

## Chapitre 8 : Le niveau Session

/home/kouna/d01/adp/bcousin/REPR/Cours/8.fm - 16 Janvier 1998 10:24

### Plan

- Introduction
- Le service et les mécanismes généraux :
  - . Transfert de données, la gestion de la connexion, la gestion de la synchronisation, la gestion des activités
- Le protocole :
  - . les unités fonctionnelles, les TPDU, leurs champs
- Conclusion

### Bibliographie.

- D.Dromard & al., Réseaux informatiques : cours et exercices, Eyrolles, tome 2, 1994. Chapitre 3.
- G.Pujolle & al., Réseaux et télématique, Eyrolles, Tome 1, 1987. Chapitre 12.
- H.Nussbaumer, Téléinformatique, Presses polytechniques romandes, tome 3, 1991. Chapitre 1.
- A.Tanenbaum, Réseaux, InterEditions, 1997. Chapitre 7.

## 1. Introduction

### 1.1. Présentation

#### ❑ Services d'aide à la gestion du dialogue entre les entités Session :

- Structuration des échanges
  - . ex : activités et tour de parole (les jetons)
- Resynchronisation du dialogue
  - . ex : point de synchronisation

#### ❑ Souplesse et adaptation :

- négociation
- les unités fonctionnelles permettent de choisir les fonctions nécessaires pour rendre le service requis par les entités supérieures.

#### ❑ Première des couches hautes !

Définition du service de la couche Session : IS 8326 (X215)

Spécification du protocole de la couche Session : IS 8327 (X225)

## 2. Les mécanismes généraux

Principaux **services** :

- gestion de la connexion : établissement et libération
- transfert de données : normales, exprès et typées
- points de synchronisation
- gestion d'activité et les informations de capacités
- resynchronisation
- signalisation d'anomalie

Les **jetons** :

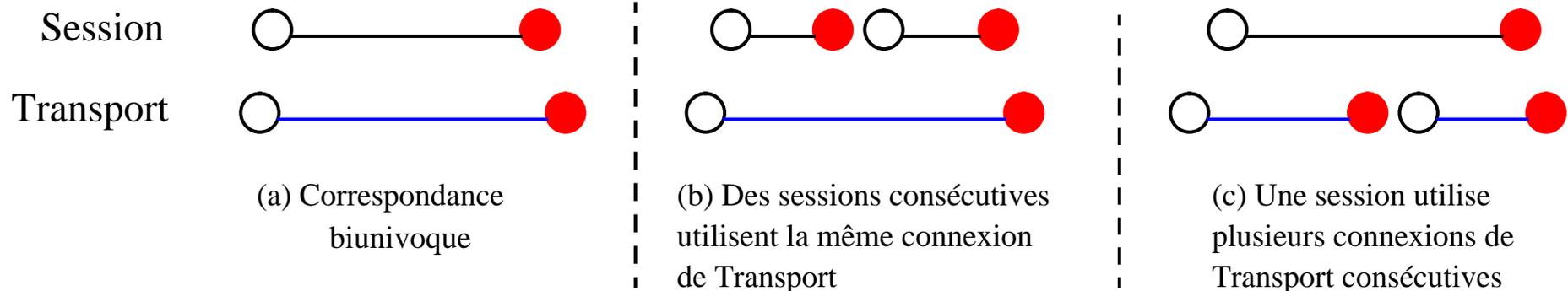
- symbolisent le droit **exclusif** d'utiliser certains services de la couche Session
- Quatre jetons :
  - . Le jeton de données (“data token”)
  - . Le jeton de terminaison (“release token”)
  - . Le jeton de synchronisation mineure (“synchronise minor token”)
  - . Le jeton de synchronisation majeure ou d'activité (“synchronize major/ activity token”)

La disponibilité d'un jeton est définie par l'utilisation du service associé.

## 2.1. La connexion de la couche Session

□ Fournir aux utilisateurs de Session les moyens d'établir des connexions appelées **sessions** et d'y transférer des données en bon ordre.

□ Trois correspondances possibles entre connexions Session et connexions Transport.



☞ Les adresses Session peuvent être différentes des adresses Transport.

□ **Pas de multiplexage** possible de plusieurs connexions (Session) sur une même connexion Transport.

□ **Pas d'éclatement** d'une connexion (Session) sur plusieurs connexions Transport.

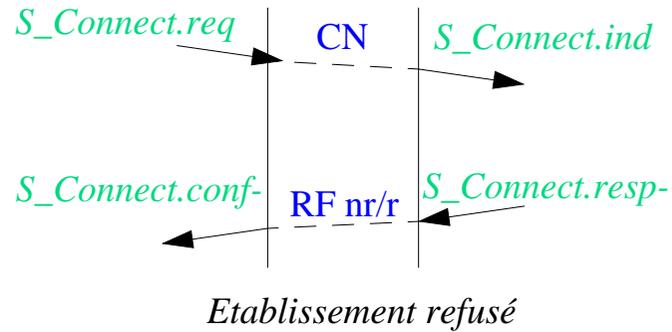
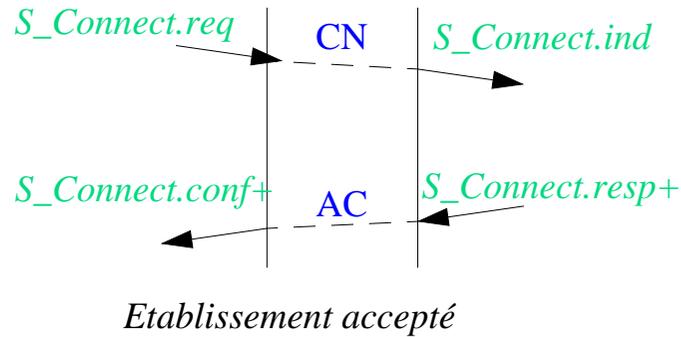
## 2.2. L'établissement d'une connexion

### 2.2.1 Fonctions

Fonctions similaires à celles des connexions des niveaux inférieurs, mais étoffées :

- affectation de la connexion Transport
- mise en correspondance des adresses Transport et Session
- identification de la connexion
- choix des paramètres de qualité de service
  - . débit, délai et taux d'erreur
  - . niveau de priorité relatif des connexions Session entre-elles
  - . protection contre les actions malveillantes
- négociation des paramètres de la connexion
  - . les unités fonctionnelles de session (choix du sous-ensemble : profil)
  - . la version du protocole : 1 = en 1985; 2 = actuelle
  - . négociation de l'attribution initiale des jetons
  - . taille maximum des TSDU
  - . numérotation initiale des points de synchronisation
- transfert de données initiales

## 2.2.2 Primitives



**CN** : connect S\_PDU; **AC** : accept S\_PDU; **RF** : refuse S\_PDU

Une variante :

- **RF-r** : Refus avec réutilisation possible de la connexion Transport vacante
- **RF-nr** : Refus sans réutilisation (notamment si le transfert de données exprès était requis)

Double demande simultanée d'établissement de connexion :

- 🖱️ deux connexions Session (car deux connexions Transport auront été utilisées !)

## 2.3. Libération de la connexion

### 2.3.1 Fonctions

La libération de la connexion peut être :

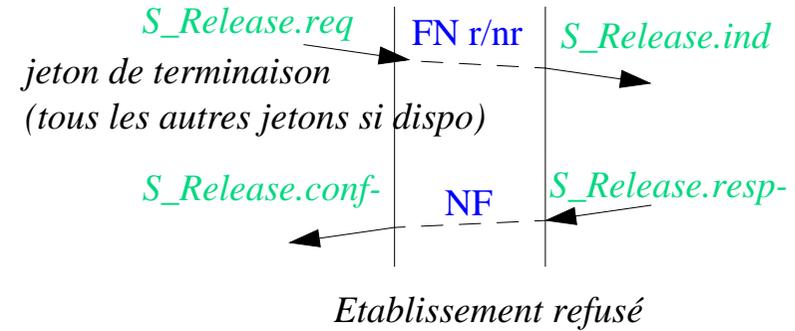
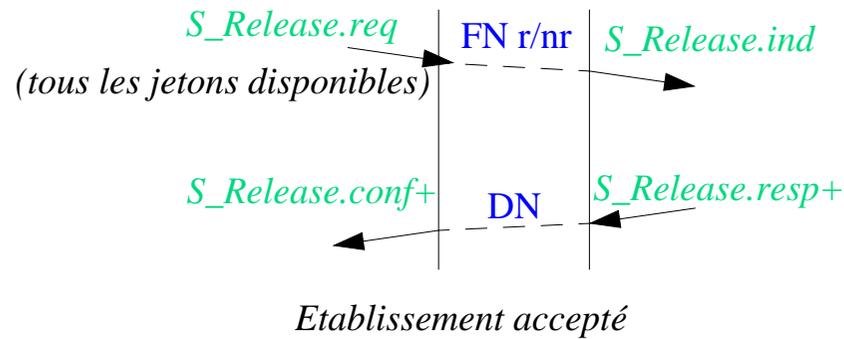
- **brutale** (“abort”) ou **ordonnée** (c’est-à-dire normale)
  - . la libération normale n’entraîne aucune perte de données
  - . requiert la possession de tous les jetons par le demandeur de terminaison
- par l’**utilisateur** (“user”) ou par le **fournisseur** (“provider”) de service

Aucune libération ne peut avoir lieu tant qu’une activité est en cours.

La terminaison peut être négociée :

- l’utilisateur distant peut refuser la libération de la connexion
- unité fonctionnelle particulière
- le *jeton de terminaison* gère le service de terminaison négocié
- la libération libère généralement la connexion Transport associée

### 2.3.2 Primitives



FN : finish S\_PDU; DN : disconnect S\_PDU; NF : not finished S\_PDU

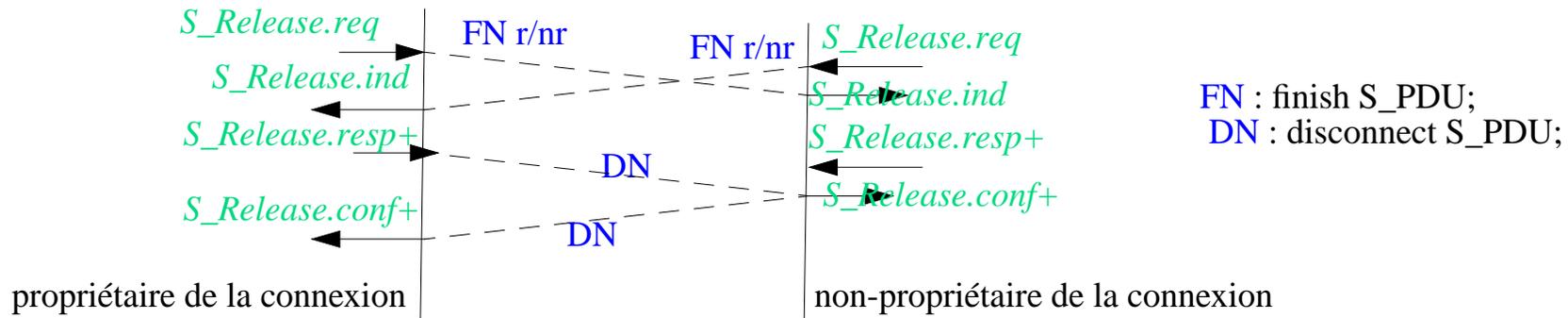
Les primitives possèdent un paramètre *Données utilisateur*.

Une variante :

- FN-n : Terminaison avec réutilisation possible de la connexion Transport vacante
- FN-nr : Terminaison sans réutilisation (notamment si le transfert de données exprès était actif)

Double demande de libération de connexion simultanée :

- impossible si des jetons sont utilisés
- sinon

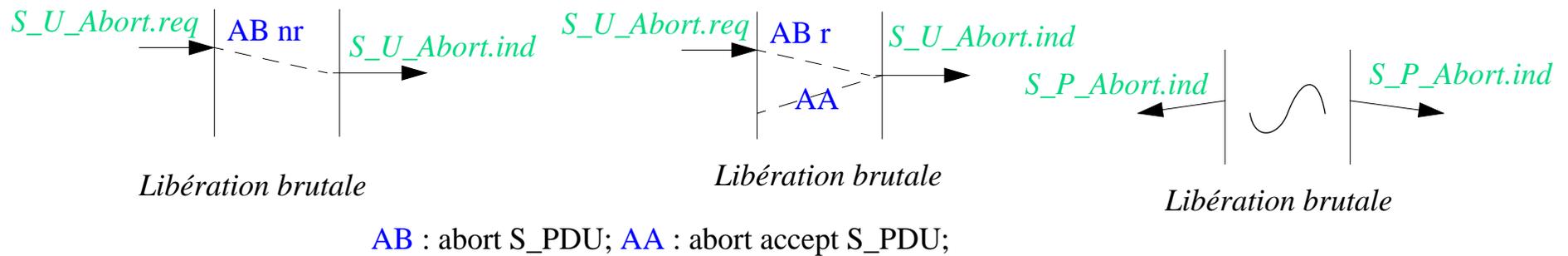


- . afin de permettre la terminaison ordonnée des données contenues par les S\_Release PDU

### 2.3.3 Libération brutale ou rupture

La libération brutale (“abort”) :

- demandée à la suite d’une erreur ou d’un incident
- n’est pas soumise à la possession des jetons
- n’importe quand, par n’importe qui (et même simultanément!)
- utilise le flux exprès de la connexion Transport, si disponible

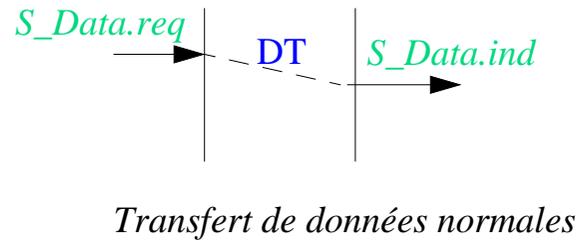


## 2.4. Transfert de données

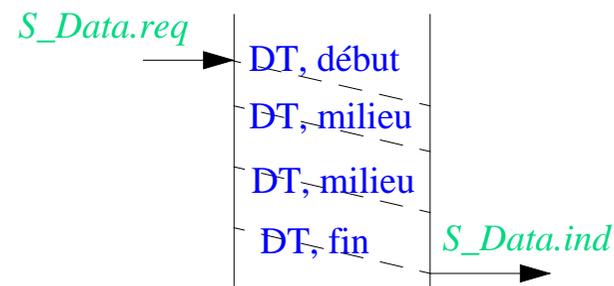
### 2.4.1 Transfert de données normales

#### Segmentation

- item de délimitation (“enclosure item”) : début, milieu, fin
- S\_SDU potentiellement de taille infini



DT: data S\_PDU;



#### Concaténation:

- règles spécifiques assurant une séparation non ambiguë

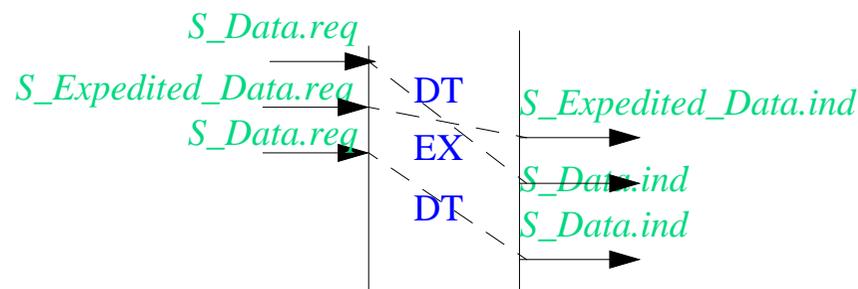
La connexion Transport assure un transfert des données parfaitement fiable :

- Pas de mécanismes de protection contre les erreurs
- Pas de multiplexage, ni d'éclatement de connexion

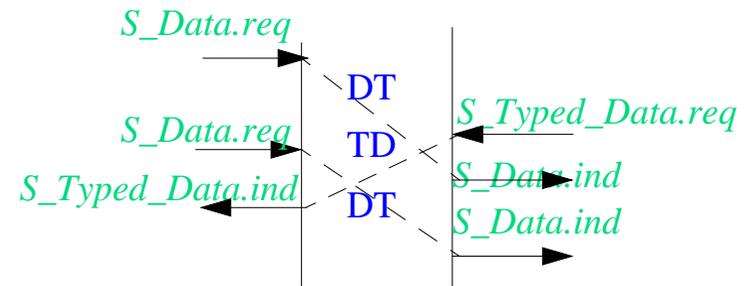


□ Données exprès :

- taille maximum : 14 octets
- similaire aux données exprès de la couche Transport
- non soumis au tour de parole (contraintes des jetons)
- non soumis au séquençement des données normales :
  - . les “Expedited SPDU” sont transportées dans des “Expedited TPDU”, si ce service est disponible.



*Transfert de données exprès*



*Transfert de données typés*

**EX:** expedited S\_PDU; **TD:** typed data S\_PDU;

## 2.5. La synchronisation

### 2.5.1 Points de synchronisation

☐ Ce service est utilisé pour effectuer des **reprises sur erreurs** et remettre les entités de Session dans un état (cohérent) connu des interlocuteurs.

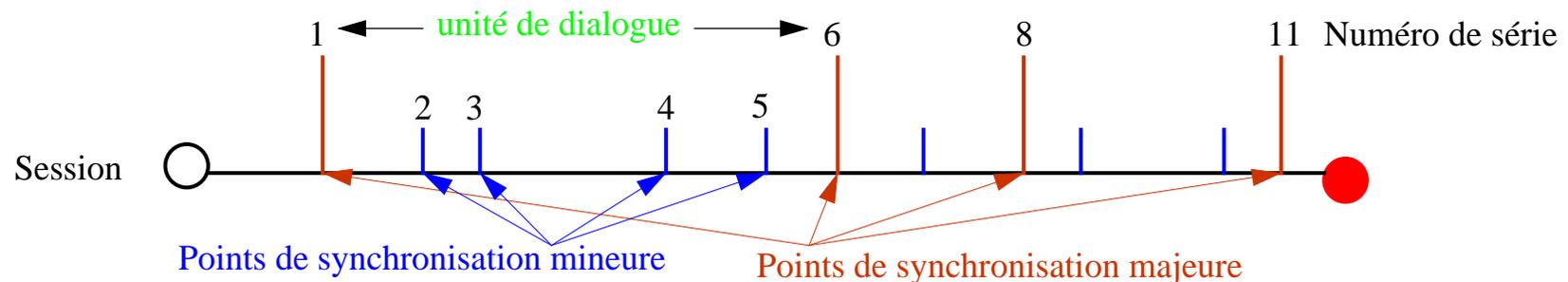
☞ Points de synchronisation :

- délimitent des unités logiques,
- identifiés par un numéro de série
  - . de 0 à 999 999 (sans modulo !)
  - . par incrémentation
- la signification réelle des points de synchronisation est du domaine de l'application.

Remarque : Les S\_PDU ne sont pas numérotées individuellement (la fiabilité de la connexion Transport est suffisante !)

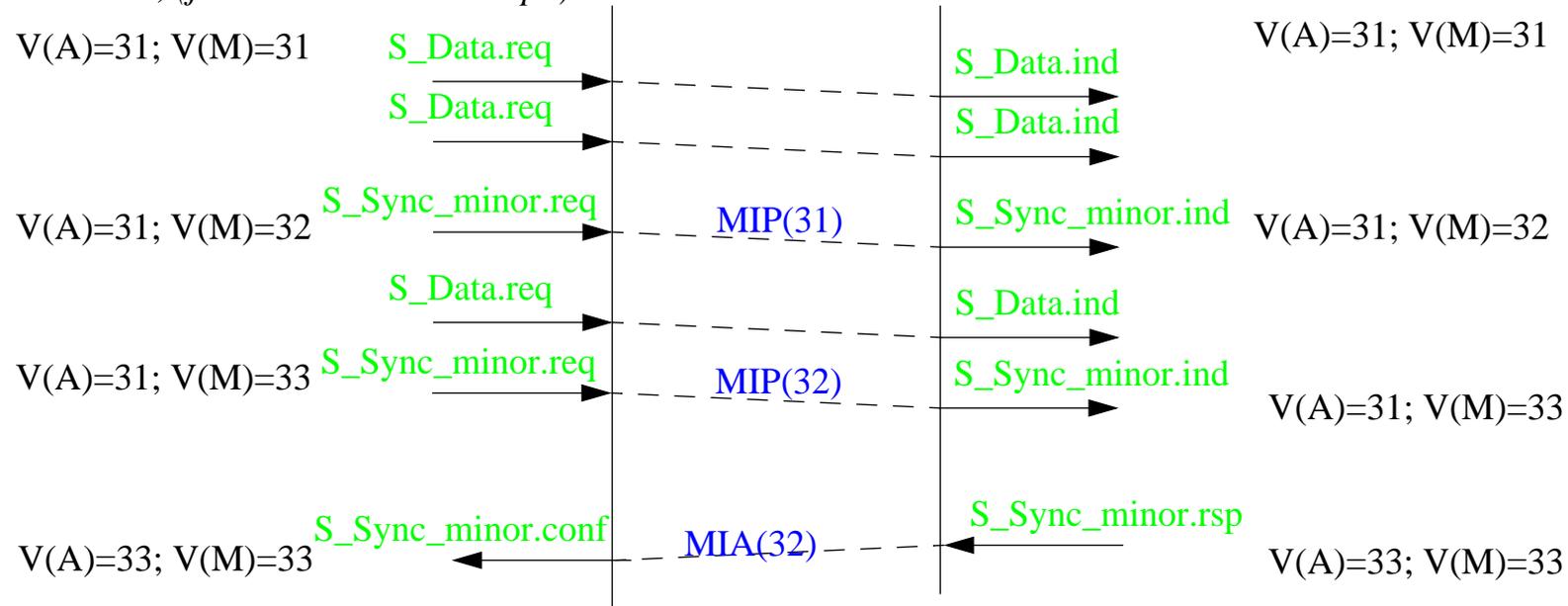
□ Deux types de points de synchronisation :

- les points de **synchronisation majeure**
    - . frontière étanche : attente de la confirmation explicite de la pose du point de synchronisation
    - . définit une **unité de dialogue**
  - les points de **synchronisation mineure**
    - . l'envoi des données peut se poursuivre
    - . confirmation non obligatoire
    - . un certain nombre de points de synchronisation mineure peuvent être en cours de pose sans que cela n'entraîne l'interruption de l'échange de données :
- ☞ mécanisme similaire au mécanisme de la fenêtre coulissante (envoi anticipé)



## 2.5.2 Primitives et scénarios

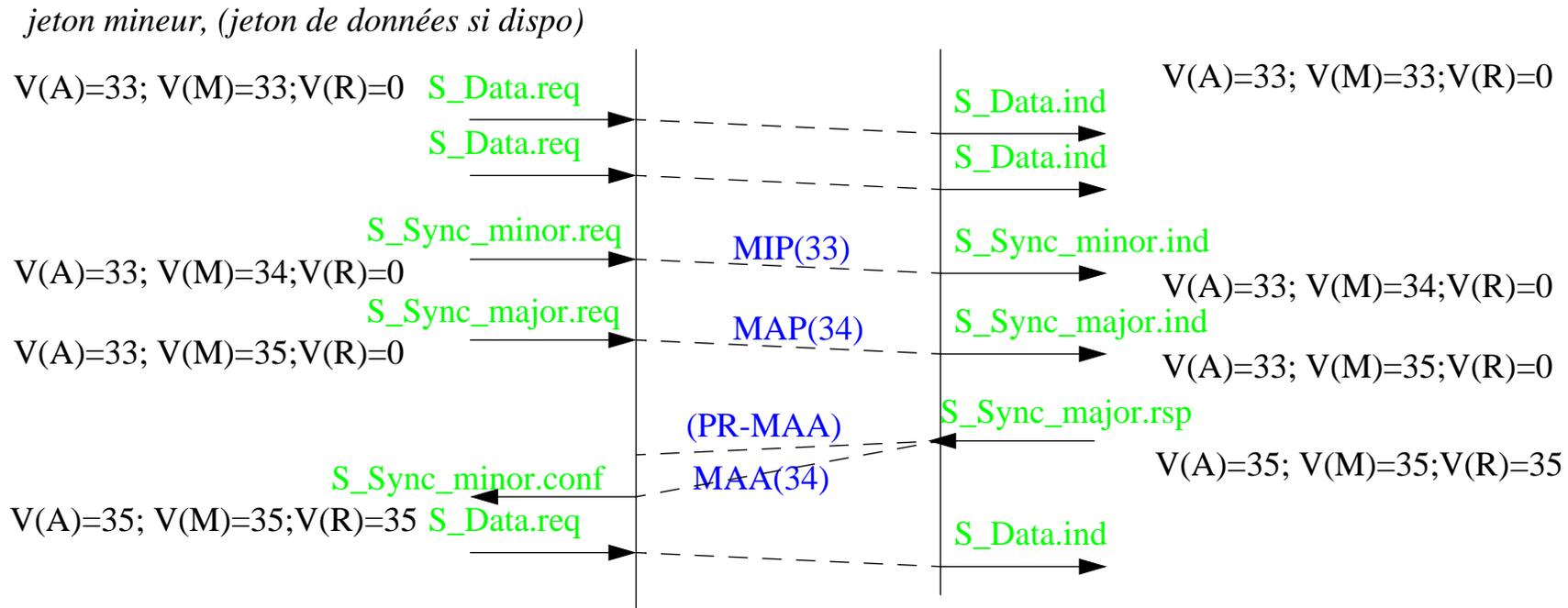
*jeton mineur, (jeton de données si dispo)*



**MIP** : Minor synchronization point SSPU (numéro de série); **MIA** : Minor sync. acknowledgment SSPU(ns)

*Pose de points de synchronisation mineure*

- V(A) : prochain numéro de série attendu
- V(M) : numéro de série courant



**MAP** : Major synchronization point SSPU (numéro de série); **MAA** : Major sync. acknowledgment SSPU(ns)  
**PR-MAA** : Prepare major sync. acknowledgment SSPU

*Pose d'un point de synchronisation majeure*

□ **V(R)** : plus grand numéro de série possible pour une resynchronisation par reprise

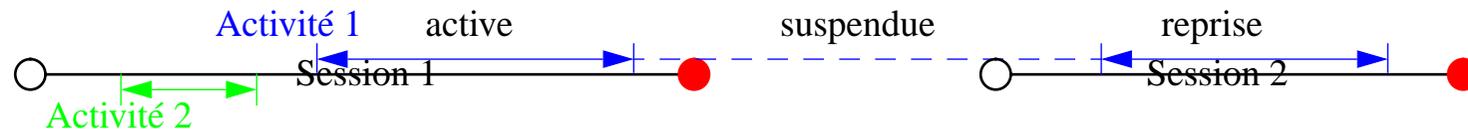
□ PR-MAA S\_SPU :

- utilise le flux exprès de Transport, si le service existe; sinon la S\_PDU n'est pas utilisée.
- prépare le traitement de l'acquittement de pose de point de synchronisation majeure
- délimite totalement la frontière avant et après la pose de point de synchronisation

## 2.6. Les activités

### 2.6.1 Présentation

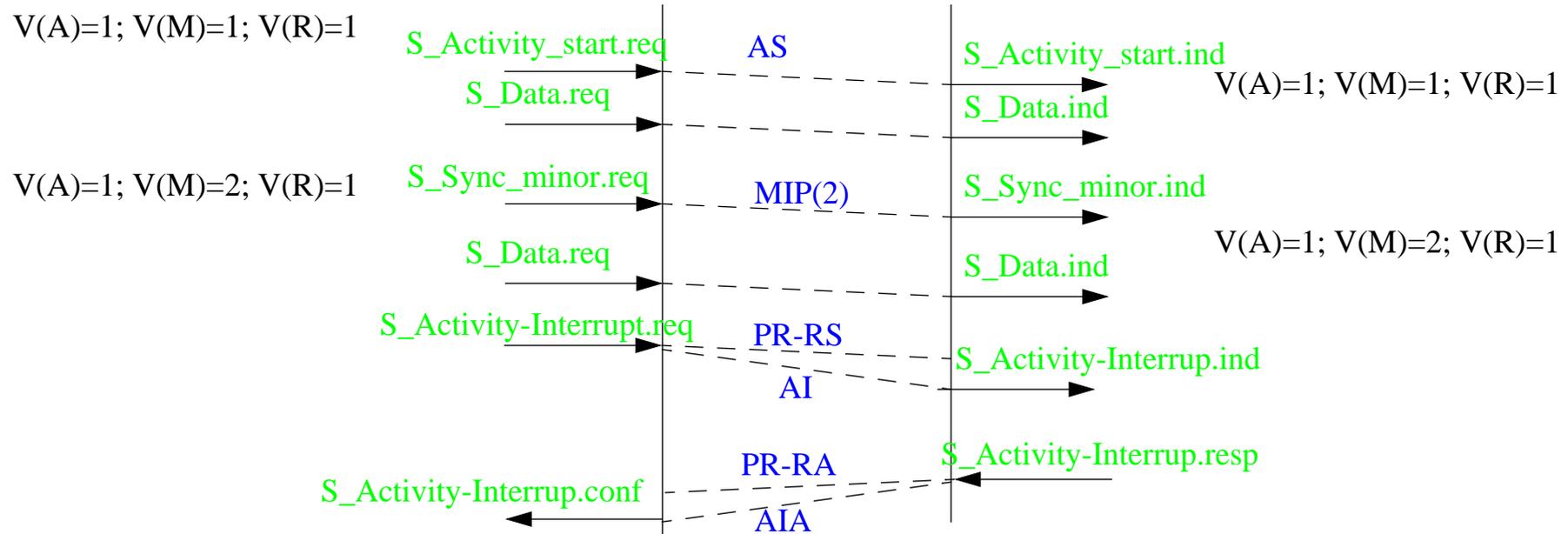
- ❑ Le flot de messages est structuré en **activités** :
  - elles sont identifiées de manière unique,
  - elles sont délimitées par des points de synchronisation majeure,
  - elles peuvent regrouper plusieurs unités de dialogue,
  - elles peuvent être suspendues et reprises aux points de synchronisation
  - elles sont totalement indépendantes les unes des autres
- ❑ Plusieurs activités successives au sein d'une même session.
- ❑ Une même activité peut se poursuivre sur plusieurs sessions



- ❑ La gestion d'activité est subordonnée à la possession du *jeton d'activité* ( $\equiv$  synchro majeure), et, s'ils sont disponibles, du jeton de données et du jeton de synchro mineure.

## 2.6.2 Lancement et interruption d'une activité

*jeton majeur/activité  
(jeton mineure et jeton de données si dispo)*



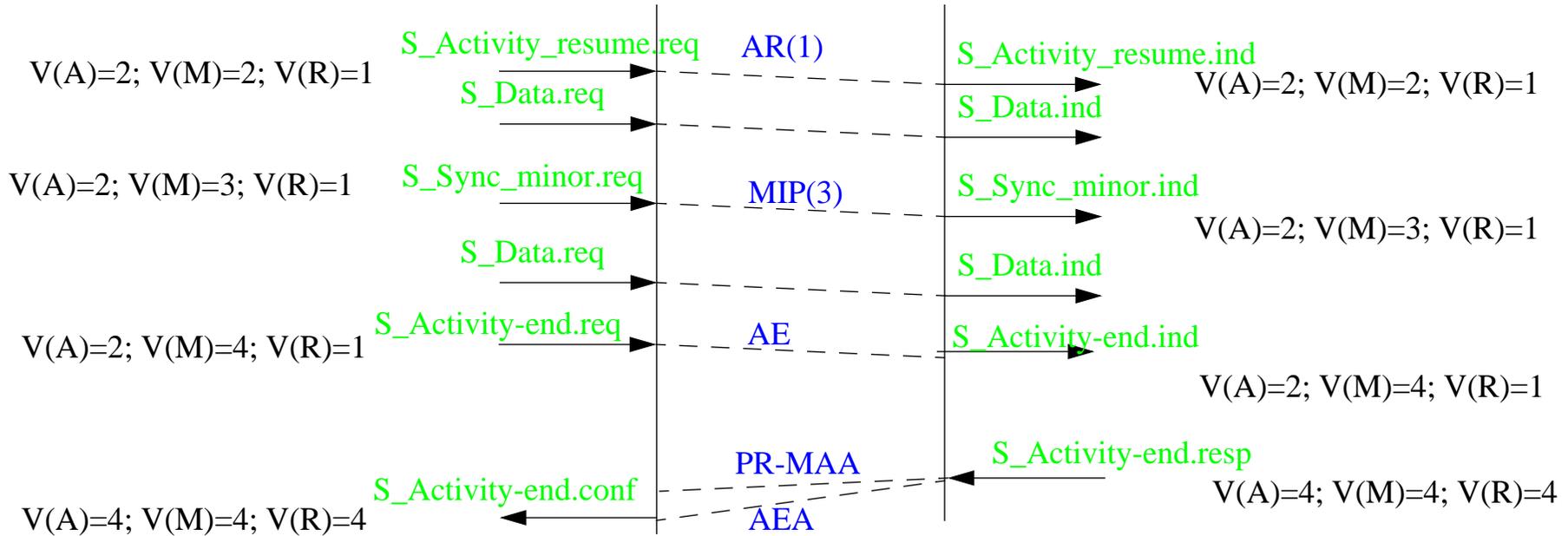
**AI** : Activity Interrupt SPDU (raison) ; **AIA**: AI. acknowledgment SPDU  
**AS** : Activity Start SPDU

*Lancement d'une activité et son interruption*

La S\_PDU AS comporte un champ de données d'au plus 6 octets.

### 2.6.3 Reprise et terminaison d'une activité

*jeton majeur/activité  
(jeton mineur et jeton de données si dispo)*



**AR** : Activity Resume SSPU (numéro de série) ; **AIA**: Activity end acknowledgment SPDU  
**AS** : Activity End SPDU; **PR-MAA** : Prepare major synchro point SPDU

*Lancement d'une activité et son interruption*

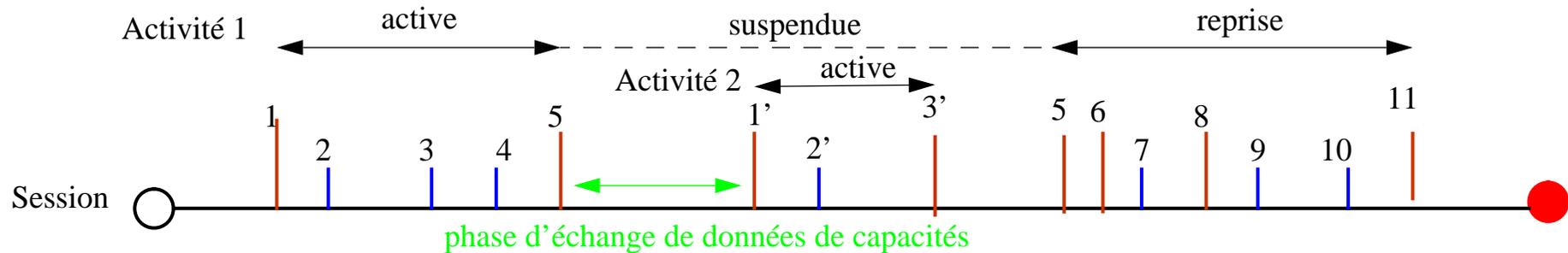
## 2.6.4 Transfert de données hors activité

Entre activité, il est possible d'échanger des données :

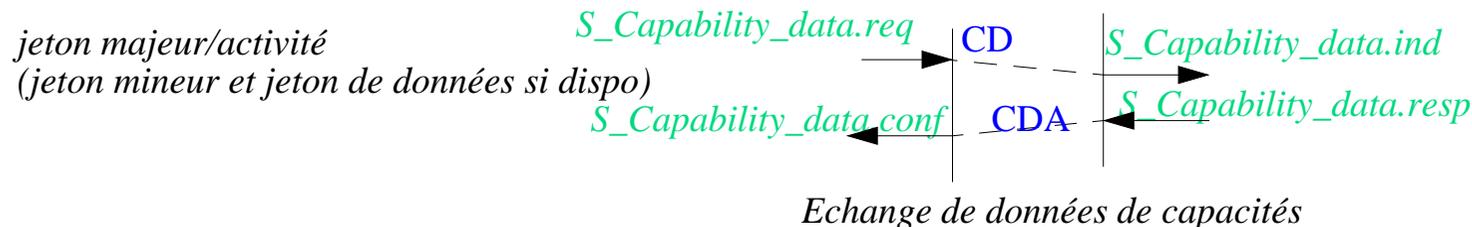
- données normales (mais sans poser des points de synchronisation)

- informations de capacités

- ☞ Capability Data



- unité fonctionnelle optionnelle : dépend de la présence de l'unité fonctionnelle de gestion d'activité et de la possession du jeton majeur (et mineur et données si dispo)
- transfert en quantité limitée avec acquittement explicite :



CD :capability data S\_PDU; CDA : capability data acknowledgment S\_PDU;

## 2.7. Resynchronisation

### 2.7.1 Fonctions :

- un nouveau numéro de série
- réattribution des jetons
- purge des données

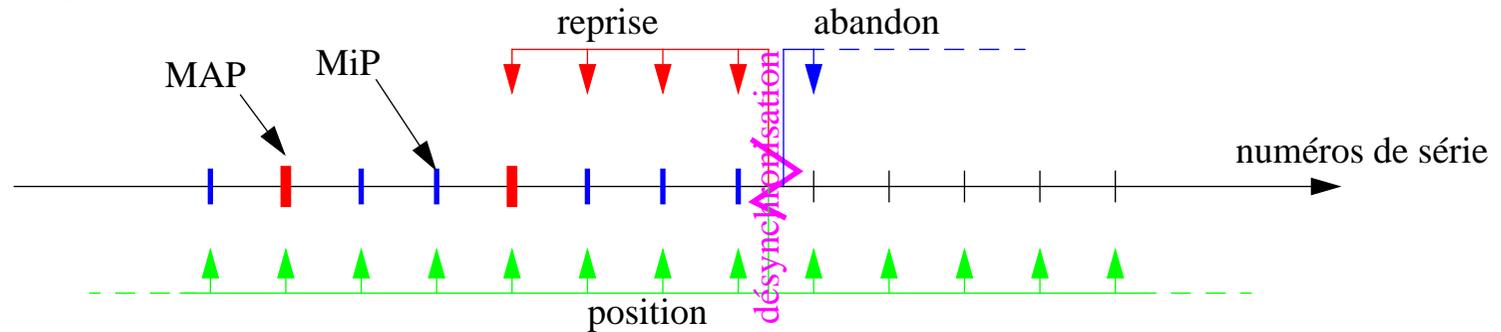
Ce service est offert par une unité fonctionnelle optionnelle :

- négocié lors de l'établissement

□ Trois types de resynchronisation sont possibles :

- **reprise** : “Restart”
  - . un point de synchronisation posé (inférieur au numéro de série courant, supérieur au dernier point de synchronisation majeure)
- **abandon** : “Abandon”
  - . un nouveau point de synchronisation (supérieur au numéro de série courant, généralement le suivant)
  - . certaines données sont irrémédiablement perdues
- **position** : “Set”
  - . un point de synchronisation quelconque !

. Exemple :



La signification des points de synchronisation et la gestion de la resynchronisation est du domaine de l'application. Par exemple :

- l'application peut conserver une copie des données échangées tant que les points de synchronisation majeure ne sont pas confirmés, en cas de problème il sera possible de restaurer l'échange après reprise à l'aide des copies conservées.



## 2.8. Les unités fonctionnelles

**Tableau 1 : Les unités fonctionnelles**

Unité fonctionnelle	Services	Primitives	Paramètres
Noyau	Etablissement de la connexion	S_connect.req (.ind) S_connect.resp (.conf)	Ident. de connexion, adresses source et dest., résultat, QoS, proposition de l'U.F., numéro initial de série, attribution des jetons, données
	Transfert de données normales Terminaison normale	S_data.req (.ind) S_release.req (.ind) S_release.resp (.conf)	données données réponse, données
	Rupture de la connexion	S_U_abort.req (.ind) S_P_abort.ind	données raison
Terminaison négociée	Gestion des jetons	S_token-give.req (.ind) S_token-please.rep (.ind)	jetons, données jetons, données
Transmission à l'alternat	Gestion des jetons	cf au-dessus	
Transfert de données exprès	Transfert de données exprès	S_expetited-data.req (.ind)	données
Transfert de données typées	Transfert de données typées	S_typed-data.req (.ind)	données
Echange d'info. de capacités	Echange d'information de capacités	S_capability-data.req (.ind) S_capability-data.resp (.conf)	données données
Synchronisation mineure	Synchronisation mineure	S_sync-minor.req (.ind) S_sync-minor.resp (.conf)	type, numéro de série, données numéro de série, données
	Gestion des jetons	cf au-dessus	

**Tableau 1 : Les unités fonctionnelles**

Unité fonctionnelle	Services	Primitives	Paramètres
Synchronisation majeure	Synchronisation majeure Gestion des jetons	S_sync-minor.req (.ind) S_sync-minor.resp (.conf) cf au-dessus	numéro de série, données numéro de série, données
Resynchronisation	Resynchronisation	S_resynchronize.req (.ind) S_resynchronize.resp (.conf)	type, numéro de série, jetons, données numéro de série, jetons, données
Gestion d'activité	Lancement d'activité Interruption d'activité  Reprise d'activité  Abandon d'activité  Terminaison d'activité  Gestion des jetons Gestion du contrôle	S_activity-start.req (.ind) S_activity-interrupt.req (.ind) S_activity-interrupt.resp (.conf) S_activity-resume.req (.ind)  S_activity-discard.req (.ind) S_activity-discard.resp (.conf) S_activity-end.req (.ind) S_activity-end.resp (.conf) cf au-dessus S_control-give.req(.ind)	ident. d'activité, données raison, données données ident. d'activité, numéro de série, ident. de l'ancienne connexion, données raison, données données numéro de série, données données données
Signalisation d'anomalie	Signalisation d'anomalie	S_U-exception-report.req (.ind) S_P-exception-report.ind	raison raison, données

### 3. Protocole

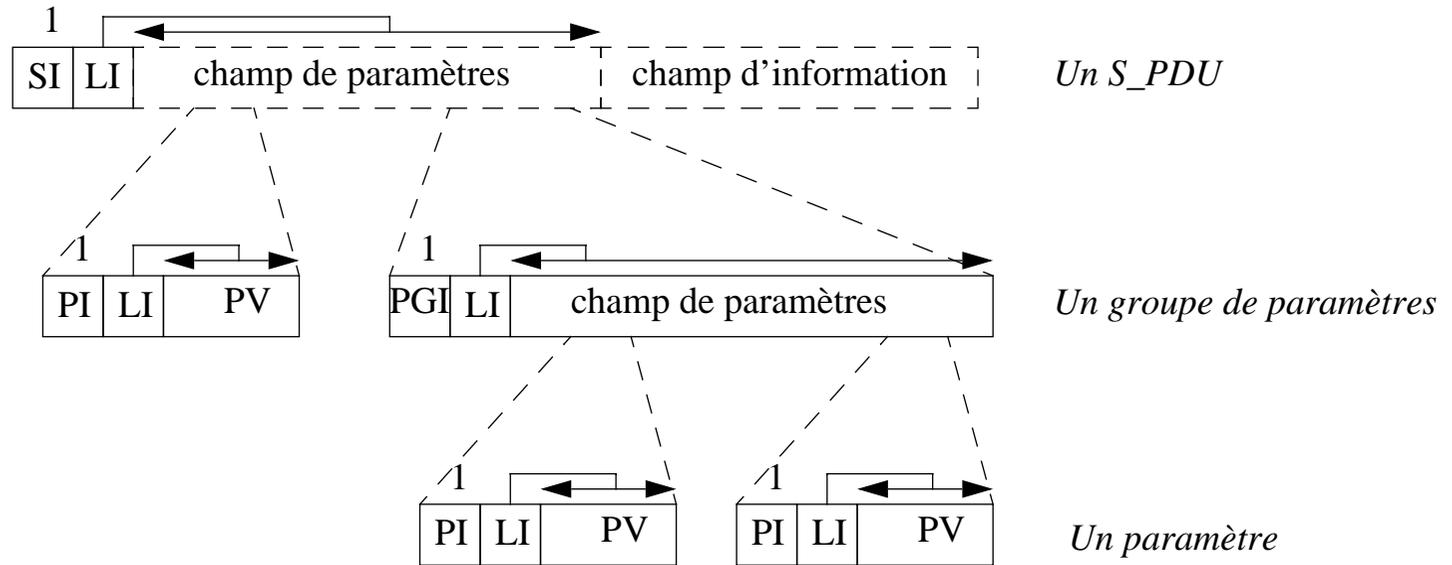
#### 3.1. Le format général des S\_PDU

Format variable, dont la longueur est multiple d'un octet

Codage TLV généralisé

- ❑ Champ SI (1 octet), "S\_PDU identifier" : identifie le S\_PDU
- ❑ Champ LI, "Length Indicator" : longueur en octet du champ paramètre
  - longueur < 255 (1 octet)
  - longueur < 65536 (3 octets) : premier octet = 255; deuxième et troisième octet=longueur
- ❑ Champ de paramètres :
  - . une suite de paramètres pouvant être rassemblés par groupes.
- ❑ Champ d'information :
  - . de longueur variable
  - . sans indication explicite de longueur

□ Quelques formats :



Champ PI : “Parameter identifier”

Champ PGI : “Parameter group identifier”

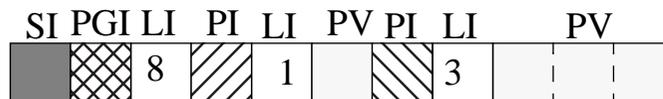
□ Exemples :



S\_PDU avec 1 paramètre de 3 octets



S\_PDU de données sans paramètre



S\_PDU avec 1 groupe de 2 paramètres de 1 et 3 octets

### 3.2. Les S\_PDU

**Tableau 2 : S\_PDU**

Symboles	Noms	Services
CN AC RF	Connect Accept Refuse	Etablissement de la connexion
FN DN NF	Finish Disconnect Not finished	Terminaison de la connexion
DT	Data	Transfert de données normales
EX	Expedited data	Transfert de données exprès
TD	Typed data	Transfert de données typées
CD CDA	Capability data Capability data ack.	Echange d'information de capacités
AB AA	Abort Abort accept	Rupture de la connexion
GT PT	Give tokens Please tokens	Gestion des jetons
MIP MIA	Minor synchronization point Minor ack.	Pose de point de synchro. mineure
MAP MAA	Major synchronization point Major ack.	Pose de point de synchro. majeure

**Tableau 2 : S\_PDU**

Symboles	Noms	Services
RS RA	Resynchronize Resynchronize ack.	Resynchronisation
AS AR AI AIA AD ADA AE AEA	Activity start Activity start ack. Activity interrupt Activity interrupt ack. Activity discard Activity discard ack. Activity end Activity end ack.	Gestion d'activité
PR-MAA PR-RA PR-RS	Prepare	Flux exprès
GTC GTA	Give tokens confirm Give tokens ack.	Passation du contrôle
ER ED	Exception report Exception data	Signalisation d'anomalie

### 3.3. La concaténation

□ La concaténation permet le transport de plusieurs S\_PDU au sein une seule T\_SDU.

Certaines S\_PDU sont obligatoirement concaténées !

Les S\_PDU sont réparties en trois catégories :

**Catégorie 0 :**

- GT et PT S\_PDU

**Catégorie 1 :** les S\_PDU qui ne sont pas normalement suivies par un transfert de jetons :

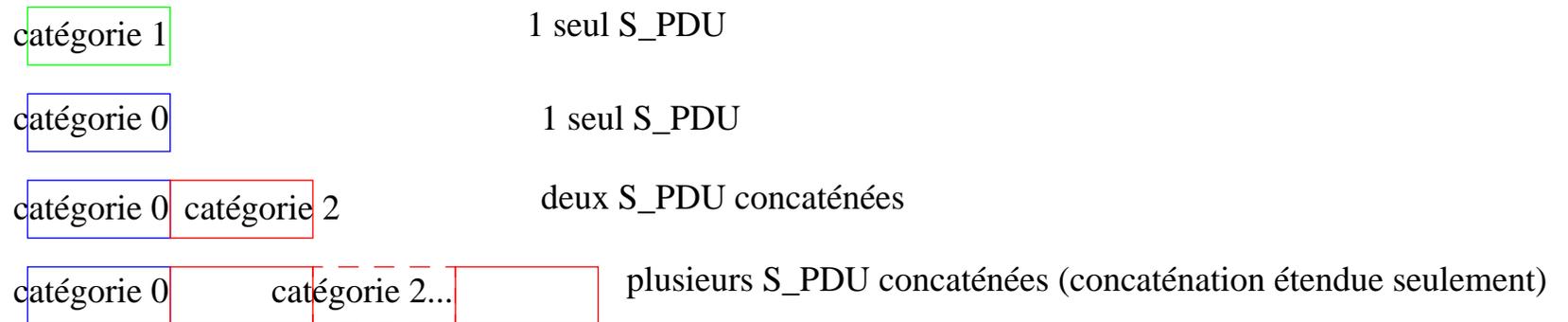
- celles d'établissement de la connexion, de données exprès, de terminaison, ...

**Catégorie 2 :** les S\_PDU qui peuvent être suivies par un transfert de jetons :

- celles de données, d'accusé de pose de synchronisation majeure, d'échange de capacités, ...

□ Règles de concaténation :

- catégorie 1 : toujours seule
- catégorie 2 : toujours précédées d'une S\_PDU de catégorie 0
- catégorie 0 : soit seule, soit suivie d'une (concaténation normale) ou jusqu'à trois (concaténation étendue) S\_PDU de catégorie 2.



Lors d'un échange d'une S\_PDU de catégorie 2, si aucun transfert de jeton n'est prévu, une S\_PDU de catégorie 0 (GT ou PT) sans paramètre est concaténée.

L'option "transfert avec optimisation du dialogue" négociée lors de l'établissement de la connexion permet à plusieurs S\_PDU de catégorie 2 d'être concaténées (3 maximum).

## 4. Conclusion

La couche Session définit et utilise de nombreux objets évolués :

- activité, dialogue, point de synchronisation, jetons
- pour répondre aux besoins de structuration et de resynchronisation

La couche Session définit de nombreux services :

- regroupés en unité fonctionnelle et négociées lors de l'établissement de la connexion
- pour répondre aux différents besoins des entités supérieures

Ces services mettent à la disposition des applications des outils (de base) dont l'utilisation et l'interprétation exactes sont laissées aux entités supérieures.

Il existe une définition de la couche Session en mode non connecté : ISO 9548

En fait le protocole de la couche Session

- est peu utilisé,
- mais comporte des mécanismes originaux adaptés :
  - . gestion de transactions, maintien de la cohérence, etc.

Dans la version 1 du protocole, des profils (ensemble d'unités fonctionnelles) ont été définis :

- Basic Synchronized Subset :
  - . noyau + terminaison négociée + transmission à l'alternat + données typées + synchronisation majeure + synchronisation mineure + resynchronisation.
- Basic Activity Subset :
  - . noyau + transmission à l'alternat + données typées + info. de capacités + synchronisation mineure + activité + anomalie
- Basic Combined Subset :
  - . noyau + transmission à l'alternat