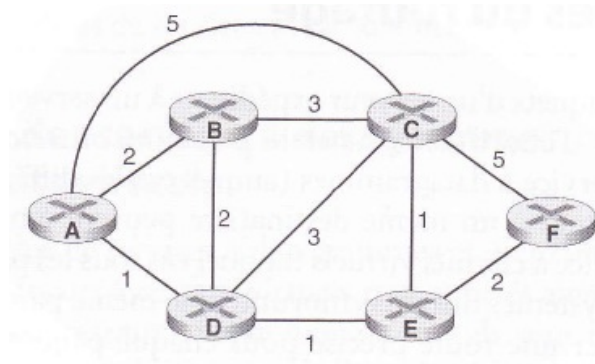


Table du nœud A:

Etape	Liste (L)	B	C	D	E	F
0	A	2,A	5,A	1,A	∞	∞
1	AD	2,A	4,D	-	2,D	∞

Algorithme de Dijkstra - Exemple

■ Etape 2

- E est rajouté dans la liste L
- En essayant d'aller plus loin en passant par E, A trouve un meilleur chemin pour atteindre C et découvre la nouvelle destination F
- B sera rajouté dans la liste L

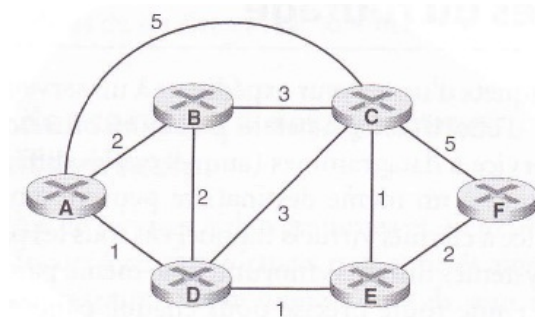


Table du nœud A:

Etape	Liste (L)	B	C	D	E	F
0	A	2,A	5,A	1,A	∞	∞
1	AD	2,A	4,D	-	2,D	∞
2	ADE	2,A	3,E	-	-	4,E

Algorithme de Dijkstra

■ Etape 3

- B est rajouté dans la liste L
- Pas de meilleur chemin trouvé en passant par B
- C sera rajouté dans la liste L

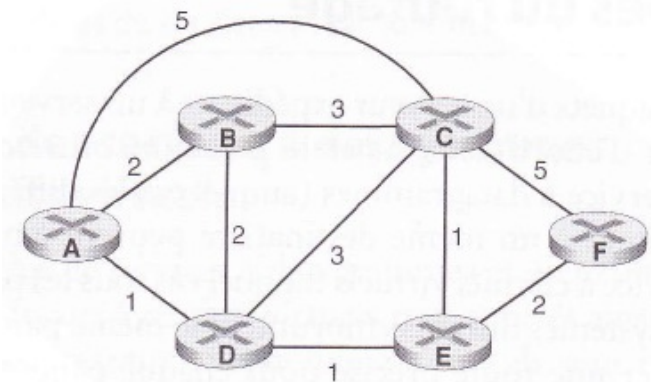


Table du nœud A:

Etape	Liste (L)	B	C	D	E	F
0	A	2,A	5,A	1,A	∞	∞
1	AD	2,A	4,D	-	2,D	∞
2	ADE	2,A	3,E	-	-	4,E
3	ADEB	-	3,E	-	-	4,E

Algorithme de Dijkstra - Exemple

■ Etape 4

- C est rajouté à la liste L
- En essayant d'aller plus loin par C, il n'y a pas de meilleur chemin trouvé
- F sera rajouté dans la liste L

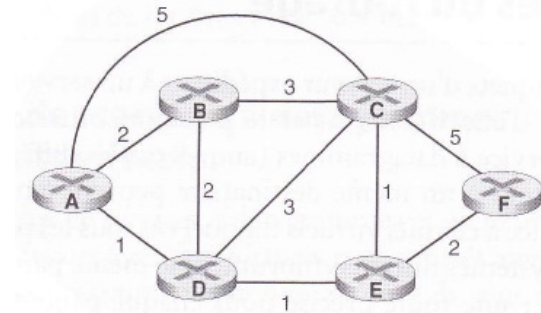


Table du nœud A:

Etape	Liste (L)	B	C	D	E	F
0	A	2,A	5,A	1,A	∞	∞
1	AD	2,A	4,D	-	2,D	∞
2	ADE	2,A	3,E	-	-	4,E
3	ADEB	-	3,E	-	-	4,E
4	ADEBC	-	-	-	-	4,E

Algorithme de Dijkstra - Exemple

■ Etape 5

- Tous les nœuds sont déjà dans la liste L
- L'algorithme est terminé

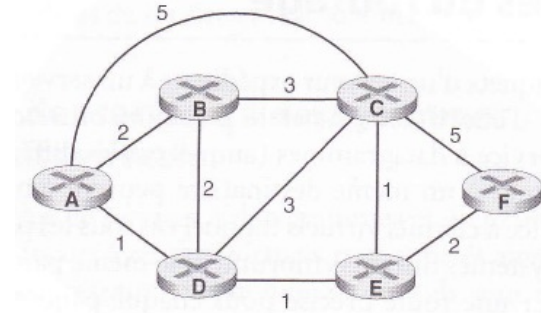


Table du nœud A:

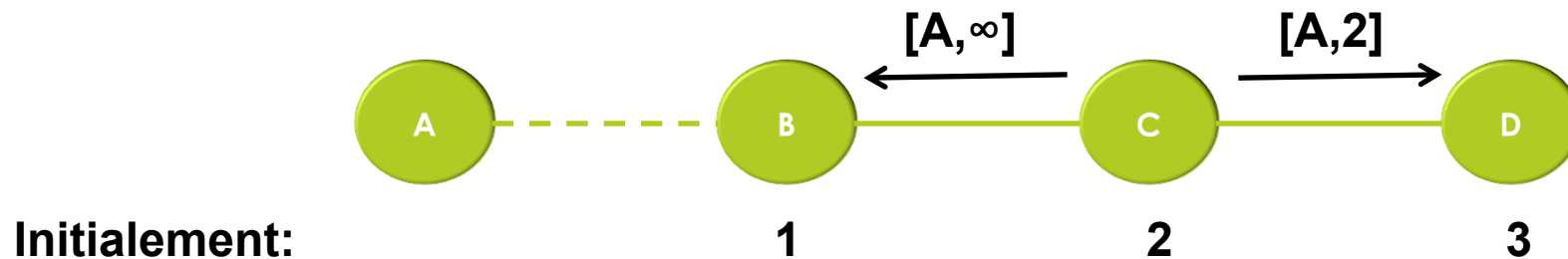
Etape	Liste (L)	B	C	D	E	F
0	A	2,A	5,A	1,A	∞	∞
1	AD	2,A	4,D	-	2,D	∞
2	ADE	2,A	3,E	-	-	4,E
3	ADEB	-	3,E	-	-	4,E
4	ADEBC	-	-	-	-	4,E
5	ADEBCF	-	-	-	-	-

Protocole RIP

- Routing Information Protocol
 - RFC 1058, 2453
 - Protocole de routage à vecteur de distance
 - Le coût du chemin est mesuré en nombre de sauts (15 au maximum)
 - Un coût de 16 équivaut l'infinie
- Message de réponse RIP/ message d'annonce RIP
 - Les routeurs échangent des information de routage (les vecteurs de distance) avec leurs voisins directs toutes les 30 seconds
 - Un routeur est déclaré hors service si un voisin n'obtient plus d'information de lui depuis plus de 180 secondes
 - Le vecteur de distance est une liste comprenant jusqu'à 25 réseaux de destination
- Message de requête RIP
 - Permettre à un nœud de demander à un voisin d'envoyer une partie ou toute la table de routage

Protocole RIP

- Utiliser la technique « Horizon partagé avec antidote » (Split horizon with poisoned reverse), une variante de la technique « Split horizon », pour éviter le problème du comptage à l'infini
- Un routeur communique toujours la distance vers une destination à son voisin dans le cas où il faut passer par ce voisin comme le prochain nœud mais le coût est mis directement à l'infini
- Ca oblige le protocole de former différents vecteurs pour envoyer sur différentes liaisons



Protocole OSPF

- Open Shortest Path First
 - RFC 1131, 2328
 - Protocole de routage à état de lien
 - Le coût du chemin n'est pas forcément le nombre de sauts (à définir par l'administrateur, e.g. capacité du lien)
- Message Hello
 - Échangés entre les voisins pour vérifier le bon état des liens
- Message d'annonce OSPF
 - Les routeurs communiquent les informations de routage (les états des liens) à tous les routeurs et non seulement à ses voisins directs
 - Les messages d'annonce sont échangés d'une manière périodique (au moins une fois toutes les demi-heures) et à chaque changement d'état d'une liaison

Protocole OSPF

- Message d'acquittement OSPF
 - Les messages d'annonce OSPF sont explicitement acquittés
 - Une annonce non acquittée sera retransmise
- Message de requête OSPF
 - Permettre à un nœud de demander à un voisin une partie de la topologie du réseau

Etat de notre architecture

