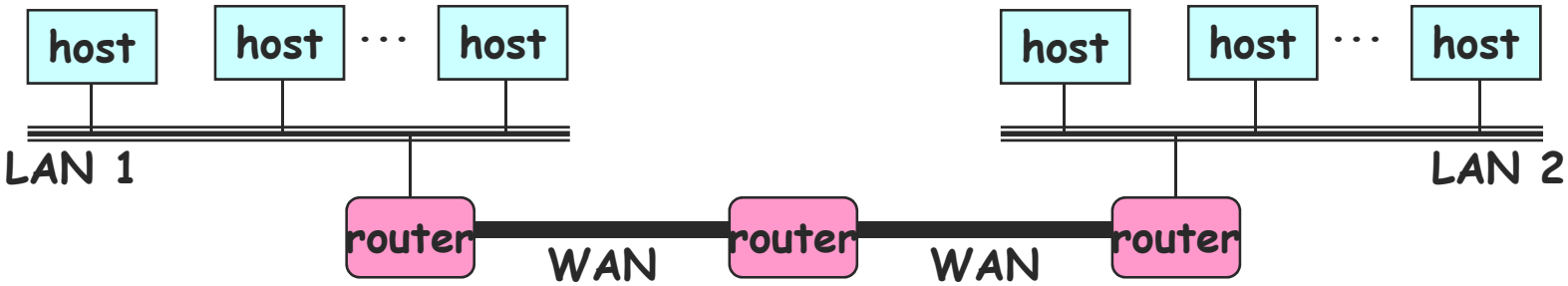
interconnexion de réseaux

Plan

- InternetWorks
- Types de service
- Congestion
- Fragmentation/réassemblage

InternetWorks

- InternetWorks
  - Plusieurs LANs incompatibles peuvent être connectés par des routeurs
  - Ces réseaux connectés sont appelés *internetworks*
    - L'Internet peut être vue comme "internetwork of internetworks"
- Ensemble des moyens permettant à différents utilisateurs reliés à différents réseaux autonomes (LAN ou WAN) de communiquer entre eux



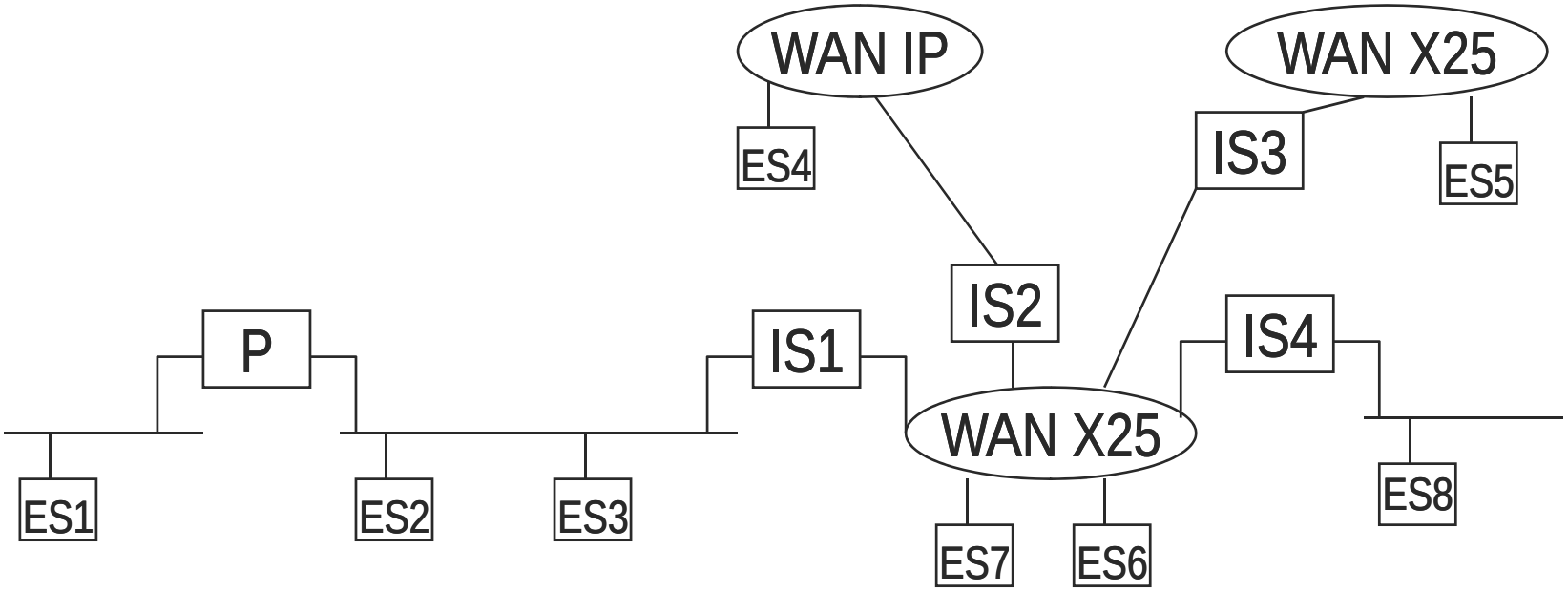
LAN 1 et LAN 2 peuvent être deux réseaux locaux complètement différents et incompatibles (exp. Ethernet et ATM)

InternetWorks

- Objectifs d'un "internetwork"
  - Assurer une connectivité de bout en bout "End-to-end"
  - Assurer une continuité et une intégration de la communication en fournissant une vision abstraite de l'« internetwork » indépendante de la topologie physique ou logique
  - Fonctionner dans un environnement hétérogène
  - « Scalable »: passage à grande échelle

InternetWorks

- Architecture
  - ES : « End System » ou hôtes
    - Nœuds supportant les applications utilisateurs
  - IS : « Intermediate System »
    - Nœuds relais servant à la communication entre ES

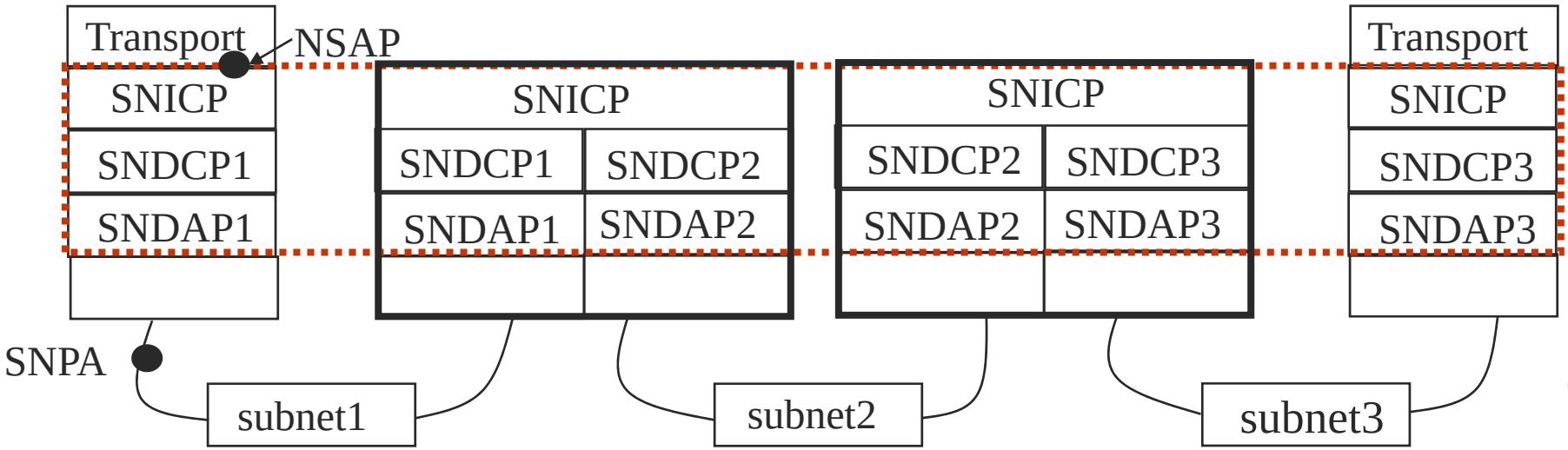


InternetWorks

- Problèmes d'interconnexion posés
  - Problèmes liés à l'hétérogénéité
    - Des réseaux raccordés : mélange de réseaux public et privés
    - Des services réseaux (mode connecté, mode non connecté)
    - Des mécanismes de contrôle de flux et de congestion
    - De l'adressage : chaque réseau a son propre adressage
    - Des paramètres de QoS offerts (délai , sécurité; taux d'erreur, …etc)
    - Des tailles maximales des paquets
    - Des rapports d'erreurs: varient selon le type de réseau
    - Dés procédures de routage : entre réseaux
  - Agrégation importante de trafic ➔ risque de congestion
  - Performances des réseaux interconnectés
  - Tarification du trafic

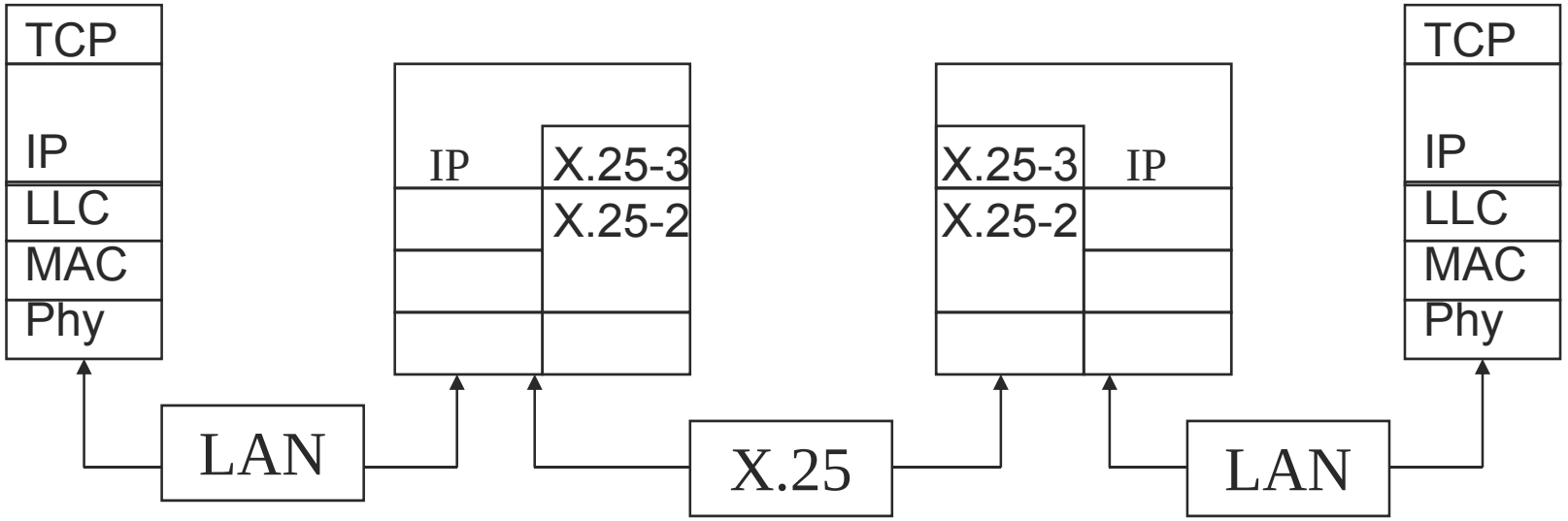
InternetWorks

- Structure de la couche réseau (IONL « Internal Organization of the Network Layer », ISO 8648)
  - **SNICP “*SubNet Independent Convergence Protocol*”** : protocole commun, fragmentation/ré-assemblage, routage *forwarding*, ISOIP..
  - ***SNDAP “SubNet Dependent Access Protocol”***, spécifique à chaque “subnet”, X25 …
  - ***SNDCP “SubNet Dependent Convergence Protocol”*** : réalise les fonctions de correspondance entre SNICP & SNDAP en particulier la correspondance entre les adresses ***SNPA “SubNetwork Point of Attachment ”*** et ***NSAP “Network Service Address Point”*** …



InternetWorks

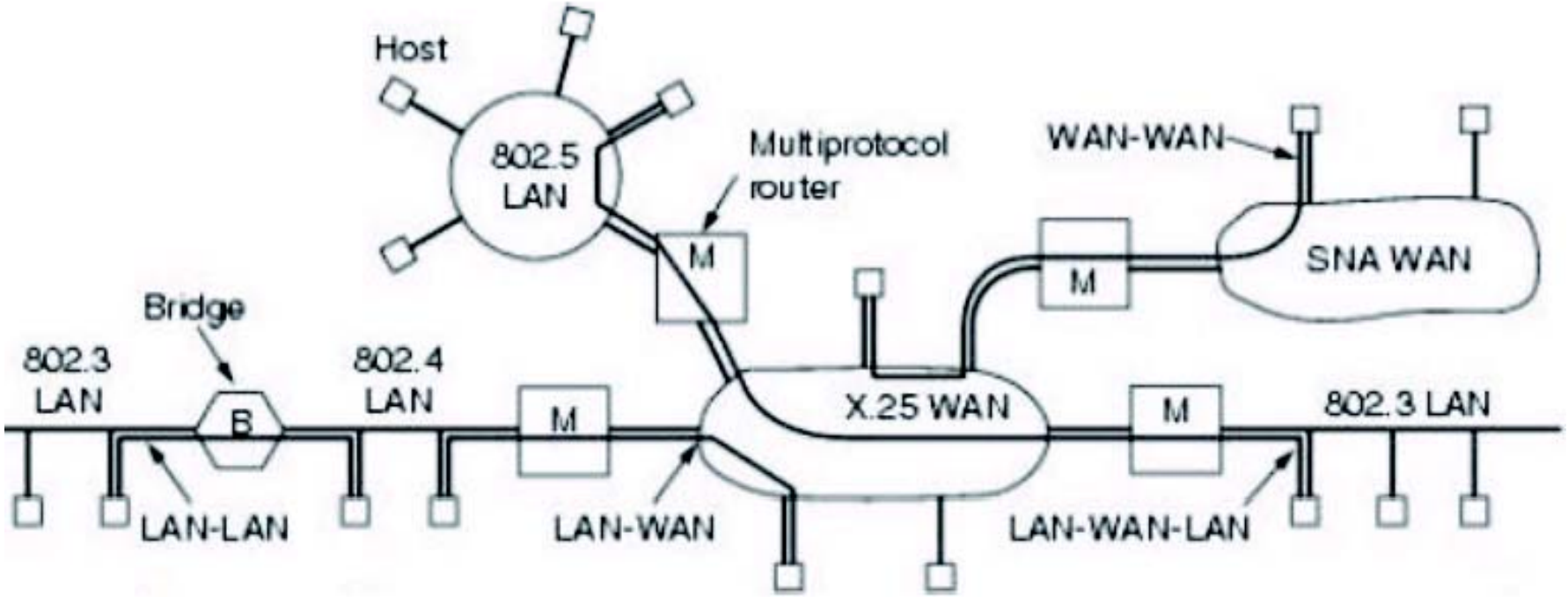
- Différentes implémentations existent qui présentent des points en commun avec ISO8648, exemple



- Le SNDCP et le SNICP se confondent
- Dans le cas du LAN, le service de la sous couche SNDAP se ramène à celui de LLC/MAC alors que sur le réseau X.25 il correspond à celui de X.25-3

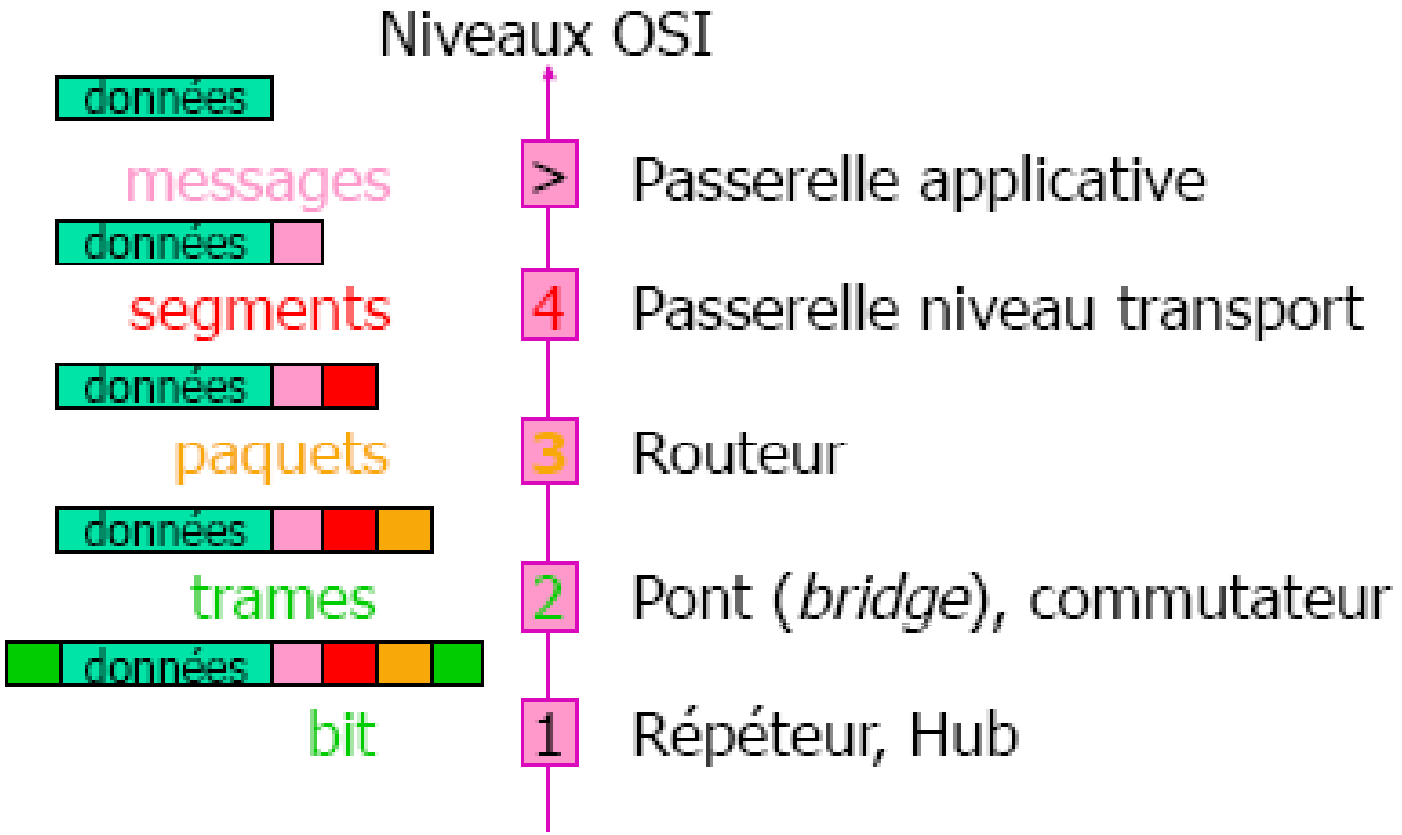
Equipements d'interconnexion

- Doivent prendre en compte l'hétérogénéité des réseaux
  - Donner des solutions aux problèmes d'interconnexions posés
- Type d'interconnexion
  - LAN-LAN, LAN-WAN, WAN-WAN, LAN-WAN-LAN…



Equipements d'interconnexion

- Niveaux d'interconnexions



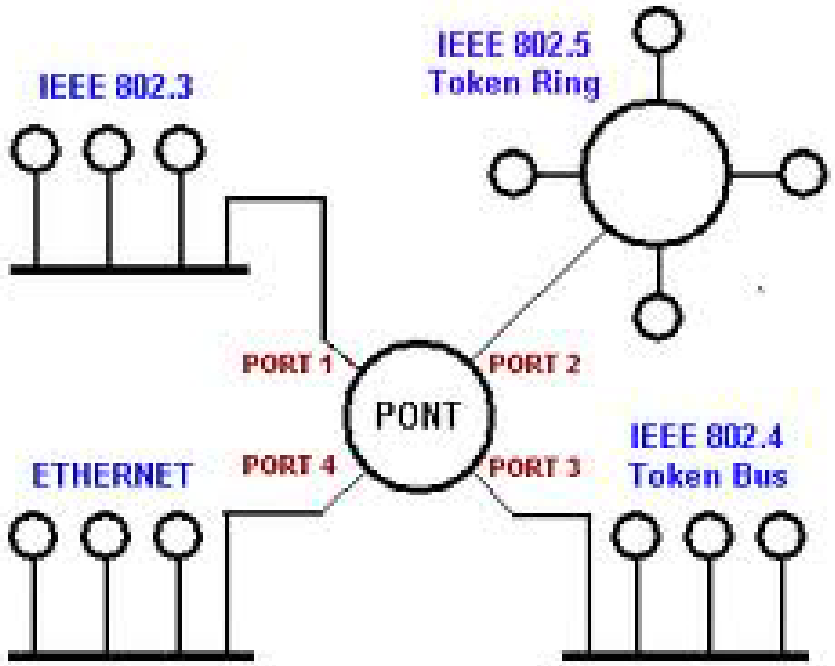
Equipements d'interconnexion

- Répéteur
  - interconnexion de réseaux de même type
  - régénération électrique du signal
  - but : augmenter la distance maximale entre deux stations en reliant deux segments, adapter deux supports différents (coaxial<->fibre optique par ex.)
  - pas d'administration mais ne diminue pas la charge et ne filtre pas les collisions

- Hub ou concentrateur : multi-répéteurs
  - un répéteur sur chaque port
  - toutes les trames sont répétées sur tous les ports

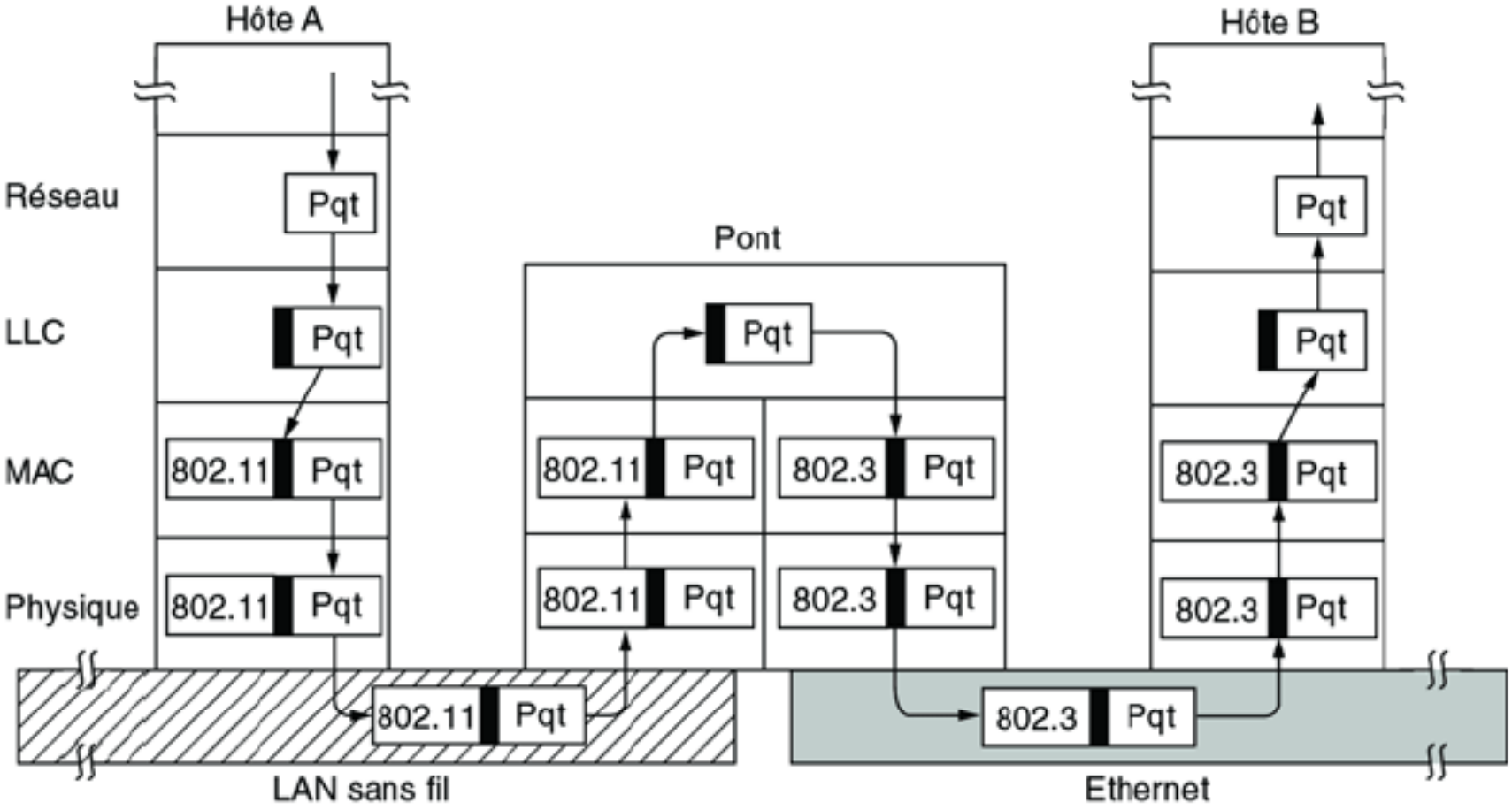
Equipements d'interconnexion

- Pont/commutateur
  - Permet d'interconnecter différents types de LAN
  - Possède autant d'interfaces que de LAN interconnectés : chaque interface contient la sous-couche MAC appropriée



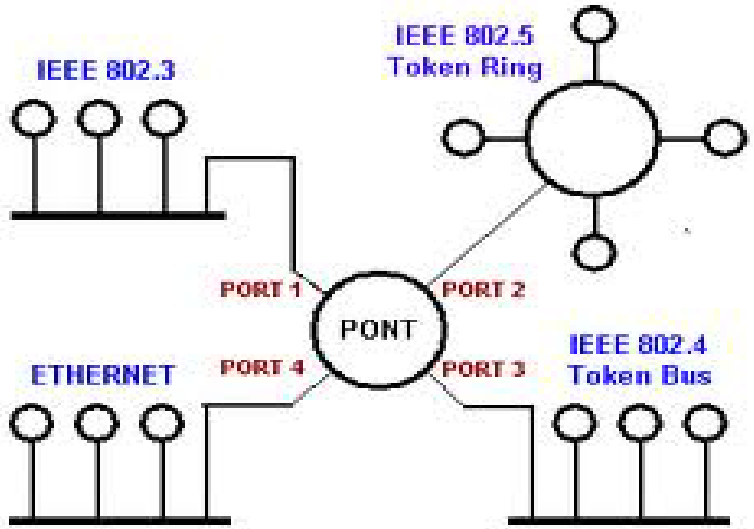
Equipements d'interconnexion

- Pont
  - Exemple d'interconnexion 802.3-802.11



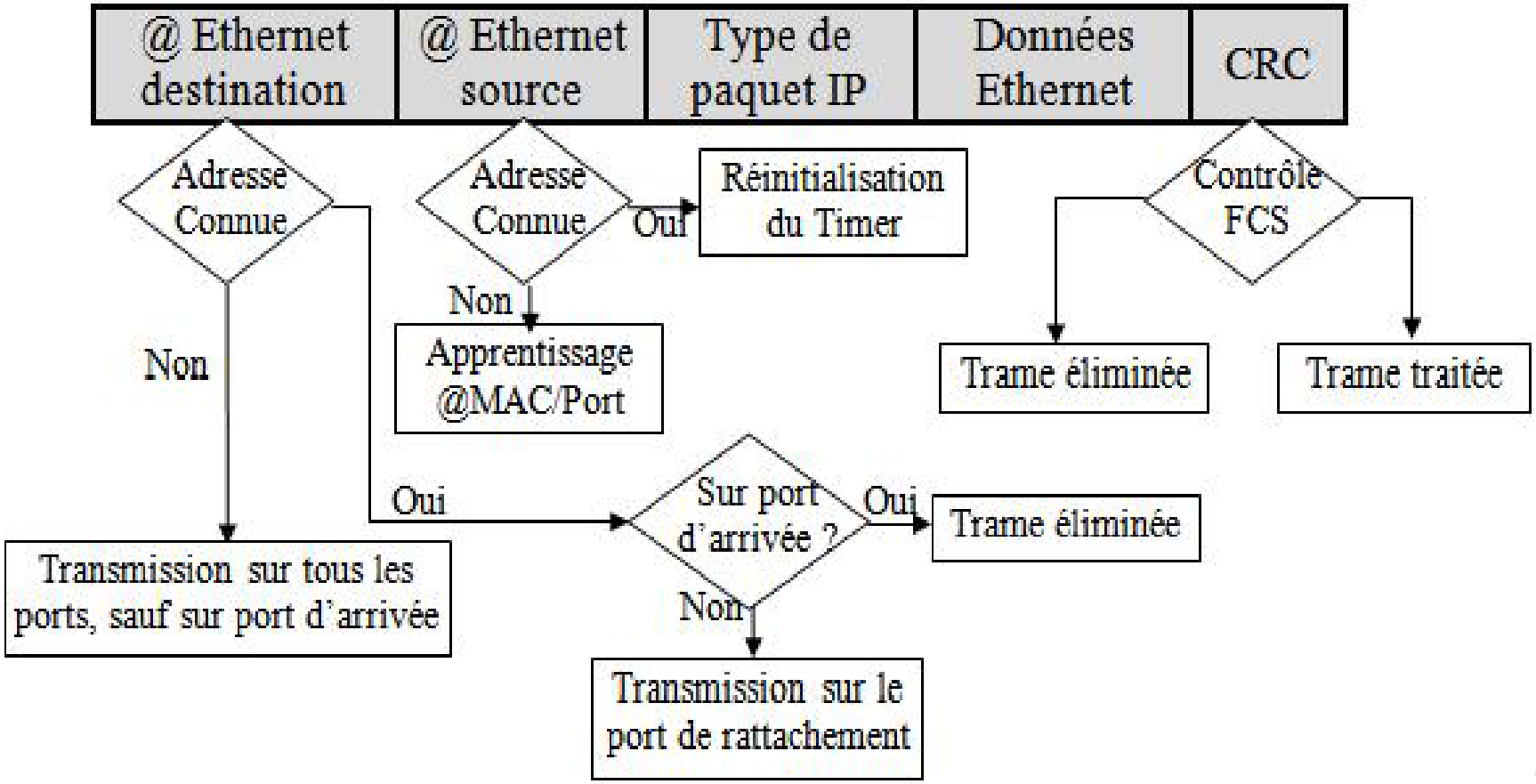
Equipements d'interconnexion

- Pont: fonctionnement
  - Sauvegarde une table de pontage (@MAC, Port de sortie, timer)
  - Algorithme: Quand un pont reçoit une trame
    - LAN destination = LAN source -> rejet de la trame
    - LAN destination <> LAN source -> acheminement
    - LAN destination inconnue, diffusion de la trame sur toutes les lignes sauf celle d'entrée



Equipements d'interconnexion

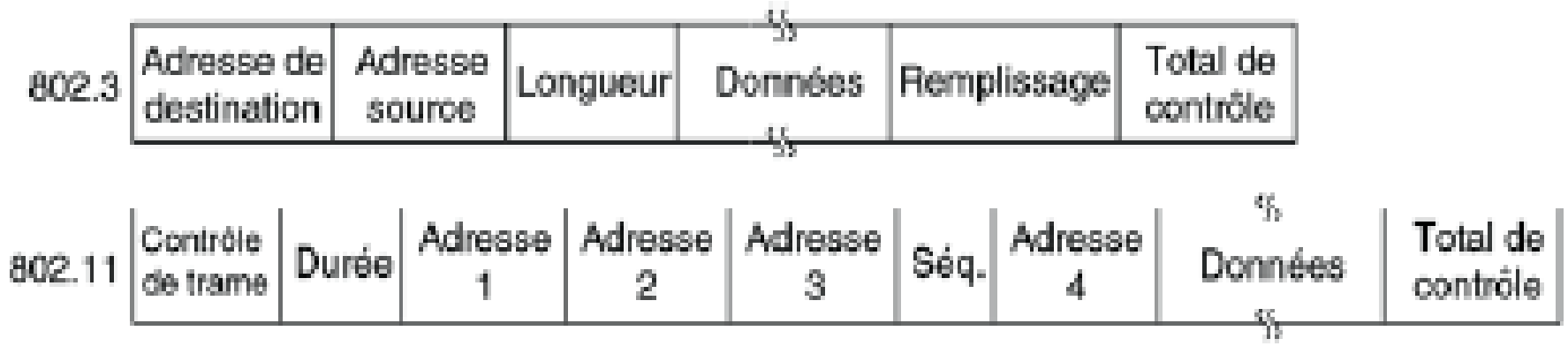
- Pont: fonctionnement
  - Algorithme: (trame Ethernet/802.3)



Equipements d'interconnexion

- Pont: reformatage des trames
  - Le format de l'en-tête des trames diffère d'un LAN à l'autre ➔ Le pont doit reformater les trames
    - consomme du temps CPU
    - Nécessité de recalculer le CRC

    ➔ augmentation des délais



Equipements d'interconnexion

- Pont: types
  - Ponts simples
    - Table de pontage statique (configurée par l'administrateur)
  - Ponts transparents
    - Table construite dynamiquement et maintenue à jour par analyse des trames entrantes
    - Mobilité de stations transparents
  - Ponts à routage par la source
    - la route à suivre est indiquée par la trame elle-même
    - La route (les routes) est déterminé avant l'envoi des données (algorithme source routing))
  - Ponts distants
    - Interconnexion de LAN distants de plusieurs centaines de kilomètres par des liaisons point à point

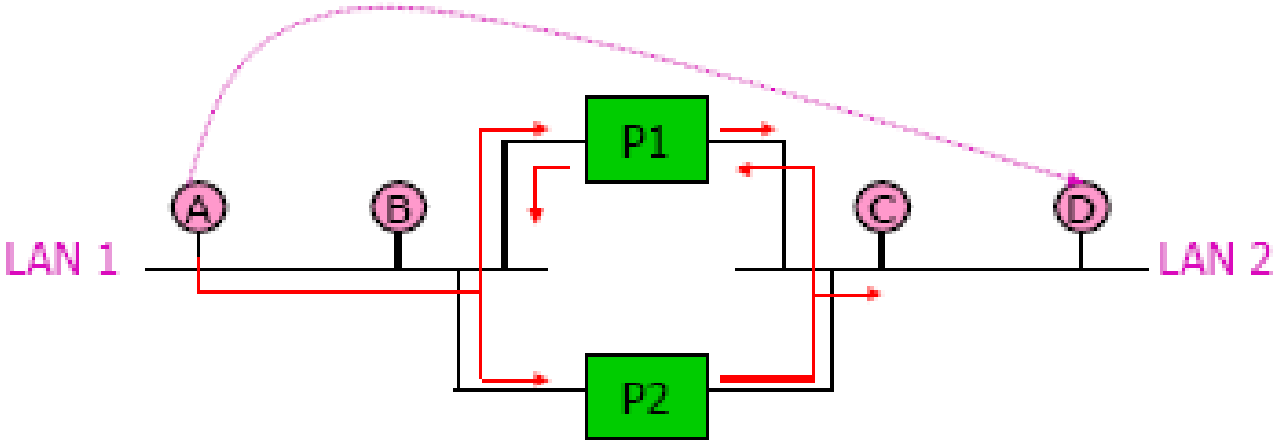
Equipements d'interconnexion

- Pont: problèmes d'interconnexions
  - Problème de stockage des trames
    - Exemples:
      - trames d'un LAN Gigabit Ethernet vers un LAN sans fil 11Mbps
      - Plusieurs LAN émettent simultanément vers un même LAN
  - Problème de fragmentation (MTU)
    - Le LAN destination a une MTU inférieure au LAN source (Exemple (MTU 802.11: 7981 octets, MTU 802.3: 1500 octets)
    - les protocoles de liaison ne font pas de fragmentation

      ➔ donc rejet des trames trop longues
  - Problème de sécurité et qualité de service
    - Chiffrement des données et QoS au niveau 2 dans 802.11 mais pas dans Ethernet

Equipements d'interconnexion

- Pont: redondance
  - Volontaire: pour la tolérance aux pannes et l'efficacité
  - Involontaire (existence de boucles après interconnexion)



A envoie une trame à D qui n'est pas encore localisée par P1 et P2 (D n'a pas encore émis de trame)

- P1 et P2 diffusent la trame sur le LAN2

- P1 capte la trame diffusée par P2 et la rediffuse sur le LAN1 (de même P2 rediffuse celle de P1)

- et ainsi de suite...

Equipements d'interconnexion

- Pont: redondance
  - Av: Permet une tolérance aux pannes et une meilleure efficacité

    Mais
  - Inc: introduit un problème de bouclage
    - duplication des trames
    - oscillations des trames de destination inconnue du fait de l'inondation
  - Solution: STP (spanning tree protocol)
  - Suppression logique des boucle en créant un arbre couvrant
    - mettre en place une topologie logique sans boucle à partir d'un noeud racine
    - nécessite un échange d'information entre ponts: messages BPDU

Equipements d'interconnexion

- Routeur
  - Permet d'interconnecter différents sous-réseaux (réseaux)
  - Rôle:
    - Principal: routage ➔ Acheminement des paquets vers le bon lien de sortie en fonction de l'adresse destination (niveau 3)
    - Secondaires : Filtrage, qualité de service…
  - Routeurs multi-protocoles
    - Capable de router plusieurs protocoles de niveau 3
    - Conversion du format des paquets entre deux sous réseaux de natures différentes
- Bridge-Routeur
  - fonctionnalités des ponts et routeurs dans un même chassis : si protocole routable (IP), table de routage sinon (Netbios) table d'acheminement

Equipements d'interconnexion

- Passerelle
  - élément d'interconnexion niveau transport ou application
  - Exemple de passerelle niveau transport :
    - lien entre une connexion TCP et une connexion SNA
  - Exemple de passerelle applicative :
    - Courrier électronique -> SMS

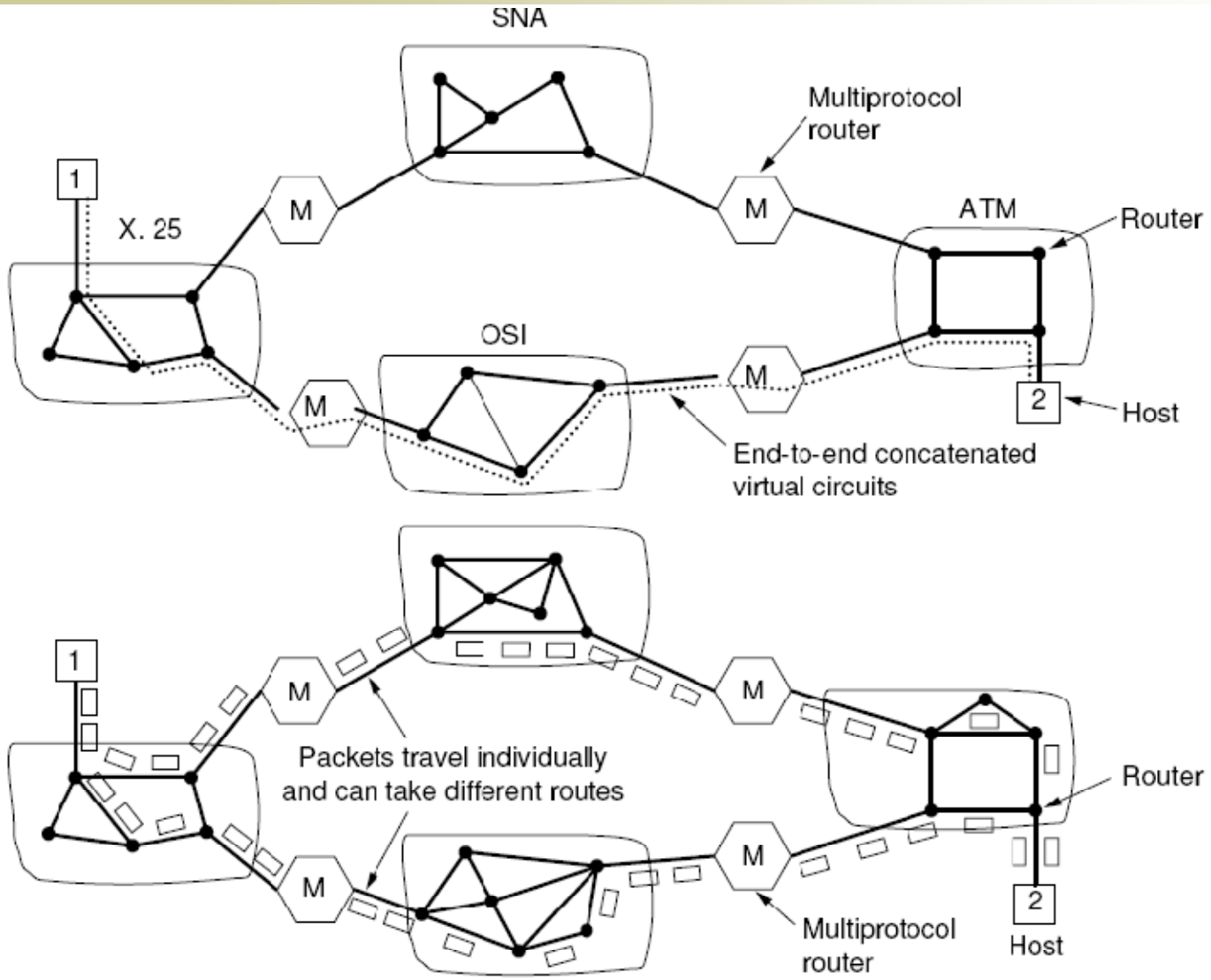
hétérogéniété des services réseaux

mode connecté

et

mode non connecté

Type du service



pictures from TANENBAUM A. S. *Computer Networks 3rd edition*

Type du service

- Service orienté connexion
  - Une connexion de niveau réseau s'appelle circuit virtuel
  - Le chemin associé au circuit virtuel dans le réseau est alloué à l'établissement de la connexion. La décision de routage n'est prise qu'au cours de la phase d'établissement de la connexion.
  - les paquets contiennent seulement le numéro de circuit utilisé
  - Tous les paquets circulant sur le même circuit virtuel empruntent le même chemin.
  - Exemple : protocole ATM (Asynchronous Transfer Mode), X25

- Service sans connexion (Unité de donnée: *datagramme*)
  - Chaque paquet est envoyé indépendamment des autres et routé séparément.
  - Des paquets successifs peuvent donc suivre des routes différentes
  - chaque paquet doit contenir l'@ destination
  - Exemple: le protocole IP (Internet Protocol)

Avantages / inconvénients des services

- Mode Connexion
  - ☺ Pas de déséquencement
  - ☺ Ressources réservées au départ
    - ☺ Garantie de qualité de services facile
    - ☺ Pas de problème de congestion ultérieure
  - ☹ Ressources réservées inutilement
  - ☹ Temps d'acheminement plus long (temps d'établissement de la connexion au départ) -> problème pour le temps réel
  - ☹ Délicat en cas de défaillance d'un routeur

Avantages / inconvénients des services

- Mode Sans Connexion
  - ☺ Temps d'acheminement plus rapide
  - ☺ Défaillance d'un routeur → pertes seulement des paquets, adaptation rapide
  - ☺ Pas de ressources réservées inutilement
  - ☹ Qualité de services difficile à garantir
  - ☹ Congestion résolue difficilement
  - ☹ Problème de IP actuel
  - ☹ Calcul du routage à chaque paquet

ISSUE	DATAGRAM SUBNET	VC SUBNET
Circuit setup	Not possible	Required
Addressing	Each packet contains the full source end destination address	Each packet contains a short vc number
State information	Subnet does not hold state information	Each established vc requires subnet table space
Routing	Each packet is routed independently	Route chosen when vc is set up; all packets follow this route
Effect of node failure	None, except for packets lost during the crash	All vcs that passed through the failed equipment are terminated
Congestion control	Difficult	Easy if enough buffers can be allocated in advance for each vc set up
Complexity	In the transport layer	In the network layer
Suited for	Connection-oriented and connectionless service	Connection-oriented service

Primitives de Service – Mode Connecté

- Un service est défini par un ensemble de primitives (ou opérations) disponible pour un utilisateur ou une entité pour y accéder
- Il y a 4 classes de service
  - REQUEST
    - Une entité sollicite un service (ou demande une connexion)
  - INDICATION
    - Une entité est informée d'un événement (le récepteur reçoit une demande de connexion)
  - RESPONSE
    - Une entité répond à un événement (le récepteur envoie l'autorisation de la connexion)
  - CONFIRM
    - Une entité accuse la réception de la réponse à sa demande (l'émetteur reçoit une confirmation de la connexion)

Primitives de Service – Mode Connecté
(ISO 8348)

- Primitives d'établissement d'un circuit virtuel
  - N_CONNECT.request (dest, source, conf, tel, qos, d_utilisateur)
  - N_CONNECT.indication (dest, source, conf, tel, qos, d_util)
  - N_CONNECT.response (répondeur, conf, tel, qos, d_utilisateur)
  - N_CONNECT.confirm (dest, source, conf, tel, qos, d_util)

- Primitives de rupture de circuit virtuel
  - N_DISCONNECT.request (origine, raison, d_utilisateur, adr_en_rep)
  - N_DISCONNECT.indication (origine, raison, d_utilisateur, adr_en_rep)

- Primitives d'échange sur circuit virtuel
  - N_DATA.request (données)
  - N_DATA.indication (données)
  - N_DATA_ACKNOLEDGE.request ()
  - N_DATA_ACKNOLEDGE.indication ()

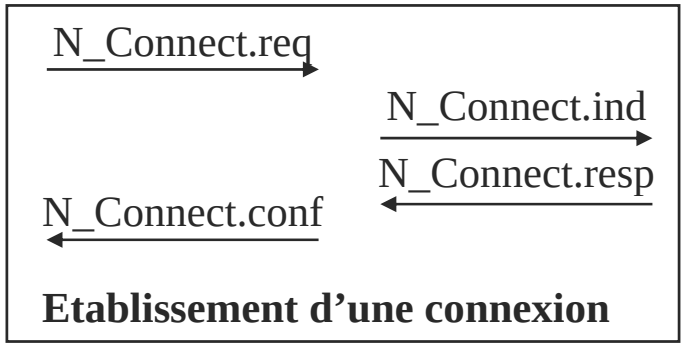
Primitives de Service – Mode Connecté
(ISO 8348)

- Envoi de données exprès sur un circuit virtuel
  - N_EXPEDITED_DATA.request (données)
  - N_EXPEDITED_DATA.indication (données)

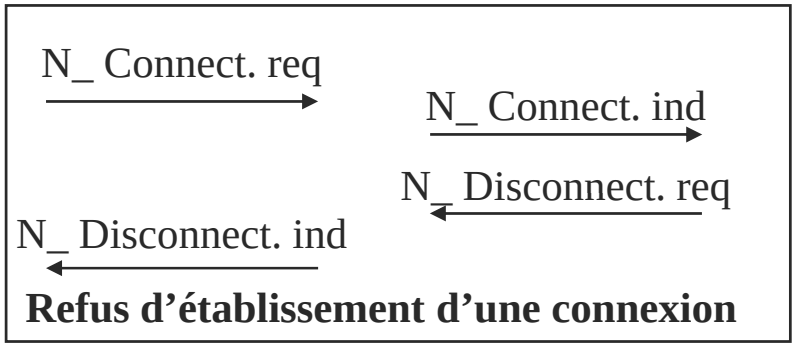
- Commandes de contrôle d'un circuit virtuel
  - N_RESET.request (origine, raison)
  - N_RESET.indication (origine, raison)
  - N_RESET.response ();
  - N_RESET.confirmation ();

Primitives de Service – Mode Connecté
(ISO 8348)

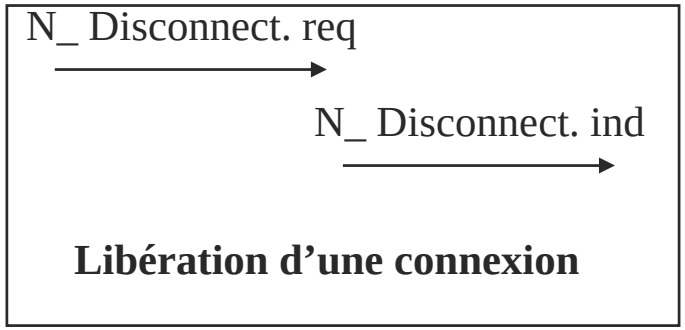
Quelques enchaînements de primitives



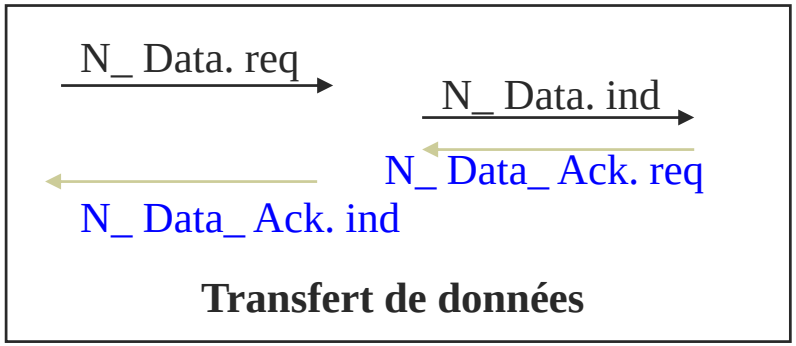
Etablissement d'une connexion



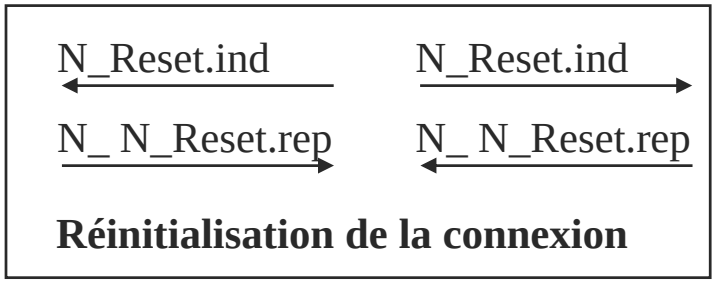
Refus d'établissement d'une connexion



Libération d'une connexion



Transfert de données



Réinitialisation de la connexion

Primitives du service orienté connexion

- Établissement de connexion ou circuit virtuel

PRIMITIVES	PARAMETRES
N_CONNECT.request	adresse source, adresse destination, confirmation réception, données exprès, qos, données utilisateur
N_CONNECT.indication	adresse source, adresse destination, confirmation réception, données exprès, qos, données utilisateur
N_CONNECT.response	adresse en réponse, confirmation réception, données exprès, qos, données utilisateur
N_CONNECT.confirmation	adresse en réponse, confirmation réception, données exprès, qos, données utilisateur

  - *« **Confirmation réception** » :* si vrai, demande l'acquittement des paquets de données transmis par la suite
  - *« **Données exprès** » :* si vrai, autorise l'envoi de paquets de données exprès transmis en priorité en dehors de tout contrôle de flux (**ex** : permet l'interruption prioritaire de programmes lancés à distance)
  - *« **Quality of Service** » : deux listes de valeurs définissant les qualités souhaitée et acceptable par l'appelant (**ex** : débit, délai de transfert, taux d'erreurs, etc.)*
  - *« **Données utilisateurs** » : faible volume de données transmis lors de la demande de connexion pour réduire les délais (**ex** : numéro de carte bancaire)*

Primitives du service orienté connexion

- Libération de connexion ou circuit virtuel

N_DISCONNECT.request	origine, raison, données utilisateur
N_DISCONNECT.indication	origine, raison, données utilisateur

  - *Sans adresses source et destinataire car circuit virtuel*
  - *Permet d'indiquer les raisons de la demande de déconnexion*

- Signalisation de défaillances du réseau de transport

N_RESET.request	origine, raison
N_RESET.indication	origine, raison
N_RESET.response	origine, raison
N_RESET.confirmation	origine, raison

  - *Permet à un équipement intermédiaire d'indiquer qu'il est complètement congestionné*

Primitives du service orienté connexion

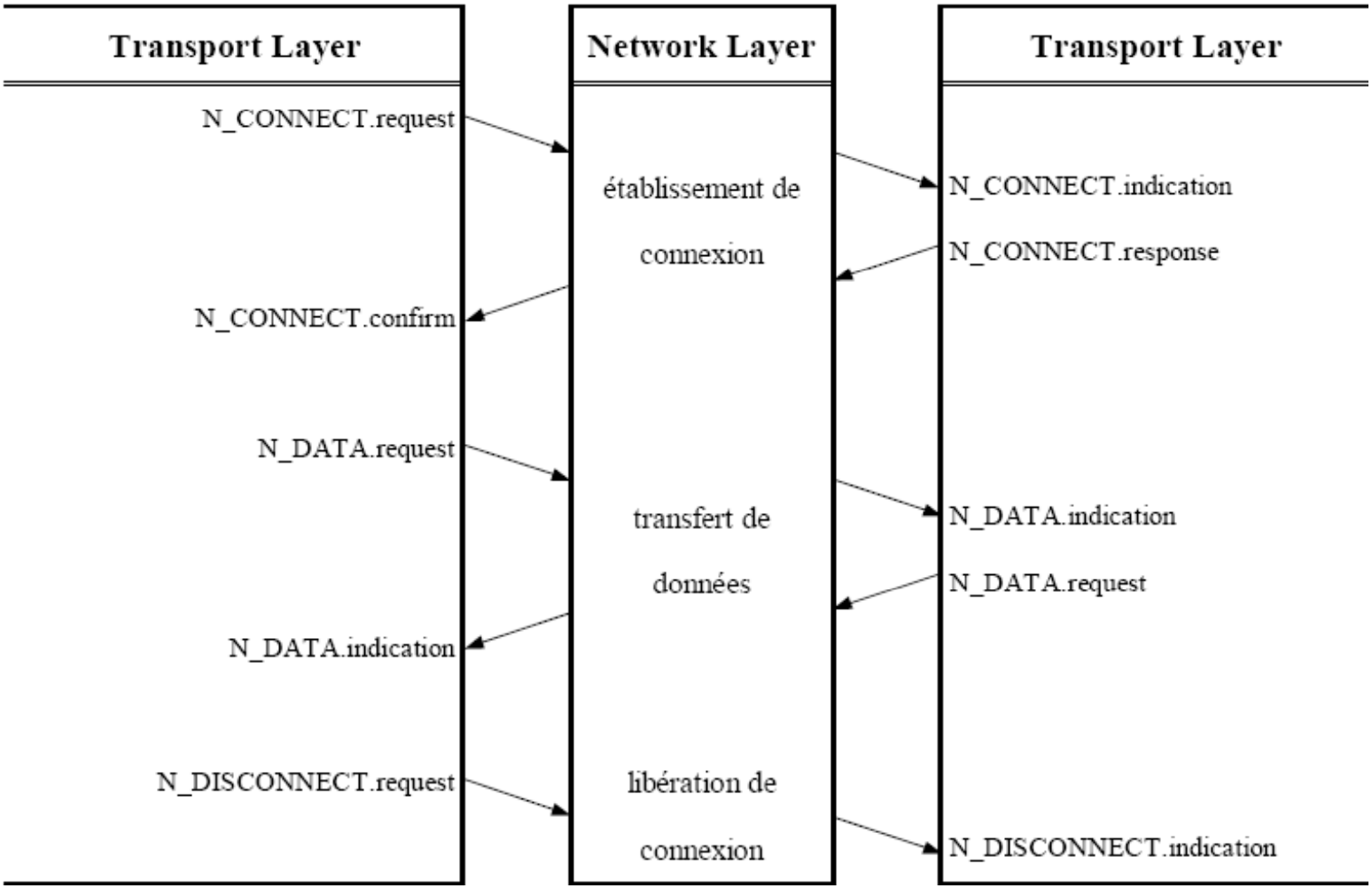
- Primitives du service orienté connexion
  - Transfert de données sur un circuit virtuel

N_DATA.request	données utilisateur
N_DATA.indication	données utilisateur
N_DATA_ACKNOWLEDGE.request	
N_DATA_ACKNOWLEDGE.indication	
N_EXPEDITED_DATA.request	données utilisateur
N_EXPEDITED_DATA.indication	données utilisateur

  - *« **N_DATA** » : sans adresse source et destination car transmission sur circuit virtuel uniquement*
  - *« **N_DATA_ACKNOWLEDGE** » : permet l'acquittement de données*
    - Sans numéro d'ordre à l'émission ou à la réception
    - Permet uniquement de vérifier que le bon nombre de paquets ont été reçues correctement
  - *« **N_EXPEDITED_DATA** » : permet l'envoi de données exprès*

Primitives du service orienté connexion

- Mise en oeuvre du service orienté connexion



Primitives du service orienté connexion

- Correspondance avec services réseaux

Primitives OSI	Paquets X.25
N_CONNECT.request	Envoie d'un **Call request**
N_CONNECT.indication	Arrivée d'un **Incoming call**
N_CONNECT.response	Envoie d'un **Call accepted**
N_CONNECT.confirmation	Arrivée d'un **Call connected**
N_DATA.request	Envoie d'un **Data**
N_DATA.indication	Arrivée d'un **Data**
N_RESET.request	Envoie d'un **Reset request (ou Restart request)**
N_RESET.indication	Arrivée d'un **Reset request (ou Restart request)**
N_RESET.response	Envoie d'un **Reset confirmation (ou Restart confirmation)**
N_RESET.confirmation	Arrivée d'un **Reset confirmation (ou Restart confirmation)**

Primitives du service orienté connexion

- Correspondance avec services réseaux (suite)

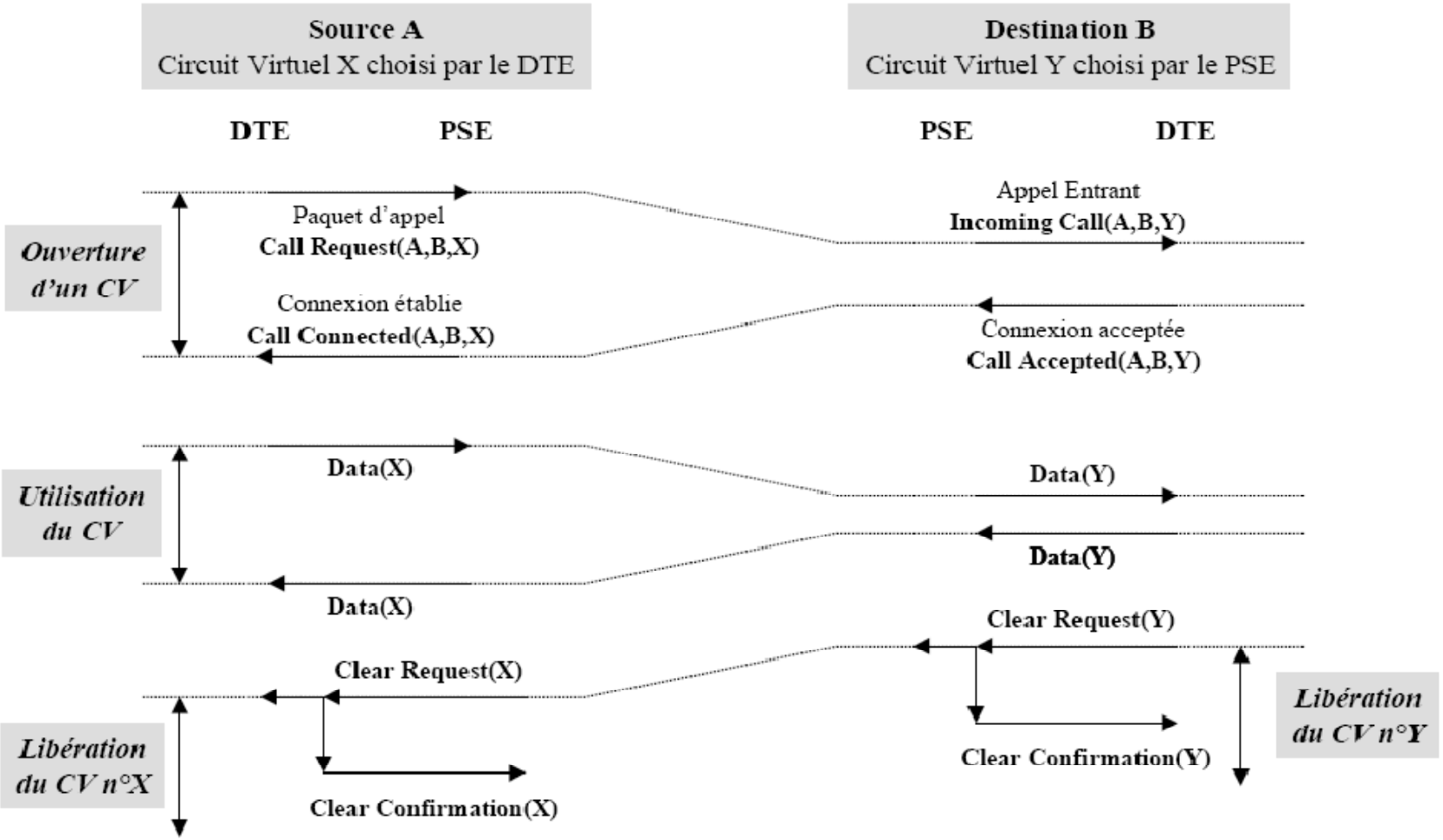
Primitives OSI	Paquets X.25
N_REPORT.indication	Envoie d'un **Diagnostic**
N_DATA_ACKNOWLEDGE.request	Envoie d'un **RR, RNR ou REJ**
N_DATA_ACKNOWLEDGE.indication	Arrivée d'un **RR, RNR ou REJ**
N_EXPEDITED_DATA.request	Envoie d'un **Interrupt request**
N_EXPEDITED_DATA.indication	Arrivée d'un **Interrupt confirmation**
N_DISCONNECT.request	Envoie d'un **Clear request**
N_DISCONNECT.indication	Arrivée d'un **Clear request**
N_DISCONNECT.response	Envoie d'un **Clear confirmation**
N_DISCONNECT.confirmation	Arrivée d'un **Clear confirmation**

Primitives du service orienté connexion

- Établissement de connexion

DTE (Data Terminal Equipment): ES (End System)

PSE (Packet Switching Equipment): IS (Intermediate System)



Primitives du service sans connexion

- Primitives du service sans connexion
  - Trois types de primitives

PRIMITIVES	PARAMETRES
N_UNIDATA.request	adresse source, adresse destination, qos, données utilisateur
N_UNIDATA.indication	adresse source, adresse destination, qos, données utilisateur
N_FACILITY.request	qos
N_FACILITY.indication	adresse destination, qos, raison
N_REPORT.indication	adresse destination, qos, raison

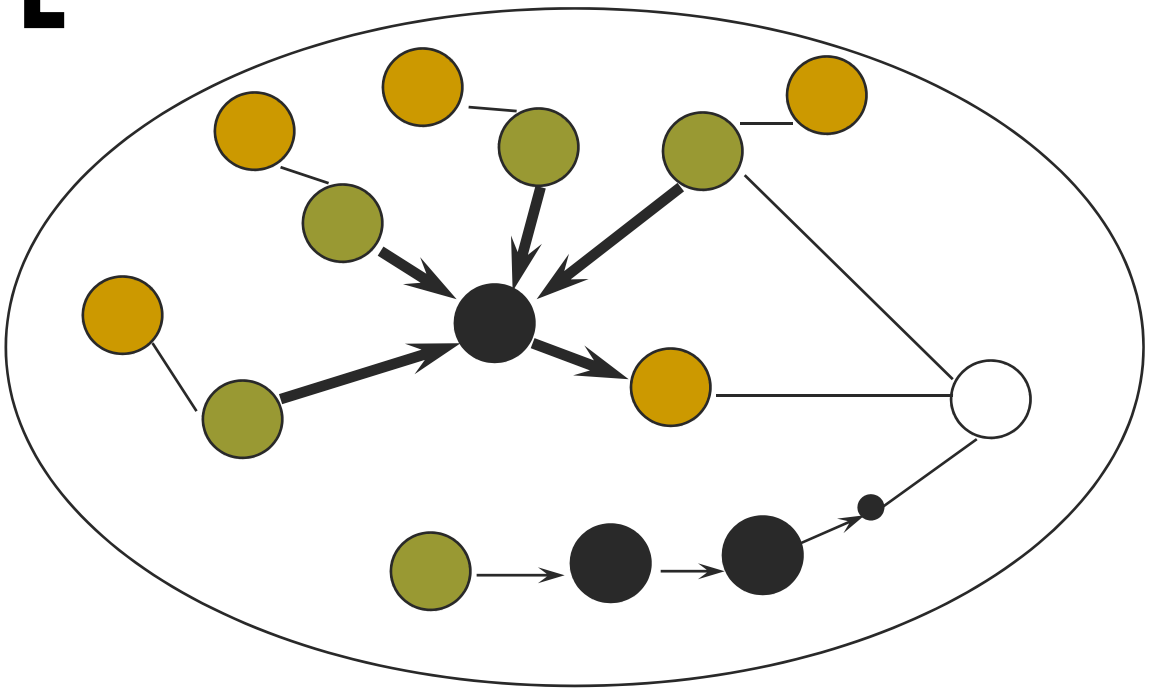
  - *« **N_UNIDATA** » :* avec adresse source et destination car pas de circuit virtuel
  - *« **N_FACILITY** » :* pour savoir si une qualité de service souhaitée peut être obtenue
  - *« **N_REPORT** » :* permet de signaler des problèmes au niveau du réseau de transport

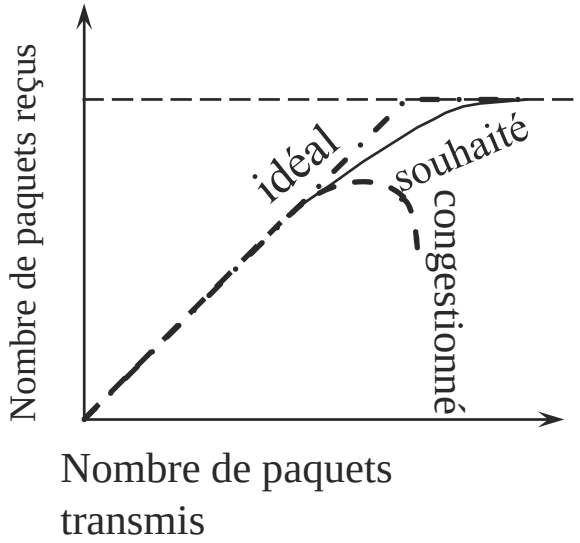
Primitives de Service – Mode Non Connecté
(ISO 8348)

- Primitives d'échange d'informations
  - N_UNIDATA.request (source, destination, qos, données)
  - N_UNIDATA.indication (source, destination, qos, données)
- Primitives de contrôle
  - N_FACILITY.request(qos)
  - N_FACILITY.indication(destination, qos)
  - N_REPORT.indication(destination, qos, raison)
- Désignation des N_SAP (longueur <= 20 octets)
  - Authority and Format Identifier (AFI):
    - qualifie le type de l'adresse contenue dans le dernier champ
  - Initial Domain Identifier (IDI) :
    - précise le domaine auquel appartient le dernier champ
  - DSP
    - Adresse proprement dite

Interconnexion de réseaux et contrôle de la congestion

Congestion





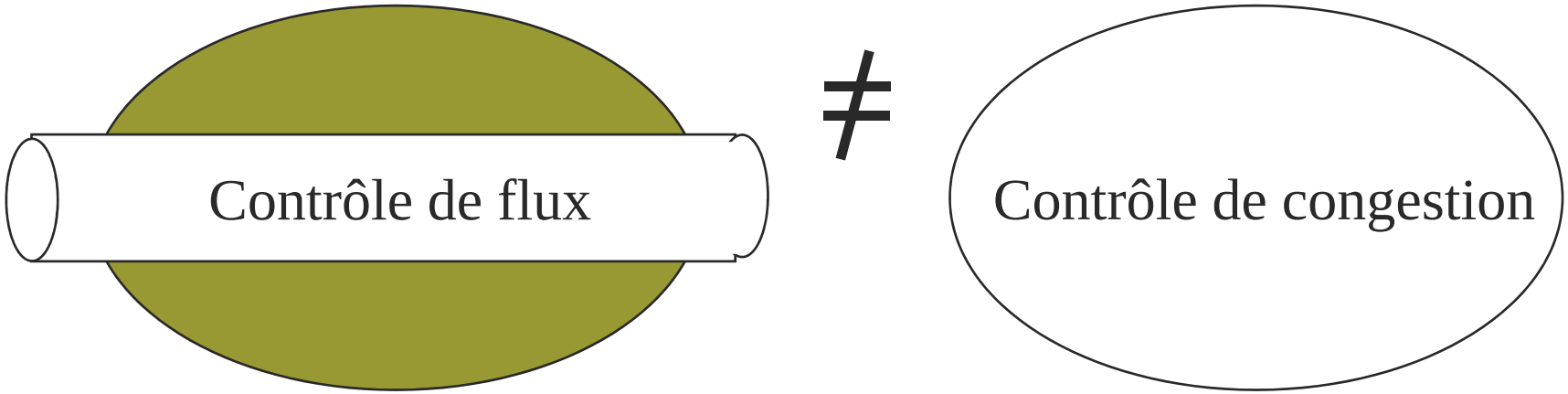
- congestion +

- Congestion: trop de paquets présents dans le réseau ⇨ dégradation de performances
- Un nœud congestionné finit par rejeter les paquets
- Augmenter la taille des mémoires tampons n'évite pas le rejet et accentue la congestion

Congestion

- Causes de la congestion
  - Taille insuffisante des mémoires tampons des différents routeurs
    - débordement des tampons de files d'attente
  - Capacité des liens du réseau trop faible pour la charge donnée
    - Trafic trop important en entrée par rapport aux capacités des lignes en sortie
  - Performance CPU des routeurs
    - processeurs trop lents dans les routeurs
    - Différence de puissance de traitement d'un routeur à l'autre
  - Rafales: émissions irrégulières des sources

Contrôle de congestion



Contrôle de congestion = assurer que le sous-réseau est capable de transporter le trafic présent

≠

Contrôle de flux = assurer le trafic point à point entre un émetteur et un récepteur (i.e. assurer que l'émetteur ne soit pas trop rapide vis à vis du récepteur)

Contrôle de congestion

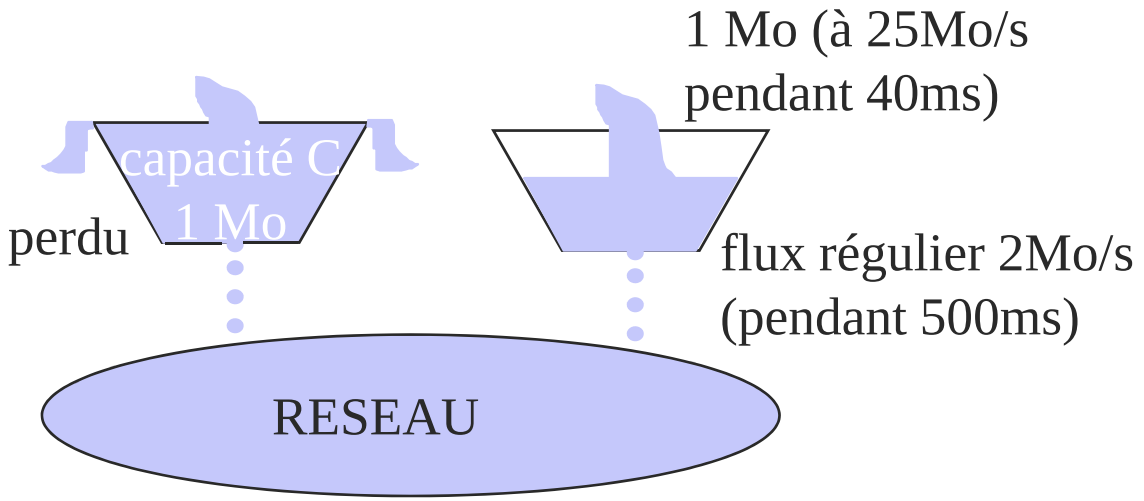
- Deux approches de contrôle de congestion
  - Algorithmes en boucle ouverte
    - concevoir un système qui évite, au mieux, les problèmes de congestion (prévention)
  - Algorithmes en boucle fermée
    - prévoir des mécanismes pour la détection de la congestion, la rétroaction et l'ajustement du trafic (guérison)
- Concevoir des contrôles à différents niveaux
  - liaison : réduire le nombre de trames échangées / retransmises, calibrer le contrôle de flux ...
  - réseau : fixer la politique de partage des mémoires tampons, d'ordonnancement, de destruction et de routage des paquets ...
  - transport : idem que la couche liaison, le choix de la valeur des temporisateurs est plus difficile (évaluation dynamique)!

Canalisation du trafic «traffic shaping»

- Algorithme du sceau percé
  - Chaque ordinateur est relié au réseau via une interface d'accès
  - Cette interface simule le seau percé à l'aide d'une file d'attente de taille fixe
  - Si un paquet arrive dans la file et qu'elle est pleine, il est détruit
- Gestion du seau percé
  - À chaque top d'horloge, un paquet de la file d'attente est envoyé sur le réseau, sauf si celle-ci est vide
  - Tout paquet sortant est placé dans la file d'attente, sauf si celle-ci est pleine
  - Ce mécanisme transforme un flux irrégulier de paquets provenant d'un processus interne à un ordinateur source en un flux régulier de paquets sur le réseau

Canalisation du trafic «traffic shaping»

- Les techniques de canalisation du trafic ont pour but de
  - réguler la vitesse d'écoulement des données
  - Maintenir le trafic le plus constant possible
- Algorithme du seau percé «leaky bucket algorithm»



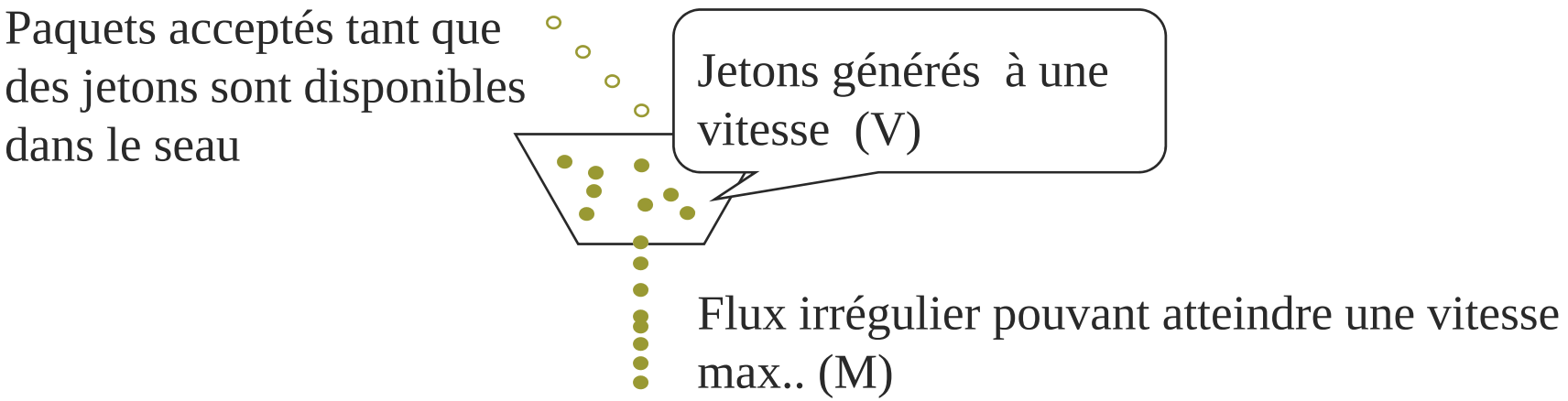
- seau : file d'attente de taille fixe
- eau : paquets ou multiple d'octets
- vitesse : paquet/sec ou octet/sec.

Canalisation du trafic (suite)

- Algorithme du seau percé à jetons
  - « Token leaky bucket algorithm »
  - Extension de l'algorithme précédent
  - Algorithme plus souple car il permet une augmentation provisoire du trafic
- Principe
  - Un jeton est engendré à chaque top d'une horloge (nombre maximum *n* de jetons)
  - Un paquet est transmis s'il reste au moins un jeton
    - Sinon, il est rejeté
  - Un jeton est détruit à chaque émission de paquet

Canalisation du trafic (suite)

- Algorithme du seau percé à jetons



  - Il est aussi possible de canaliser le trafic entre routeurs
  - Les algorithmes du seau percé n'éliminent pas la possibilité qu'un paquet soit détruit dans le réseau

Canalisation du trafic (suite)

- Exemple
  - S = burst length (seconds): durée d'une rafale
  - ρ = token arrival rate (bytes per sec): taux de génération des jetons
  - C = bucket size (bytes): Capacité
  - M = maximum output rate (bytes per sec): Capacité du lien

(On génère des jetons à intervalles *Δt*, jusqu'à une capacité *C*. L'algorithme permet d'émettre des rafales de durée *S*, sur un lien de capacité *M)*

$$C + \rho S = M S$$

  - the bucket size 250 kB (tokens)
  - the tokens are generated at the rate of 2 MB/second
  - maximum output rate is 25 MB/sec
  - the bucket is full, a 1 MB burst arrives
  - the bucket can leak at the full 25 MB/sec for about 11 ms

S = C / (M - ρ) = 250 kB / (25 MB/sec - 2 MB/sec) = **11 ms**

Canalisation du trafic (suite)

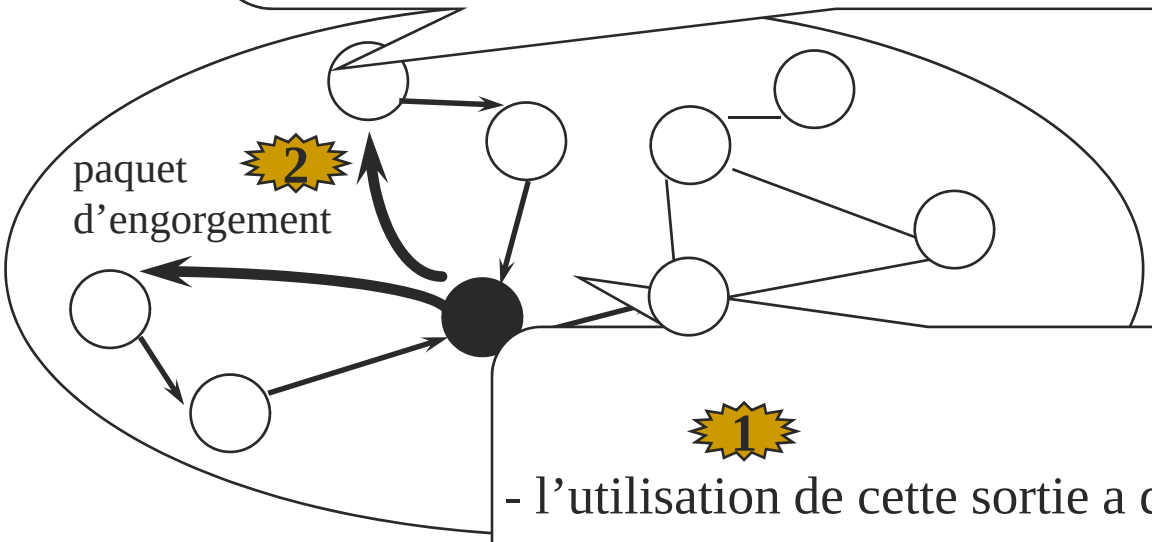
- Appropriée pour les réseaux de type CV : négocier un accord (contrat de service) sur la nature du trafic à l'établissement d'un CV. Exemple de spécification de flux [Partridge 92] :
  - paramètres de trafic : taille max. d'un paquet, capacité du seau à jetons, vitesse de remplissage du seau, vitesse max. de transmission
  - paramètres de service : taux acceptable de perte des données, taille tolérée de données perdues consécutivement, retard toléré avant envoi, gigue (variation du délai d'acheminement), garantie d'objectif de qualité

Technique des paquets d'engorgement

- Cette méthode peut être utilisée pour tous les types de réseaux
- Chaque routeur va surveiller ses lignes de sortie en quantifiant leur utilisation
- En cas d'alerte, des paquets d'engorgement sont envoyés
- Principe
  - Utilisation récente de la ligne :
    - $u_{nouveau} = a*u_{ancien} + (1-a)f$
    - f : échantillonnage instantané de la ligne
    - a : coefficient compris entre 0 et 1
  - Quand $u_{nouveau}$ dépasse un seuil, un paquet d'engorgement est envoyé à tout ordinateur source concerné qui réduit alors ses envois

Technique des paquets d'engorgement

- réduit le trafic (50%) vers la destination 3
- ignore les paquets d'engorgement pendant une certaine durée
- si au bout d'un certain délai, aucun paquet d'engorgement n'est reçu, le trafic est augmenté par petit incrément



Rétro-contrôle

1
- l'utilisation de cette sortie a dépassé un seuil
- à la sortie, le paquet est marqué

Tech. des paquets d'engorgement (suite)

- Critique 1 :
  - si la source ne collabore pas et ne réduit pas son trafic, elle peut ainsi profiter de la situation.
- ➔ Solution: algorithme du temps équitable
  - pour chaque sortie, les paquets sont envoyés de façon cycliques selon la source "Weighted Fair Queuing"

- Critique 2 :
  - la taille des paquets est variable.
- ➔ Solution - algorithme du temps équitable pondéré –
  - appliquer le même algorithme par octet

Tech. des paquets d'engorgement (suite)

- Critique 3 :
  - lorsque le débit est important (155Mb/s) ainsi que le temps de transit (30ms), une grande quantité de données aura été injectée dans le réseau (4,5 Mb) avant que le paquet d'engorgement n'arrive à la source.

➔ Solution - contrôle de l'engorgement en pas à pas [Mishra & Kanakïa 92] –

- en remontant pas à pas vers la source et sur chaque routeur intermédiaire, le paquet d'engorgement a pour effet de réduire la vitesse vers la destination. Chaque routeur a ainsi besoin de réserver des mémoires tampon supplémentaires pour le trafic vers la destination. Le noeud de congestion est ainsi rapidement soulagé (« Hop by Hop chock packet").

Le délestage «load shedding»

- Pour les méthodes de contrôle de congestion décrites précédemment, le risque de congestion n'est pas forcément écarté, le délestage consiste alors à rejeter des paquets.
- Choix des paquets à rejeter
  - parfois il vaut mieux rejeter les paquets les plus récents (cas du protocole Go-Back-N). Pour certaines applications (temps réel) c'est l'inverse.
  - perdre une ligne de pixels d'une image est moins grave qu'un texte associé ...

Le délestage «load shedding» (suite)

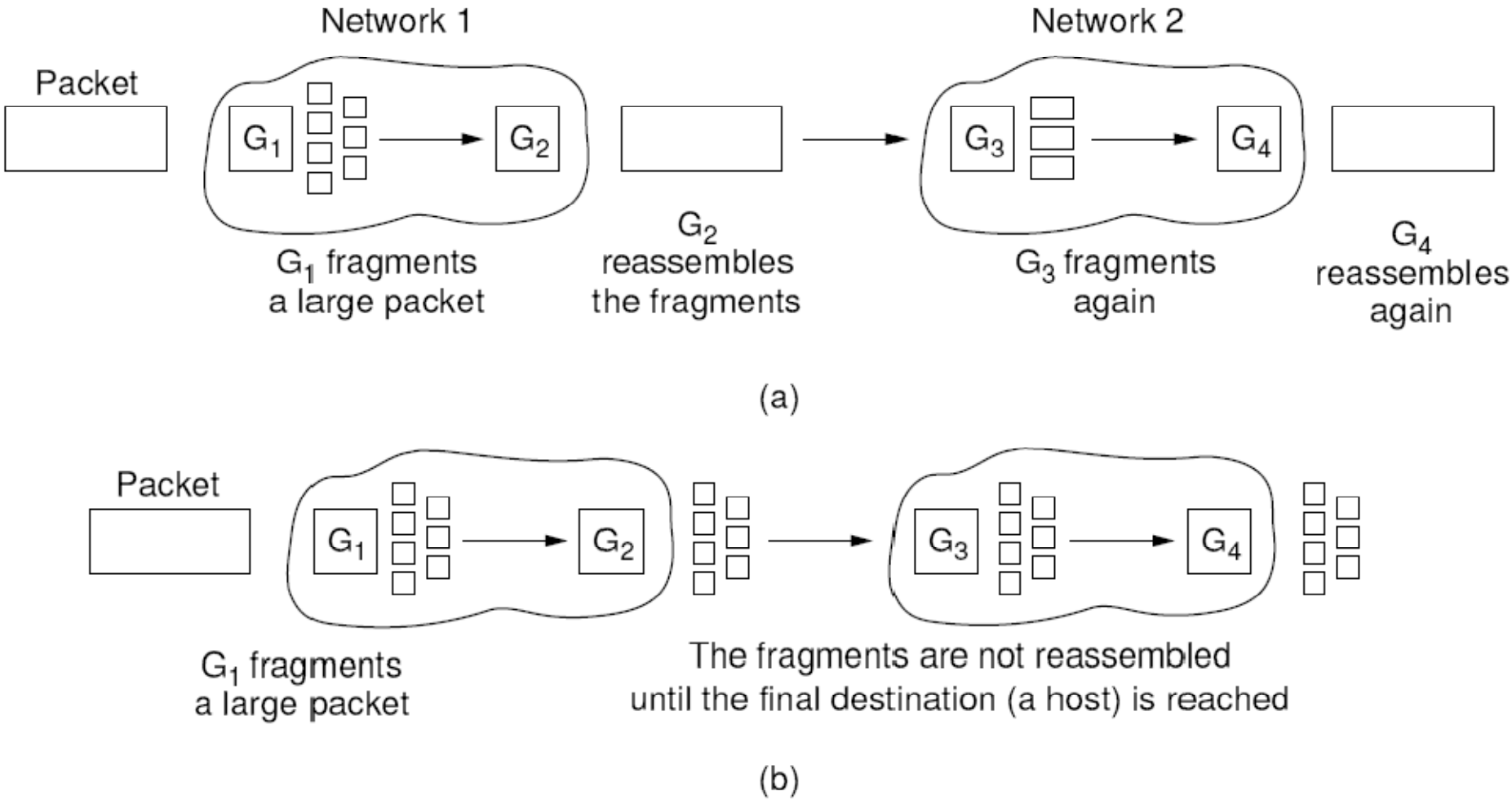
- Solutions
  - les applications marquent les paquets suivant une certaine classification de priorité
  - autoriser le dépassement des limites négociées en marquant par une faible priorité le trafic en excès
  - détruire tous les fragments / cellules d'un même paquet ...

Le contrôle de la gigue «jitter»

- Objectif : rendre le temps de transit sur le réseau assez constant dans la limite d'un certain intervalle
- Un routeur peut retarder / accélérer l'envoi d'un paquet

Interconnexion de réseaux et fragmentation /réassemblage

Fragmentation / Réassemblage



pictures from TANENBAUM A. S. *Computer Networks 3rd edition*