

# Chapitre 7 : Le niveau Transport

/home/kouna/d01/adp/bcousin/REPR/Cours/7.fm - 1 Mars 1998 15:24

## Plan

- Introduction
- Les mécanismes généraux :
  - . QoS, transferts de données spécifiques, gestion de la connexion, adressage, sécurité
- Le service
- Le protocole TP :
  - . Classes, les TPDU, les champs, un scénario
- Conclusion

## Bibliographie

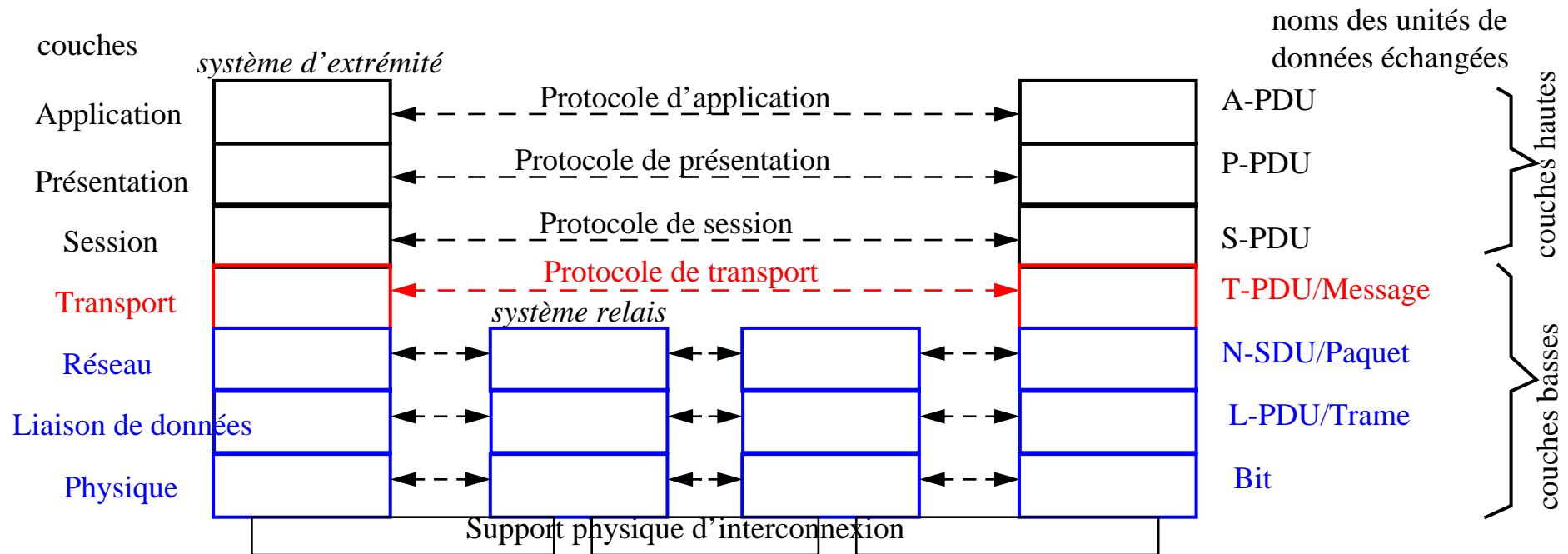
- Service et protocole Transport, ISO 8072 et 8073 (CCITT X.214 et X.224).
- D.Dromard &al., Réseaux informatiques : cours et ex., Eyrolles, tome 2, 1994. Chapitre 2.
- G.Pujolle, Les réseaux, Eyrolles, 1995. Chapitre 7.
- H.Nussbaumer, Téléinformatique, Presses polytech. romandes, tome 2, 1987. Chapitre 3.
- A.Tanenbaum, Réseaux , InterEditions, 1997. Chapitre 6.

# 1. Introduction

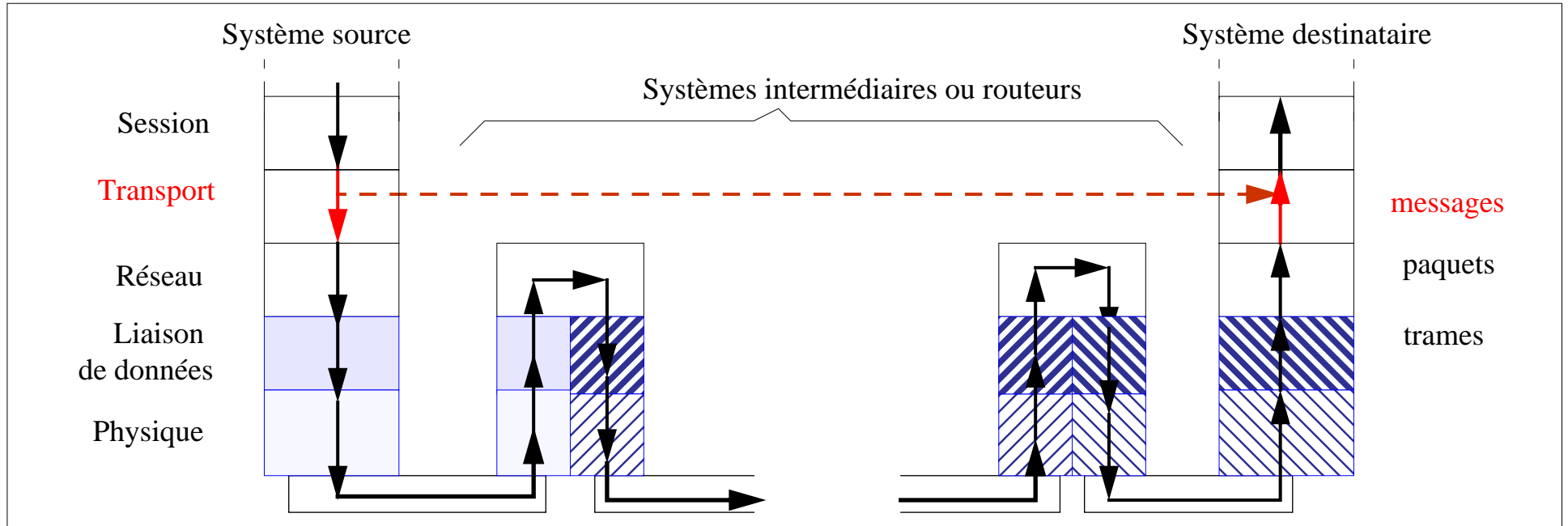
## 1.1. Présentation

- ❑ La couche Transport du modèle OSI :
  - quatrième couche du modèle de référence
  - dernière couche des services de bas niveau.

❑ La couche Transport assure la charnière entre les aspects applicatifs et les aspects liés à la transmission de données sur un réseau informatique.

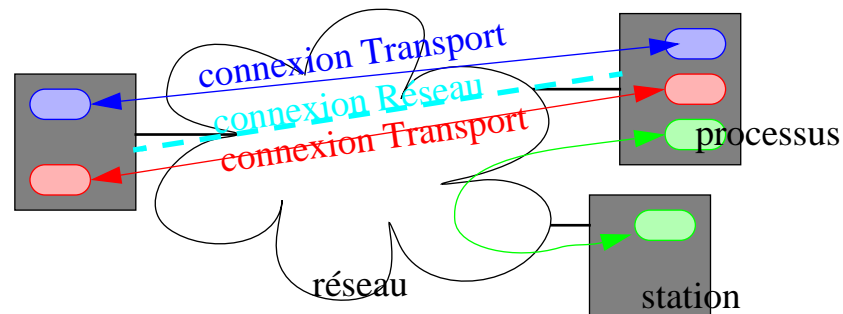


- ❑ La couche Transport assure le transport de **bout en bout** des données d'une façon sûre et efficace.
  - transfert d'informations entre systèmes d'extrémité
  - de manière efficace, fiable et économique,
  - indépendamment de la nature des réseaux sous-jacents .

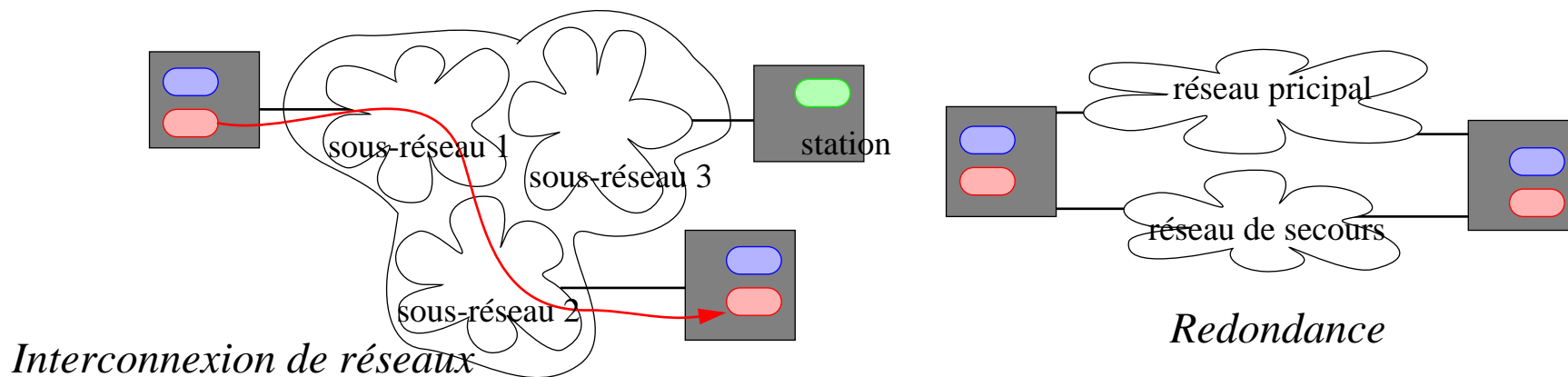


## 1.2. Ses fonctions

- Les entités communicantes ne sont plus des stations mais des processus :
  - adressage spécifique
  - multiplexage des connexions Transport sur une connexion Réseau reliant les stations supportant les processus.



- La connectivité doit être assurée et maintenue en dépit de la multiplicité des réseaux utilisés pour l'acheminement.



- ❑ La couche Transport peut offrir les deux modes classiques de service :
  - le mode connecté et le mode non connecté
- ❑ Deux types de services de transmission :
  - flux continu de données (“stream oriented service”) : un seul T\_SDU !
  - séquence d’unités de données (“record oriented service”) : une suite de T\_SDU
- ❑ Le message : c’est l’unité de données de transfert de niveau Transport (**T-PDU**).
  - Il est généralement de taille variable et composée d’un champ d’entête et d’un champ d’information.

## 2. Mécanismes généraux

### 2.1. Qualité de service (QoS)

La couche Transport optimise l'utilisation des ressources disponibles (du Réseau) pour atteindre (et maintenir) la **qualité de service** (de transmission de données) souhaitée par les entités supérieures (applicatives).

❑ La notion de QoS peut être définie par l'ensemble des **paramètres** (délai de transit, débit, etc.) caractérisant les besoins des entités applicatives.

❑ La valeur des paramètres peut être négociée :

- Lors de la phase d'établissement de la connexion, en fonction :
  - . des caractéristiques des réseaux sous-jacents (↓)
  - . des demandes des entités supérieures locales et distantes (↑)
  - . des mécanismes protocolaires disponibles (↔)

❑ Pour chaque paramètre, l'OSI distingue trois ensemble de valeurs : **préférées**, **acceptables** et **inacceptables**.

❑ La couche Transport joue un rôle de **garant de la QoS** : elle surveille les paramètres de QoS et elle doit déterminer si elle est en mesure de respecter ses engagements.

### 2.1.1 Quelques paramètres de QoS

- le délai d'établissement de la connexion
- la probabilité d'échec d'établissement de la connexion
- le débit de la connexion ("throughput") : dans chaque sens
- le temps de transit
- le taux d'erreur résiduel
- la probabilité d'incident de transfert : non respect des contraintes précédentes
- la qualité de la sécurité de la connexion : niveau de protection
- la priorité relative des connexions entre-elles : dégradation des moins prioritaires
- la probabilité de résiliation de la connexion
- le délai de libération de la connexion
- la probabilité d'échec de libération de la connexion !
- etc.

 Un **contrat** est ainsi déterminé entre les entités supérieures ayant demandé l'établissement de la connexion et la couche Transport.

## 2.2. Services fournis par la couche Réseau

❑ Les mécanismes mis en oeuvre au sein d'un protocole Transport (pour rendre le service demandé) dépendent de l'environnement dans lequel il opère : c'est-à-dire des services fournis par la couche Réseau.

Du point de vue de la couche Transport, on distingue **trois niveaux** de service Réseau :

- niveau A (service fiable sans désynchronisation),
- niveau B (service fiable avec quelques désynchronisations)
- niveau C (service peu fiable)

### 2.2.1 Service Réseau de niveau A

❑ Taux acceptable d'erreurs résiduelles : La fraction de paquets perdus, dupliqués ou corrompus est négligeable.

❑ Taux acceptable d'incidents signalés : Les désynchronisations (N\_RESET()) sont très rares.

❑ Cette catégorie de service est fournie par les **réseaux locaux**.

☞ Le travail de la couche Transport est très simple.

### 2.2.2 Service Réseau de niveau B

- ❑ Taux acceptable d'erreurs résiduelles.
- ❑ Taux inacceptable d'incidents signalés : la couche Réseau émet trop fréquemment des commandes de réinitialisation.
- ❑ Cette catégorie de service est fournie par les **réseaux publics** (X25).
- ☞ Le travail de la couche Transport devient un peu plus complexe :
  - établissement d'une nouvelle connexion
  - resynchronisation,
  - poursuite de la transmission avec masquage des N\_RESET à l'utilisateur.

### 2.2.3 Service Réseau de niveau : service peu fiable

- ❑ Taux inacceptable d'erreurs résiduelles : pertes, duplications ou corruptions.
  - ❑ Taux inacceptable d'incidents signalés.
  - ❑ Cette catégorie de service est fournie par les **réseaux à longue distance** en mode non connecté et les **réseaux par ondes hertziennes**.
- ☞ La couche Transport doit résoudre tous les problèmes de fiabilité (analogues à ceux de la couche Liaison de données).

## 2.3. Transfert de données

Des services de transfert de données spécifiques sont définis :

❑ Les **données exprès** (Expedited Data delivery”)

- traitement et transmission rapide de données
- à ne pas confondre avec les différents niveaux de priorités que peuvent avoir les unités de données normales
- de petite taille (<16 octets)
- but : la transmission d’alarme
- non soumis aux contrôles habituels (de flux, numérotation, etc.)

❑ Transfert de **données lors de l’établissement** et lors de la libération de la connexion

- transmission de données en pseudo-mode non connecté
- transmission de données accélérant l’établissement de la connexion

## 2.4. Sécurité

La couche Transport est une couche favorable à la mise en oeuvre de mécanismes de protection : elle prend en compte le transfert de bout en bout des informations.

Toutefois des mécanismes de protection peuvent être mis en oeuvre à n'importe quels autres niveaux : notamment Application ou Réseau.

□ On peut identifier deux types d'attaques :

- les attaques passives, et les attaques actives
  - . L'intrus écoute les informations échangées sur les connexions de Transport
  - . L'intrus peut agir sur les informations échangées : modification, reproduction, addition, sabotage.

Exemple :

- “replay” : réexécution d'une séquence de messages préalablement capturées
- “spoofing” : l'intrus se fait passer pour quelqu'un d'autre (fausse identité)

Même si les données sont chiffrées au niveau supérieur, l'analyse du message peut fournir des informations (directes ou indirectes) sur l'identité des correspondants, la fréquence et l'intensité de leurs échanges

□ Quelques techniques de protection :

- chiffrement (cryptage),
- authentification (des partenaires),
- intégrité des données (signature).

## 2.5. Adressage

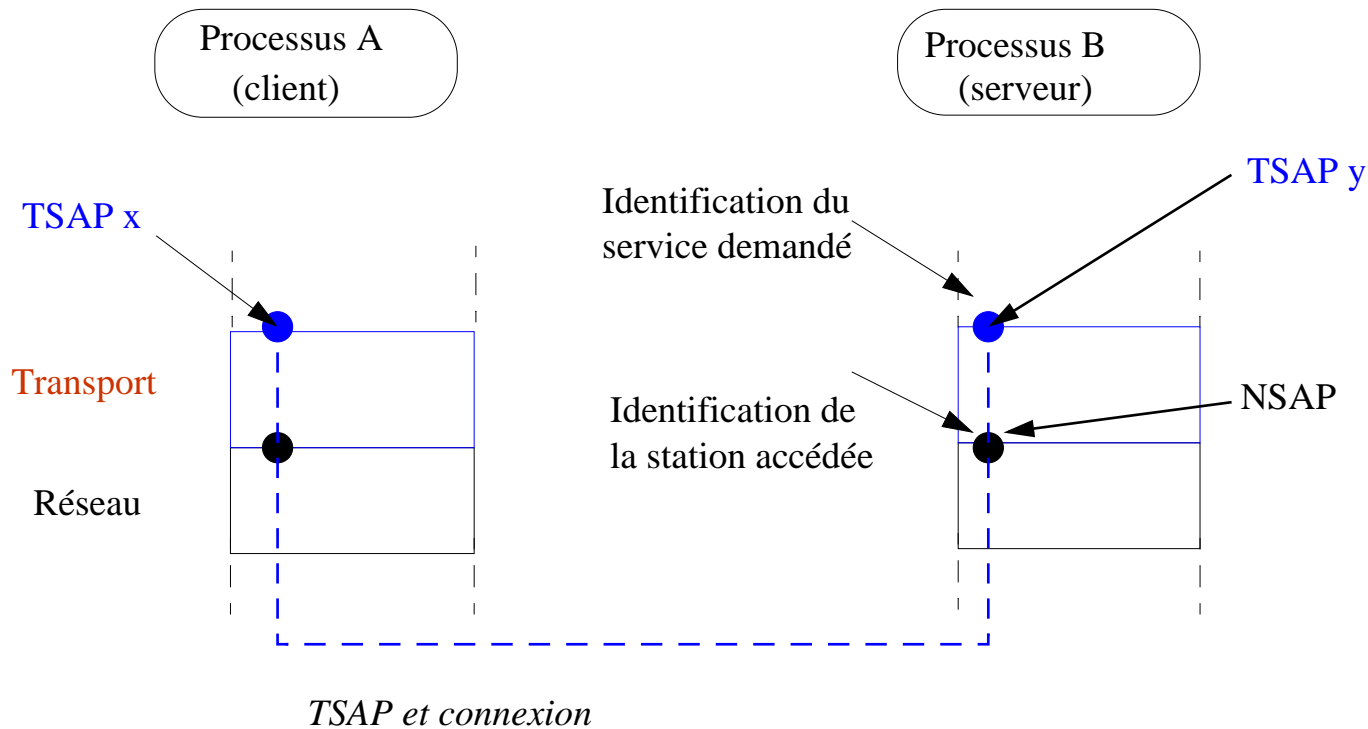
### □ Double besoin pour les adresses :

- identifier les processus dans les stations
  - . durée de vie égale à celle du processus : réutilisation des mêmes adresses
  - . connaissance de ces adresses limitée aux intervenants
- identifier des services généraux (“Well known generic service address”)
  - . durée de vie permanente
  - . connaissance totale étendue à tous les acteurs potentiels

L’adresse Transport complète l’adressage Réseau.

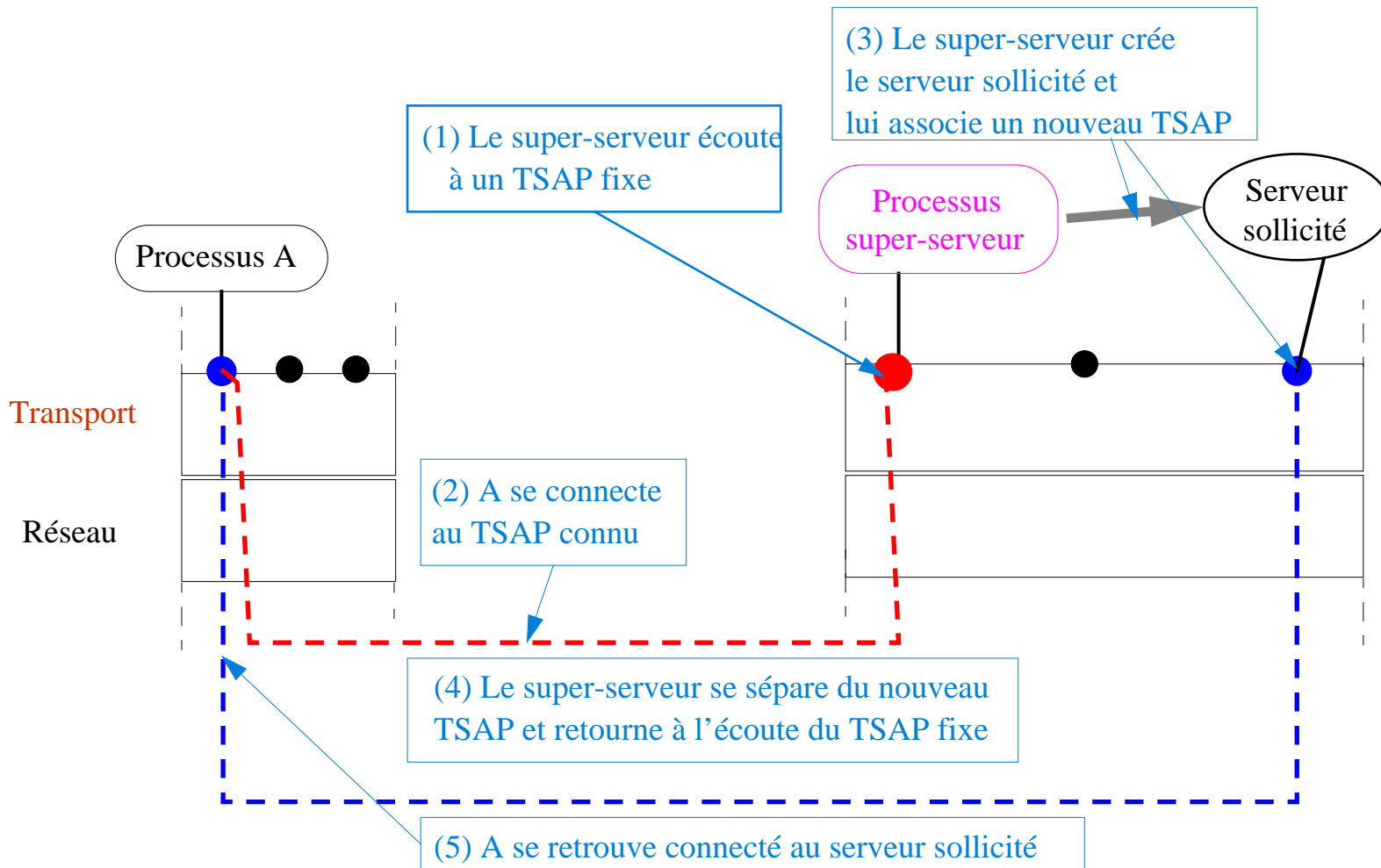
### □ L’adressage s’effectue via les points d’accès au service Transport :

- **TSAP** (“Transport Service Access Point”) auxquels chaque processus vient se connecter et où il attend une demande de connexion.



- Ces adresses peuvent ne pas figurer explicitement dans les messages de données échangés :
  - Numéro de référence :
    - . Accélération du traitement - recherche du contexte rapide par accès direct au contexte grâce au numéro de référence qui sert d'index.
    - . Minimisation du volume occupé dans l'entête.

□ Problème d'accès à un serveur :



*Etablissement d'une connexion avec un serveur*

## 2.6. Etablissement de la connexion en 2 ou 3 phases

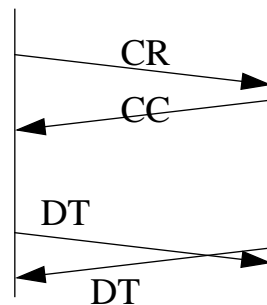
Deux séquences d'échange de messages lors de l'établissement de la connexion peuvent être définies :

- En **deux phases** (“two way handshake”) :
  - . service Réseau assure la séquentialité (généralement en mode connecté)
- En **trois phases** (“three way handshake”)
  - . service Réseau n'assure pas la séquentialité (généralement en mode non connecté)

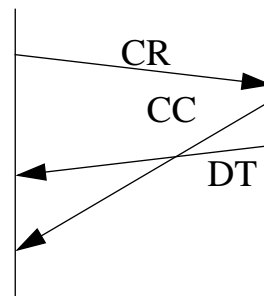
Le DT PDU peut être reçu avant que la confirmation d'établissement de la connexion (CR PDU) soit reçu :

- soit mémorisation du DT PDU dans un contexte indéfini (hors connexion) !
- soit destruction du DT PDU correct !!

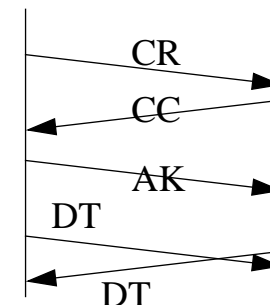
CR : connexion request  
 CC : connexion confirm  
 DT : data  
 AK : acknowledgment



*Etablissement de la connexion en 2 phases*



*Etablissement de la connexion en 3 phases*



L'établissement de la connexion en trois phases permet une négociation plus approfondie des paramètres de la connexion :

- proposition de l'initiateur de la connexion
- contre-proposition du sollicité
- accord définitif de l'initiateur

En deux phases, le sollicité peut seulement choisir parmi les propositions de l'initiateur ou refuser.

## 2.7. La gestion de la connexion

### 2.7.1 Présentation

La gestion de la connexion est plus développée :

- la phase d'établissement de la connexion est aussi une phase de négociation des paramètres de QoS
- plusieurs formes : “two-way and three-way handshake”

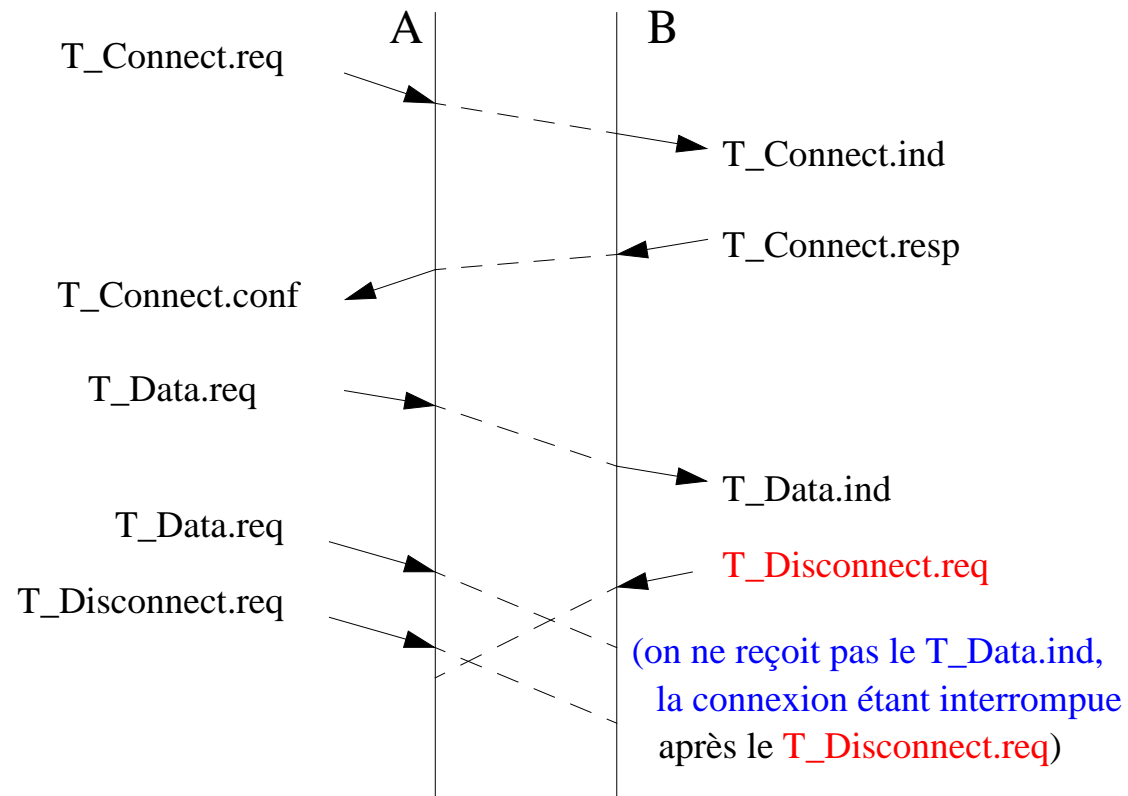
Les procédures d'établissement et de libération doivent être symétriques :

- chaque utilisateur peut prendre l'initiative des opérations

On définit deux types de libération de la connexion :

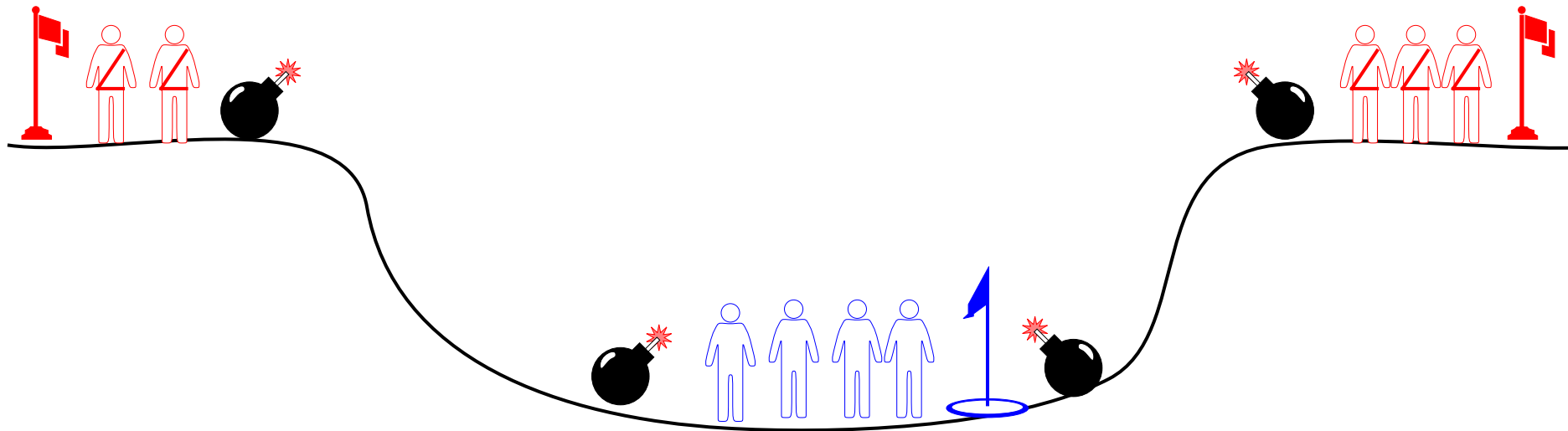
- libération ordonnée : toutes les unités de données en transit sont délivrées avant la libération effective
- libération brutale (“abrupt connection release”/ “abort”) : + rapide

## 2.7.2 La libération d'une connexion



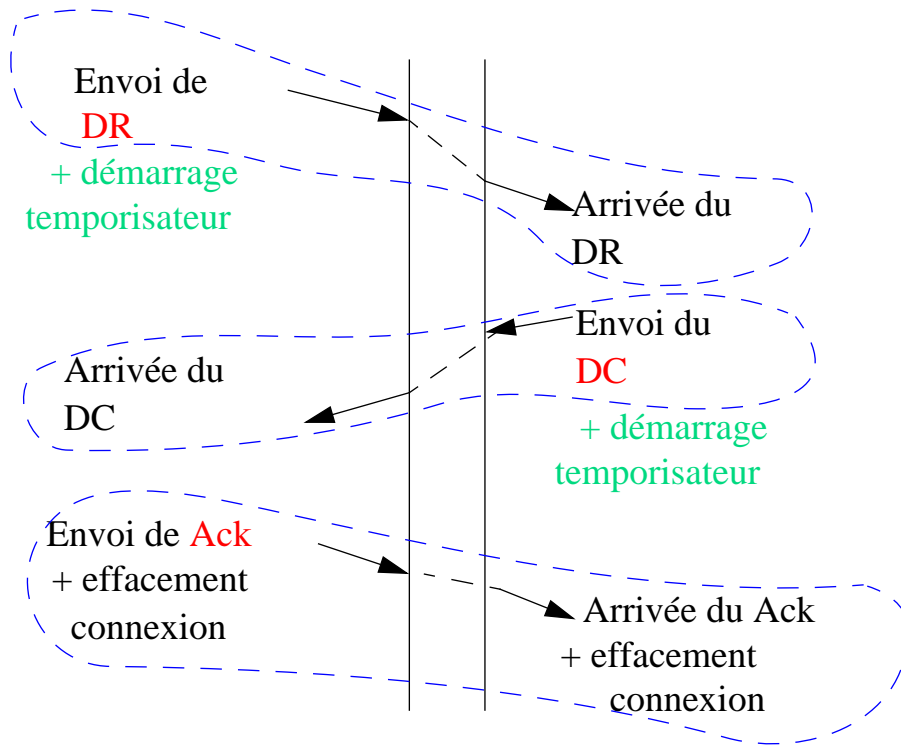
*Exemple de déconnexion brutale avec perte de données*

### 2.7.3 La solution ?

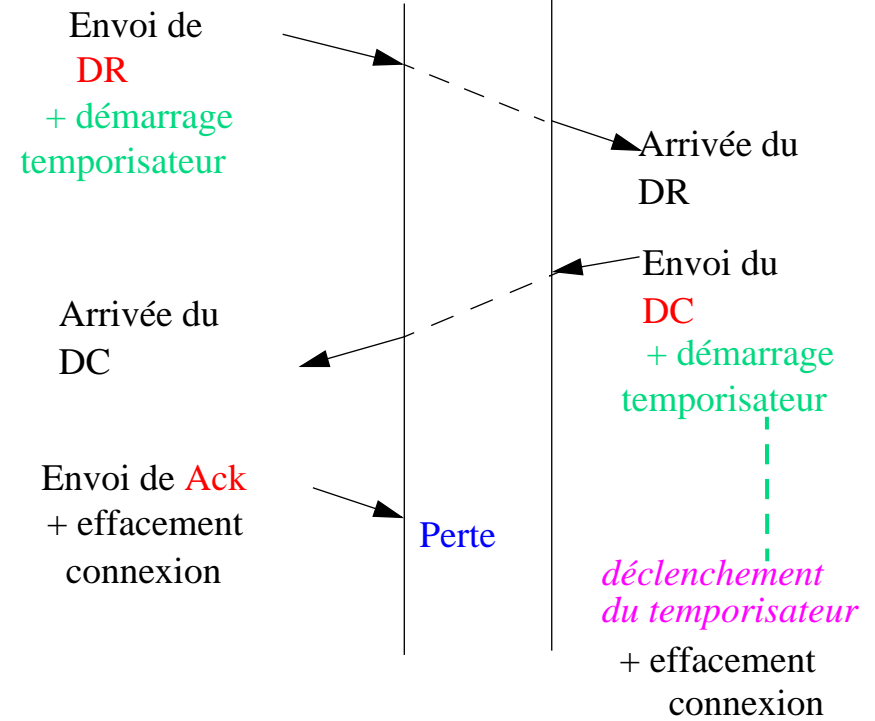


*Le problème des deux armées byzantines :*

*Comment les armées rouges vont-elles faire pour attaquer ensemble et vaincre ?*

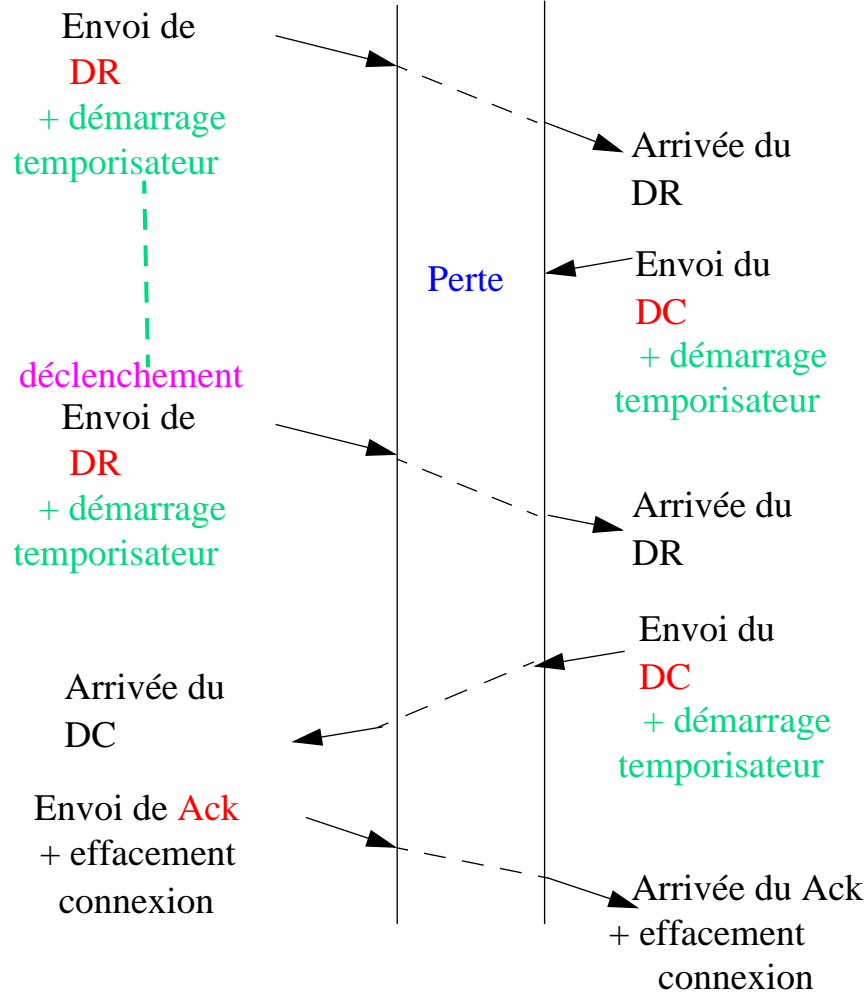


(a) Cas normal

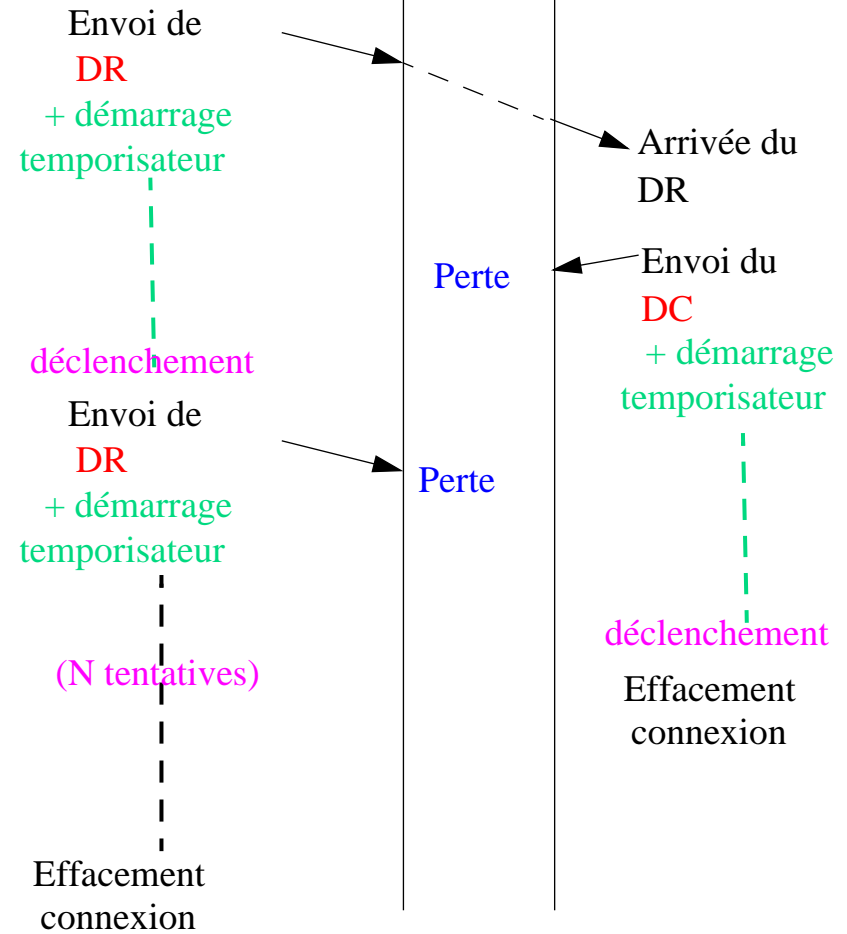


(b) Dernier acquittement perdu

Exemples de déconnexion en trois phases



(c) Confirmation de déconnexion perdue



(d) Confirmation de déconnexion et demandes de déconnexion perdues

Autres exemples de déconnexion

## 2.7.4 Rapports sur l'état de la connexion

### □ “Status reporting”

Permet aux utilisateurs de connaître la valeur des paramètres contractuels

Et les conditions actuelles de fonctionnement de la connexion

Exemple : classe de protocole utilisé, adresse de réseau et de transport, caractéristiques de la connexion (débit, délai de transit, etc.), dégradation de la qualité du service, valeurs courantes des temporisateurs.

## 2.8. Autres problèmes engendrés par le service Réseau

Certains messages peuvent errer dans le réseau pendant un certain temps :

- notamment si le service Réseau est en mode non connecté
- les paquets suivent des chemins indépendants,
- le mécanisme de retransmission produit des doubles,
- certaines portions du réseau peuvent se trouver isolées du reste :
  - . soit par panne de routeur ou par rupture de liaisons,
  - . soit à cause de l'instabilité (vitesse de convergence) des algorithmes de routage.
- les routeurs doivent mémoriser les paquets qu'ils ne peuvent temporairement acheminer.

Solution : **limiter la durée de résidence** d'un paquet au sein du réseau

- et adapter les mécanismes de la couche Transport en conséquence

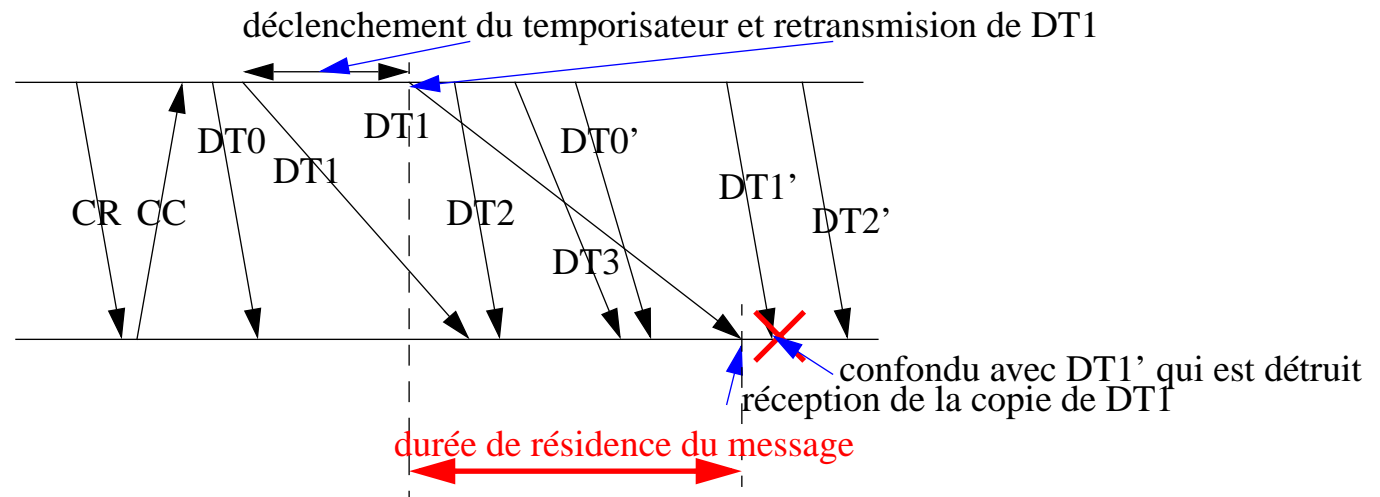
Exemple :

- dans Internet, les datagrammes IP possède un champ Time To live (TTL).

### 2.8.1 Au sein d'une même connexion

Les messages sont identifiés par une numérotation cyclique !

❑ La durée d'utilisation du cycle doit être supérieure à la durée maximale de vie des messages au sein du réseau.



Une solution à ce problème peut être apportée grâce à un plus grand cycle de numérotation :

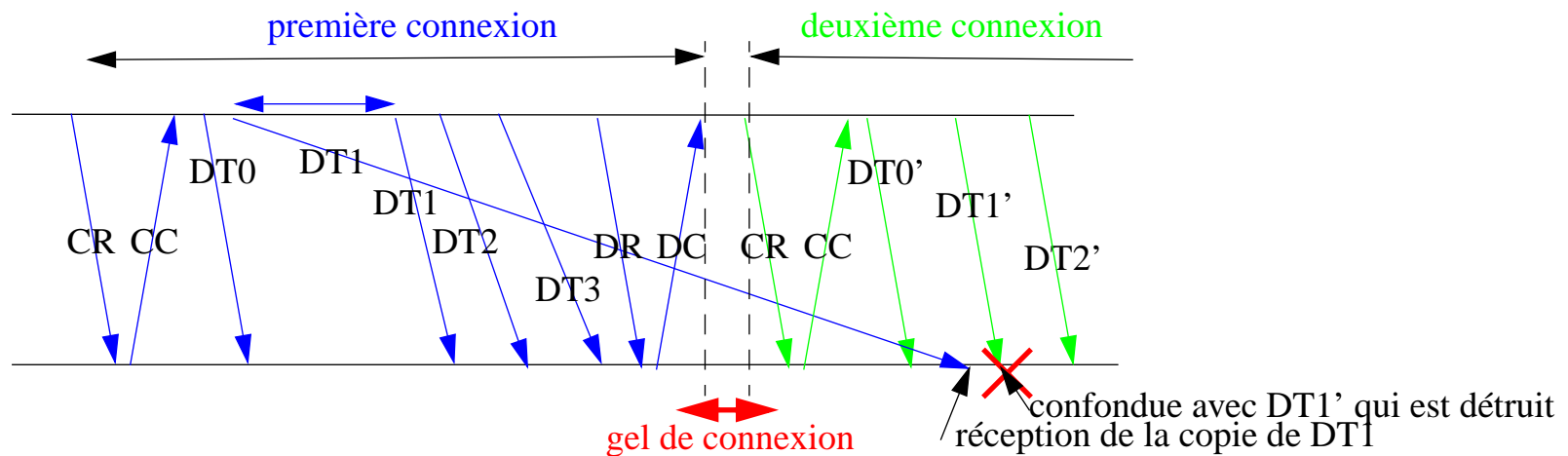
- modulo  $2^7$  --> modulo  $2^{15}$  !
- ou une durée de résidence plus faible.

Autre propriété : durée du temporisateur de retransmission > durée de résidence

## 2.8.2 Confusion entre connexions

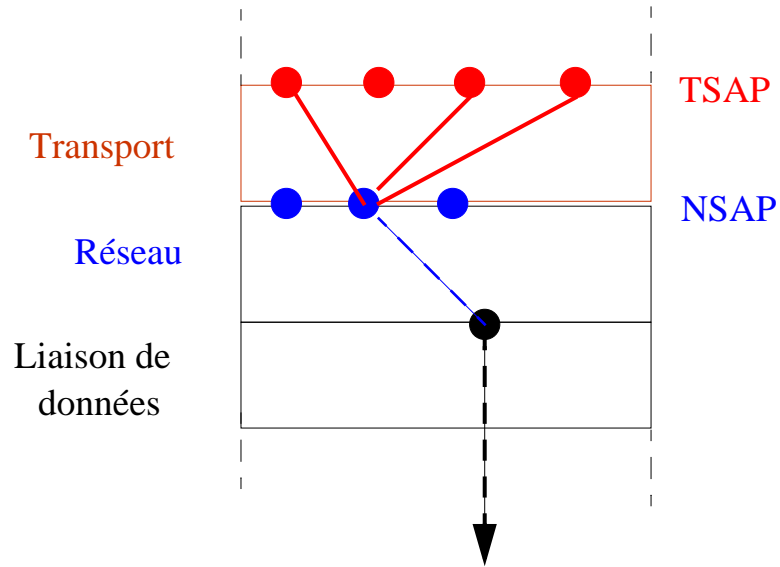
Réutilisation de l'identificateur de connexion après une durée de garde supérieure à la durée maximale de vie des messages au sein du réseau.

La référence d'une connexion libérée est gelée pendant cette durée.

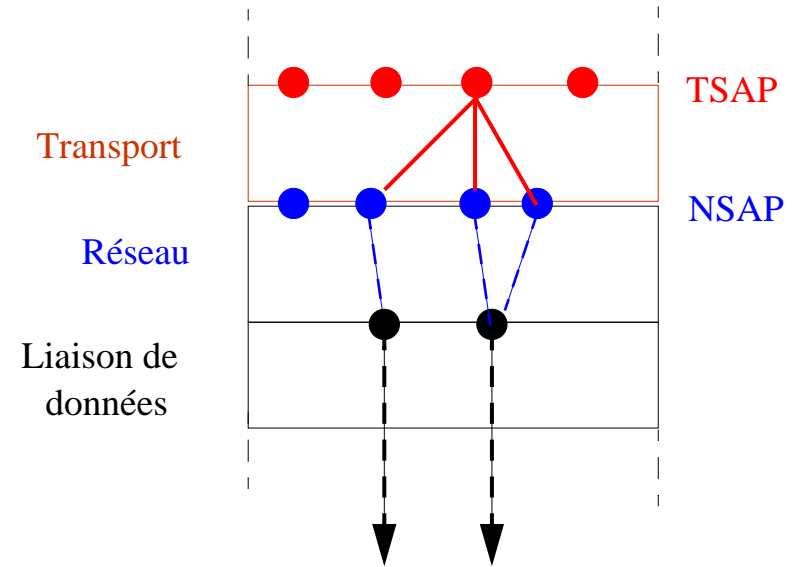


Propriété : durée du gel de connexion > durée de résidence des messages

## 2.9. Multiplexage et éclatement



*Multiplexage (amont)*



*Eclatement (ou multiplexage aval)*

### 3. Le service Transport

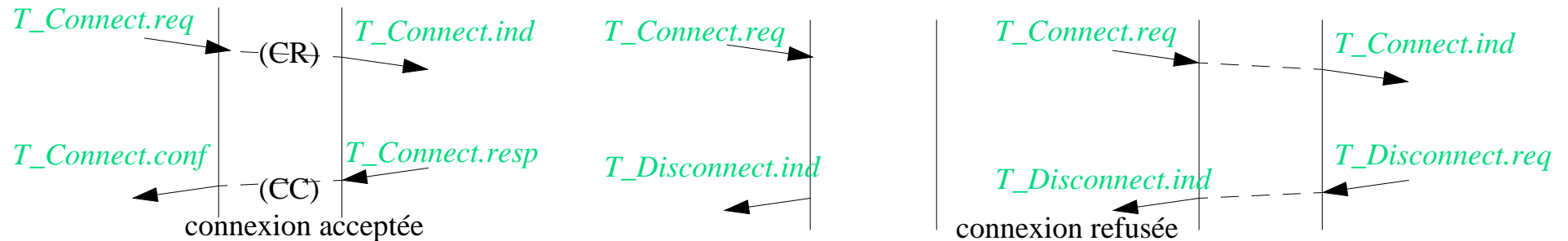
Définition du service de la couche Transport :

- En mode connecté
- norme ISO 8072 ou CCITT X.214

Transport OSI en mode non connecté (Additif 1 de la norme 7498).

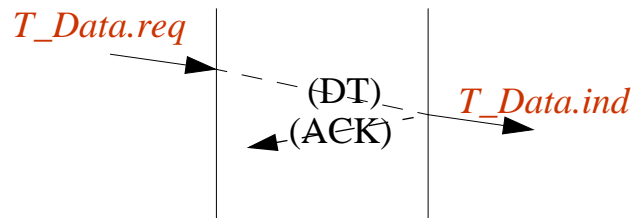
### 3.1. En mode connecté

■ **Etablissement de connexion** avec négociation de la QoS à l'aide de quatre primitives *T\_CONNECT* : *T\_Connect.req*(adresse destination, adresse source, option de données exprès, qualité de service, données utilisateur), *T\_Connect.ind*(idem), *T\_Connect.resp*(adresse de réponse, option de données exprès, qualité de service, données utilisateur) et *T\_Connect.conf*(idem).

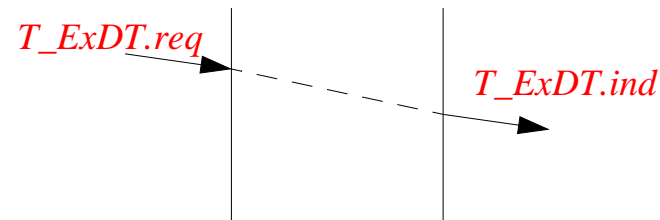


Longueur de données utilisateur  $\leq 32$  octets.

■ **Transfert de données (“normales”) avec contrôle de flux** sur une connexion de Transport avec les primitives *T\_Data* : *T\_Data.req*(données utilisateur) et *T\_Data.ind*(idem)

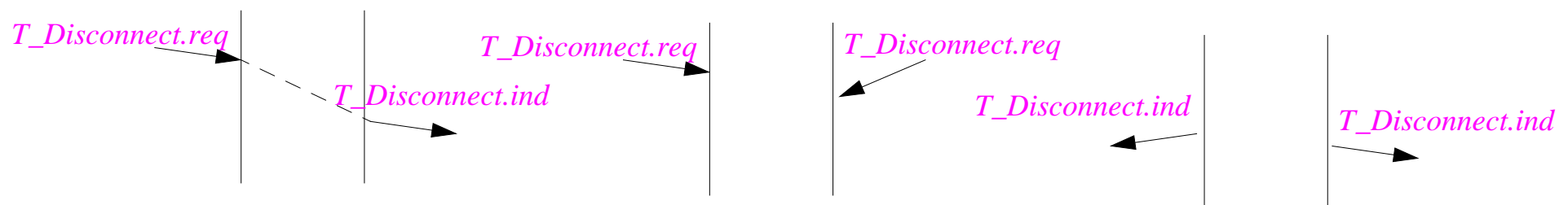


■ **Transfert de données exprès** (optionnel) avec les deux primitives *T\_Expedited\_DATA* : *T\_Expedited\_Data.req(données utilisateur)* et *T\_Expedited\_Data.ind(idem)*.



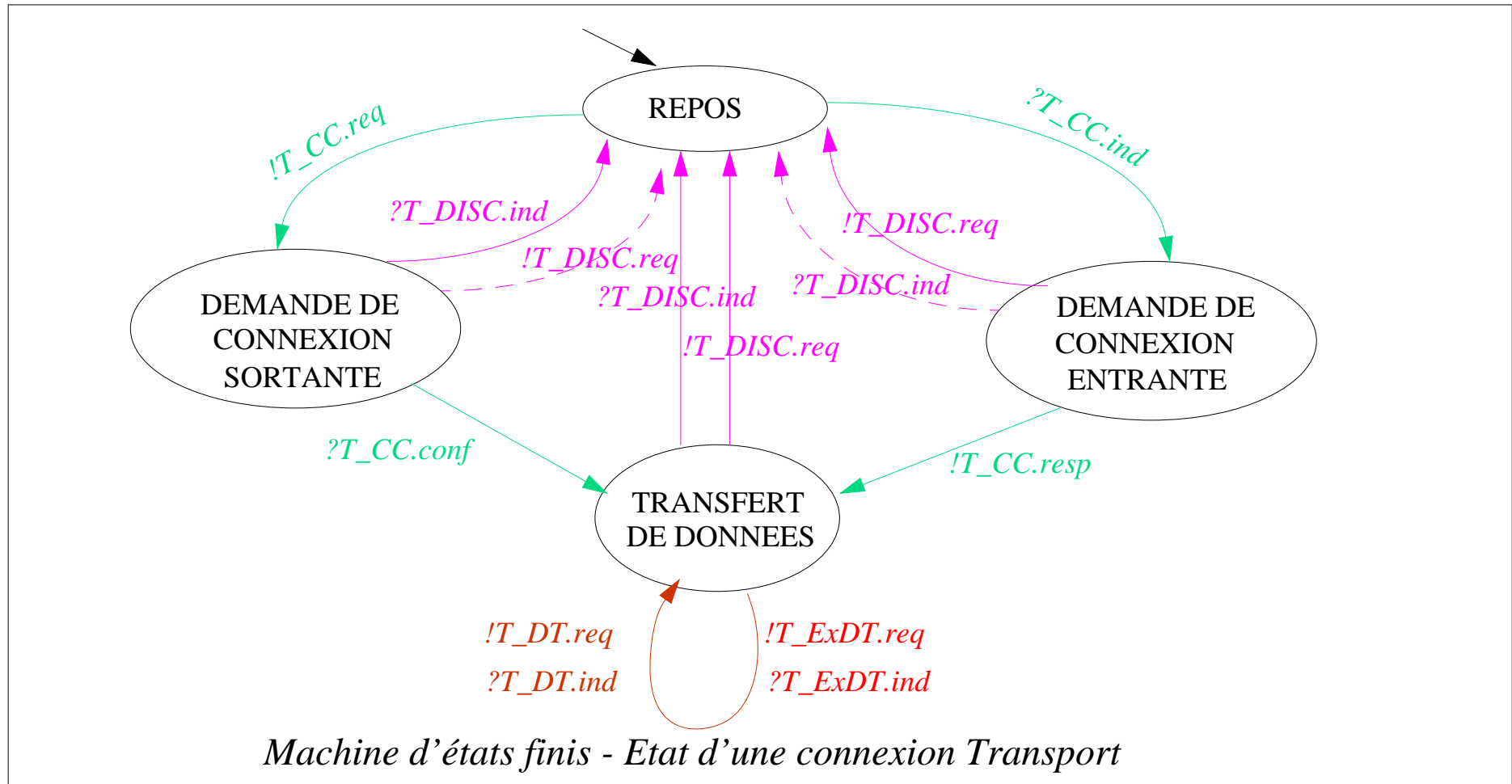
Longueur de données utilisateur  $\leq 16$  octets.

■ **Libération de la connexion** avec les primitives *T\_Disconnect* : *T\_Disconnect.req(données utilisateur)* et *T\_Disconnect.ind(cause de la déconnexion, données utilisateur)*.



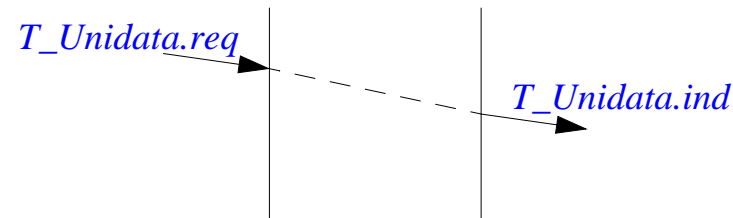
Longueur de données utilisateur  $\leq 64$  octets

L'enchaînement de ces différentes primitives est décrit formellement par :



3.2. □ En mode non connecté:

■ L'utilisateur du service Transport ne dispose que de deux primitives *T\_Unidata* liées au transfert de données : *T\_Unidata.req(adresse destination, adresse source, qualité de service, données utilisateur)* et *T\_Unidata.ind(idem)*.



**Remarque** : Ces deux primitives véhiculent tous les paramètres nécessaires à leur transmission (adressage, QoS,...) alors que les primitives T-Data du mode connecté n'en ont pas besoin.

## 4. Le protocole de Transport

□ Spécification du protocole de couche Transport :

- **TP** (“Transport Protocol”)
- Norme ISO 8073 ou CCITT X.224

□ Les mécanismes exacts mis en oeuvre au sein du protocole Transport dépendent de l’environnement dans lequel il opère : c’est-à-dire du niveau de service de la couche sous-jacente (Réseau).

On distingue **trois niveaux de service Réseau** :

- niveau A : service parfait,
- niveau B : service correct mais avec quelques incidents signalés,
- niveau C : service insuffisant.

□ **Cinq classes** de mécanismes (procédures) mis en oeuvre au sein du protocole Transport sont définies pour s’adapter :

- aux différents niveaux de service Réseau
- aux différents besoins des applications

## 4.1. Les classes de protocole Transport

### 4.1.1 La classe 0 (*classe de base*)

- ❑ Gestion de la connexion : accord ou refus d'établissement.
- ❑ Transfert de données normales (pas de données exprès)
- ❑ Segmentation des données en plusieurs messages, si nécessaire.
- ❑ Libération implicite de la connexion
- ❑ Pas de numérotation
- ❑ Pas de séquençement, ni de contrôle de flux.
- ❑ Pas de contrôle d'erreur sophistiqué :
  - . détection des erreurs protocolaires
  - . simple libération de la connexion.



Le protocole de classe 0 est simple. Il est prévu pour fonctionner au-dessus des **services Réseau de niveau A**.

Exemple : télex

#### 4.1.2 La classe 1 (*classe de base avec reprise sur erreur*)

- Classe 0 +
- Concaténation et séparation
- Libération explicite de la connexion
- Numérotation des TPDU DT
- Echange de données exprès soit dans des paquets de données soit dans des paquets exprès de la couche Réseau
- Capacité à maîtriser les resynchronisations (N\_RESET) de la couche Réseau.
- Gel des références
- Pas de contrôle de flux, pas de technique d'augmentation de la fiabilité.



Cette classe convient à un **service Réseau de niveau B** (service fiable avec resynchronisation).

### 4.1.3 La classe 2 (*classe avec multiplexage*)

- Classe 0 +
- Multiplexage et démultiplexage.
- Concaténation et séparation
- Libération explicite de la connexion
- Numérotation des TPDU DT
- Echange de données exprès dans les paquets de données de la couche Réseau
- Contrôle de flux optionnel.



Cette classe est généralement utilisée sur un **service Réseau de catégorie A**.

### 4.1.4 La classe 3 (*classe avec reprise sur erreur et multiplexage*)

C'est l'union des classes 1 et 2.

- Mais le contrôle de flux est obligatoire.



Cette classe est adaptée au **service Réseau de catégorie B**

#### 4.1.5 La classe 4 (*classe avec détection et reprise sur erreur*)

Cette classe de protocole Transport reprend toutes les fonctionnalités des classes 0, 1, 2, et 3.

- Détection d'erreurs par champ de contrôle d'erreur.
- Retransmission après temporisation
- Remise en séquence.
- Détection d'inactivité.
- Eclatement et recombinaison



Cette classe est prévue pour les **services Réseau de catégorie C** (service insuffisant).

Remarque : Le protocole de classe 4 de l'OSI est connu sous le nom TP4.

## 4.2. Les différents T-PDU

- Le protocole Transport utilise 10 types de TPDU :
  - Demande de connexion (“Connection Request” : CR),
  - Confirmation de connexion (“Connection Confirm” : CC),
  - Demande de déconnexion (“Disconnection Request” : DR),
  - Confirmation de déconnexion (“Disconnection Confirm” : DC),
  - Données (“Data” : DT),
  - Accusé de réception de données (“Data acknowledgment” : AK),
  - Données exprès (“Expedited Data” : ED),
  - Accusé de réception de données exprès (“Exp. Data acknowledgment” : EA),
  - TPDU d’erreur (“Error TPDU” : ER),
  - Rejet (“Reject” : RJ).

□ Classes et codes

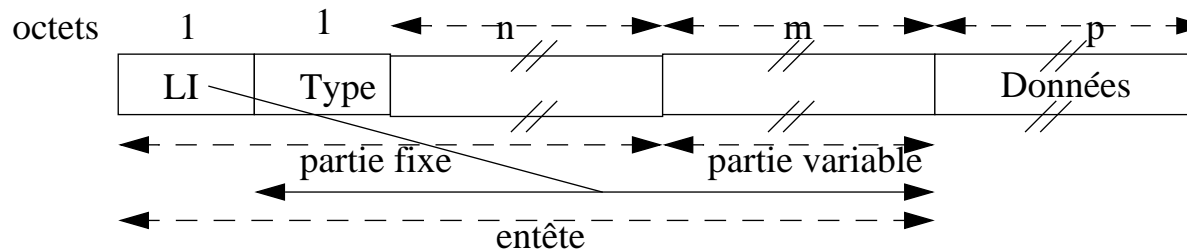
Tableau 1 : Les T\_PDU

TPDU	classes obligatoires	classes optionnelles	code
CR	0,1,2,3,4		1110
CC	0,1,2,3,4		1101
DR	0,1,2,3,4		1000
DC	1,2,3,4		1100
DT	0,1,2,3,4		1111
AK	3,4	1,2	0110
ED	1,3,4	2	0001
EA	1,3,4	2	0010
ERR	1,2,3,4		0111
RJ	1,3		0101

### 4.3. Le format général des TPDU

□ Les TPDU comportent :

- une entête avec partie fixe et variable,
- et un champ de données de longueur variable



*Format général des unités de données du protocole*

Le **champ LI** (“Length Indicator”) :

- longueur de l’entête en octets, non-compris le champ LI lui-même.
- la valeur 255 est réservée pour des futures extensions

Le **champ Type** :

- les 4 premiers bits définissent le type du TPDU
- les 4 bits suivants codent la valeur du crédit pour les TPDU CR, CC, AK et RJ, et valent 0 pour les autres TPDU.

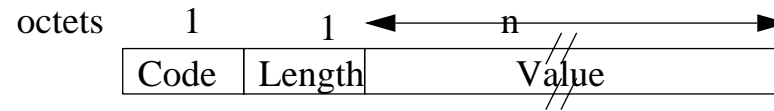
Le reste de la partie fixe dépend du type du TPDU.

#### 4.4. La partie variable de l'entête

Comporte des paramètres en nombre et en taille variables.

□ La technique de codage :

- TLV (Type -longueur-valeur) :



*Format général des paramètres de la partie variable de l'entête*

Le **champ Code** définit le type du paramètre.

Le **champ Length** définit la longueur en octets du champ Value.

Le **champ Value** contient la valeur du paramètre

□ Quelques paramètres :

- Taille maximale des TPDU : par défaut et au minimum 128 octets [0xD0]
  - . sur 1 octet : la puissance de 2 de la taille en octets (de  $2^7$  à  $2^{13}$ )
  - . Exemple : Option fixant la taille maximum à 1Ko :

Type Long. Val.

D0	1	8
----	---	---

Option de 3 octets

- TSAP de l'appelant [0xD1]
  - . champ Value de longueur variable : identifie l'appelant
- TSAP de l'appelé [0xD2]
  - . champ Value de longueur variable : identifie le service appelé
- Champ de contrôle d'erreur [0xD3] sur 2 octets :
  - . premier octet : la somme de tous les octets du TPDU est nulle
  - . deuxième octet : la somme pondéré de tous les octets du TPDU est nulle
  - . calcul arithmétique simple, modulo 255 : "Fletcher checksum"
  - . complète la protection assurée par le calcul polynômial de la couche Liaison de Données
- Négociation d'options [0xD6] :
  - . premier bit : données exprès,
  - . deuxième bit : contrôle d'erreur.

- Classes de protocole de repli [0xD7]
  - . même format que l'octet Class options
  - . plusieurs, par ordre de préférence
- Numéro de version du protocole [0x81]
- Délai maximum d'expédition d'un acquittement (en ms)
- Débit : moyen et minimal, pour chacun des deux sens de transmission
- Délai de transit : moyen et maximal, pour chacun des deux sens de transmission
- Taux résiduel d'erreur : moyen et minimal
- Priorité choisie : de 0 à 64K (0 est la plus élevée)
- Nombre maximum des tentatives de retransmission
- Délai maximum d'établissement des connexions
- etc.

## 4.5. Le champ de données

- Le champ **Données** du TPDU est de longueur variable :
  - sa longueur n'est pas explicitement fournie
  - la longueur du champ Données est déduite sans ambiguïté
    - . grâce à la connaissance apportée par la couche Réseau
    - . et par différence avec la longueur de l'entête

La longueur maximale du TPDU :

- par défaut 128 octets
- peut être négociée lors de l'établissement de la connexion grâce à un paramètre de la partie variable de l'entête.

### 4.6. Le format des TPDU

octets	1	2	3	4	5	6	7					
	LI	CR   Cdt	0		Source Reference		Class options	Partie variable	Données			
	LI	CC   Cdt	Destination Ref.		Source Reference		Class options	Partie variable	Données			
	LI	DR   0	Destination Ref.		Source Reference		Cause	Partie variable	Données			
	LI	DC   0	Destination Ref.		Source Reference		Partie variable					
	LI	DT   0	EOT   T(S)	Données						format réduit		
	LI	DT   0	Destination Ref.	EOT   T(S)	Partie variable		Données			format normal		
	LI	DT   0	Destination Ref.	EOT	T(S)		Partie variable		Données	format étendu		
	LI	AK   Cdt	Destination Ref.	0   T(R)	Partie variable					EA pas de champ crédit		
	LI	AK   0	Destination Ref.		T(R)			8	9	10	Cdt	Partie variable
	LI	RJ   Cdt	Destination Ref.	0   T(R)								2 versions : normale ou étendue
	LI	ER   0	Destination Ref.		Cause	Partie variable						

## 4.7. Les champs

### 4.7.1 Les champs “Destination and source references” :

- ❑ Chaque entité choisit localement sa propre référence, et utilise celle de l’autre !

Accélère la recherche de contexte, et diminue la surcharge.

Ce procédé se différencie :

- des numéros de voie logique (NVL) du protocole X25 car chaque paquet comporte **un seul** NVL et un nouveau NVL est partagé à **chaque couple d’entités** Réseau adjacentes.
- des TSAP, car ceux-ci identifient de manière unique et **permanente** certains services.

Le champ “Destination reference” est nul pour le TPDU CR

Seul le champ “Destination reference” est utilisé lorsque la connexion est établie.

La classe 0 et 1 utilise un format réduit de DT TPDU, sans “Destination reference”

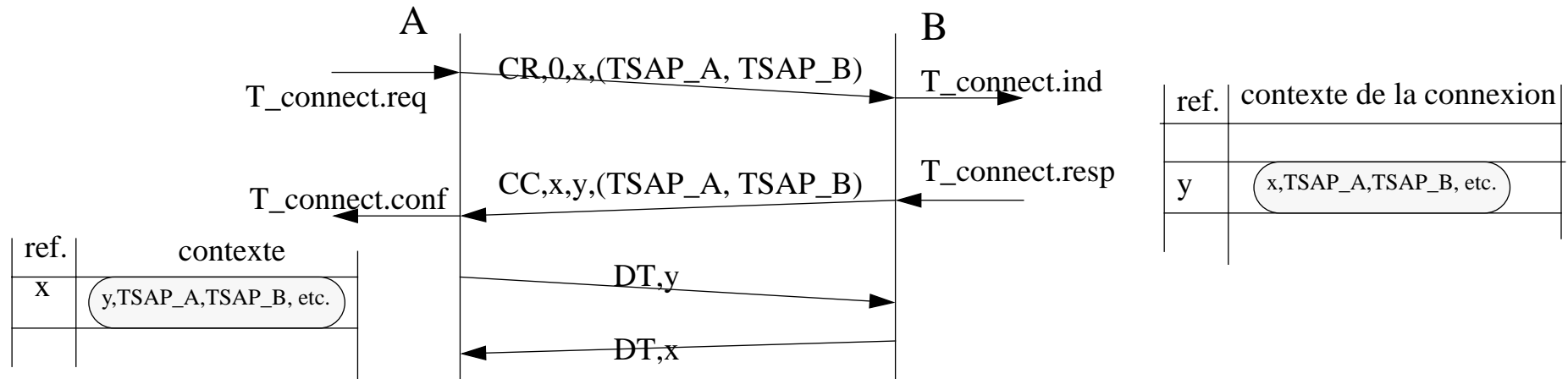


pas de multiplexage

## 4.7.2 L'établissement de la connexion

En trois phases pour la classe 4 et en deux phases pour les autres classes.

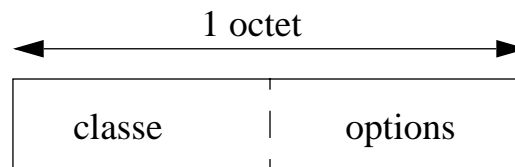
Association entre références et TSAP source et destination :



Les références sont utilisées pour accéder directement au contexte de la connexion lors de la phase de transfert des données.

### 4.7.3 Le champ “Classe options”

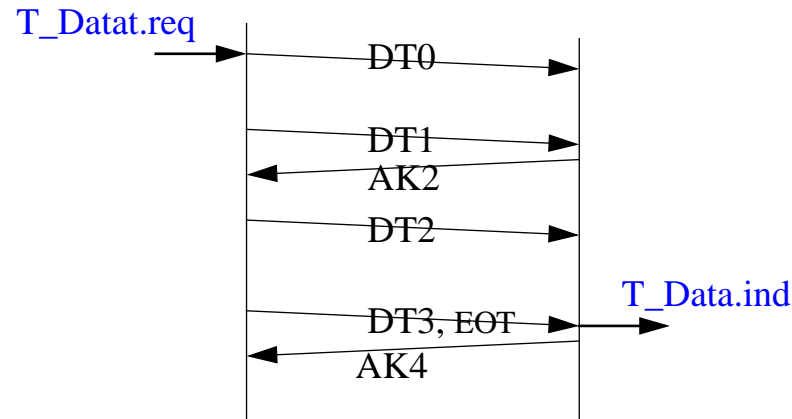
- 4 premiers bits codent la **classe** de Transport préférée :
  - 0000 : classe 0 demandée
  - 0001 : classe 1 demandée
  - etc.
  - des classes de repli peuvent être définies grâce à un paramètre spécifique de la partie variable de l’entête
  
- 4 bits suivants codent les **options** demandées :
  - numérotation normale ou étendue :
    - . T(S) et T(R) sur 1 ou 4 octets, Cdt sur 4 ou 16 bits [classe 2, 3, 4].
  - utilisation du contrôle de flux [classe 2].



#### 4.7.4 Le bit EOT du champ T(S)

- “End of TSDU” : identifie le dernier TPDU du TSDU.

Utilisé par le mécanisme de segmentation :



Echange de plusieurs TPDU pour transmettre un seul message :



Présent dans les TPDU : DT et ED

Fonction équivalente au bit M du protocole X25.

#### 4.7.5 Le champ Cause

□ La cause de la déconnexion (code) :

- demande de la couche supérieure (128)
- TSAP inconnu (134)
- réseau congestionné (129)
- cause inconnue
- etc.

□ La raison peut être précisée à travers un paramètre placé dans la partie variable de l'entête

Utilisé par les TPDU DR, ERR

#### 4.7.6 L'échange de données

□ Les messages sont numérotés (sauf ceux de la classe 0) :

- format normal : 1 octet
- format étendu: 4 octets

□ Utilise le mécanisme de la fenêtre coulissante à crédit variable

La longueur totale du TPDU DT est négociée lors de l'établissement de la connexion.

En classe 0 : pas de format étendu, pas de numérotation des TPDU (toujours 0)

Des données peuvent être transmises par les TPDU CR, CC et DR, mais pas DC :

- toutefois la longueur de leur champ de données doit être inférieure à respectivement 32 et 64 octets

#### 4.7.7 La fenêtre coulissante

Le champ **T(S)** :

- identifie le TPDU DT ou ED (modulo  $2^7$  ou  $2^{31}$ )
- utilisé par le mécanisme de gestion de la fenêtre coulissante ( $\equiv N(S)$  et  $P(S)$ )
- toujours nul pour la classe 0

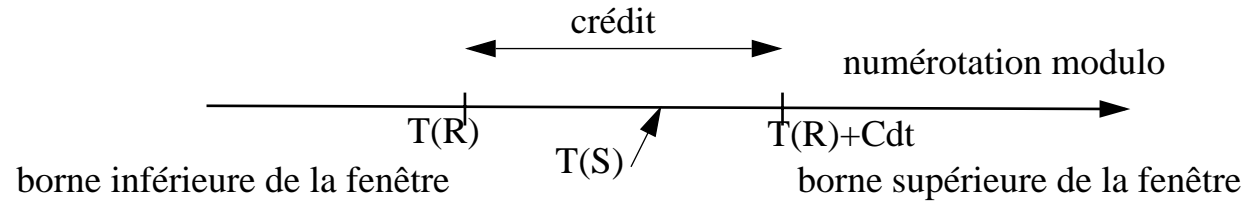
Le champ **T(R)** :

- numéro du prochain TPDU attendu
- utilisé par le mécanisme de gestion de la fenêtre coulissante ( $\equiv N(R)$  et  $P(R)$ )
- présent dans les TPDU AK et EA
- pas de “piggybacking” !

Le champ **Crédit** :

- nombre de TPDU que l'on peut transmettre par anticipation,
- définit la largeur **variable** de la fenêtre coulissante,
- permet de gérer finement le contrôle de flux,
  - . adaptation au capacité de stockage et de traitement du récepteur
  - . adaptation à la charge du réseau
- Deux formats : court sur 4 bits et étendu sur 2 octets.

- présent dans les TPDU DT et ED, CR et CC :
  - . le crédit initial est déterminé par les TPDU CC et CR.



- en classe 4 : le crédit peut être arbitrairement réduit
  - . réquisition : les messages déjà envoyés sont détruits
- Crédit = 0 : émission interdite !

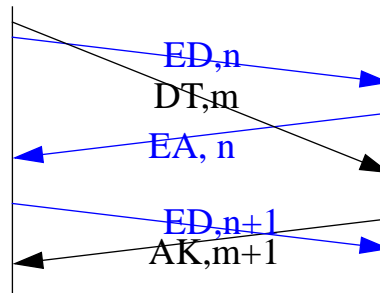
❑ Pas de piggybacking !

#### 4.7.8 L'échange de données exprès

- Transmission indépendante du contrôle de flux appliqué aux flux de données (normales)

Un seul TPDU ED à la fois, numéroté et acquitté par un seul EA:

- protocole “send and wait”



Longueur maximum du champ de données : 16 octets

- Le bit EOT est toujours à 1 (pas de multiplexage possible ni utile)!

Un TSDU ED émis avant un TSDU DT est toujours remis avant ce dernier :

- il n'est jamais doublé...
- ... mais il peut doubler

#### 4.7.9 Traitement des erreurs

□ Demande de retransmission :

- les TPDU REJ
- retransmission des TPDU de numéro supérieur ou égal au champ T(R)

Utilisé par les classes 1 et 3.

Employé après réception d'un TPDU incorrect :

- Corruption : détectée grâce au champ de contrôle d'erreur

Employé après un dé(re)-synchronisation de la couche inférieure :

- paquet N\_Reset

□ La classe 4 met en oeuvre un mécanisme de retransmission par temporisateur au niveau de l'émetteur.



## 5. Conclusion

La couche Transport

Entre couches basses (réseau) et couches hautes (application)

Offre adaptée aux besoins de la couche supérieure (↑), en fonction des services offerts par la couche Réseau (↓) :

- OSI : TP a 5 classes
- Internet : 2 protocoles (TCP ou UDP)

Notion de QoS

Mécanisme de fenêtre coulissante à crédit

Le surcoût dû à l'empilement Transport+Réseau+Liaison de données devient important :

- quantité d'octets transférés vis-à-vis des données utiles,
- temps de traitements liés aux mécanismes mis en oeuvre.