

# Architecture de Protocoles: Modèle en Couche

A. S. Hafid

E-mail: [ahafid@iro.umontreal.ca](mailto:ahafid@iro.umontreal.ca)

Phone: (514) 343-2446

# Plan

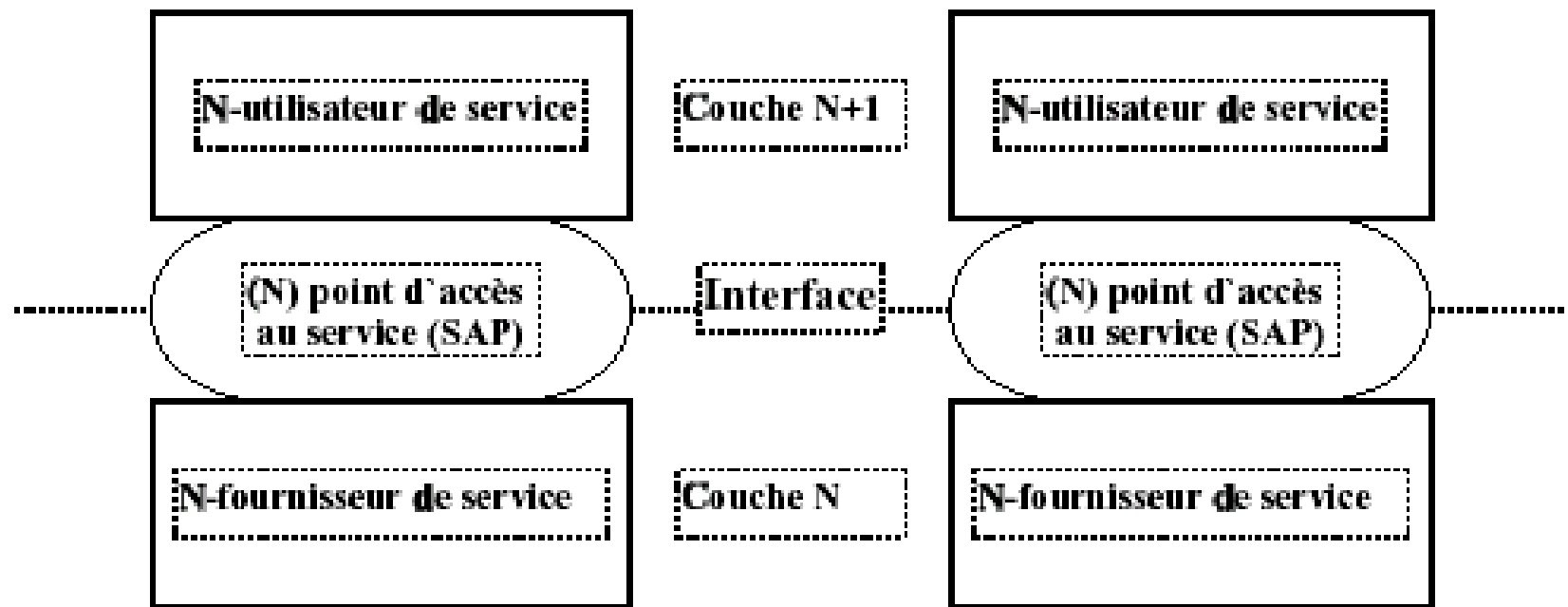
- Principes du modèle à couche
- Fonctionnalités
- Types de primitives utilisées
- Service sans connexion Vs. Service orienté connexion
- Service fiable Vs. Service non fiable
- Modèle de référence OSI
- Modèle de référence TCP/IP
- Conclusion

# Principes du Modèle?

# Principes du Modèle

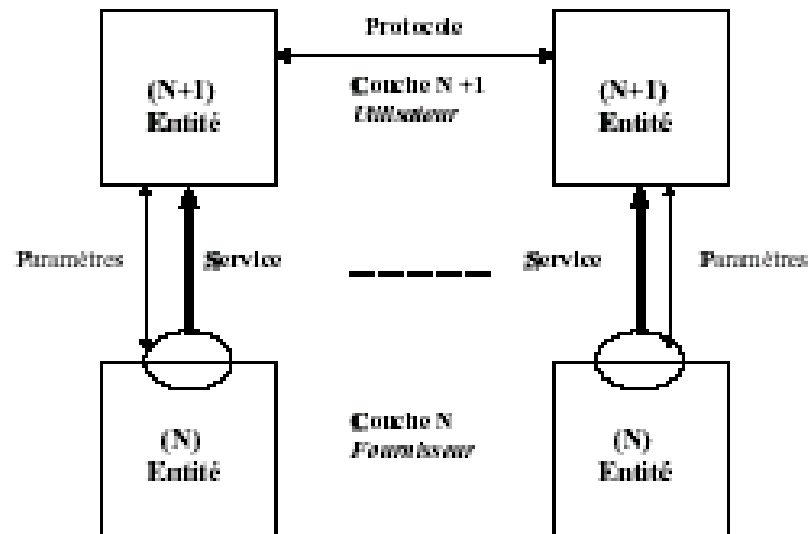
- Un niveau du modèle par niveau d'abstraction
- A chaque niveau correspond une fonction bien définie
- La fonction de chaque niveau est choisie en ayant en tête la définition de standards internationaux
- Les frontières entre niveaux sont choisies pour minimiser les flux d'informations entre niveaux
- Le nombre de niveaux est choisi pour correspondre aux différentes fonctions mais ne doit pas être trop grand parce que le système serait trop lourd

# Interactions entre Couches Voisines



L'interaction entre fournisseur et utilisateur d'un service de deux couches voisines

# Relations Entre Entités



*Relation entre entités*

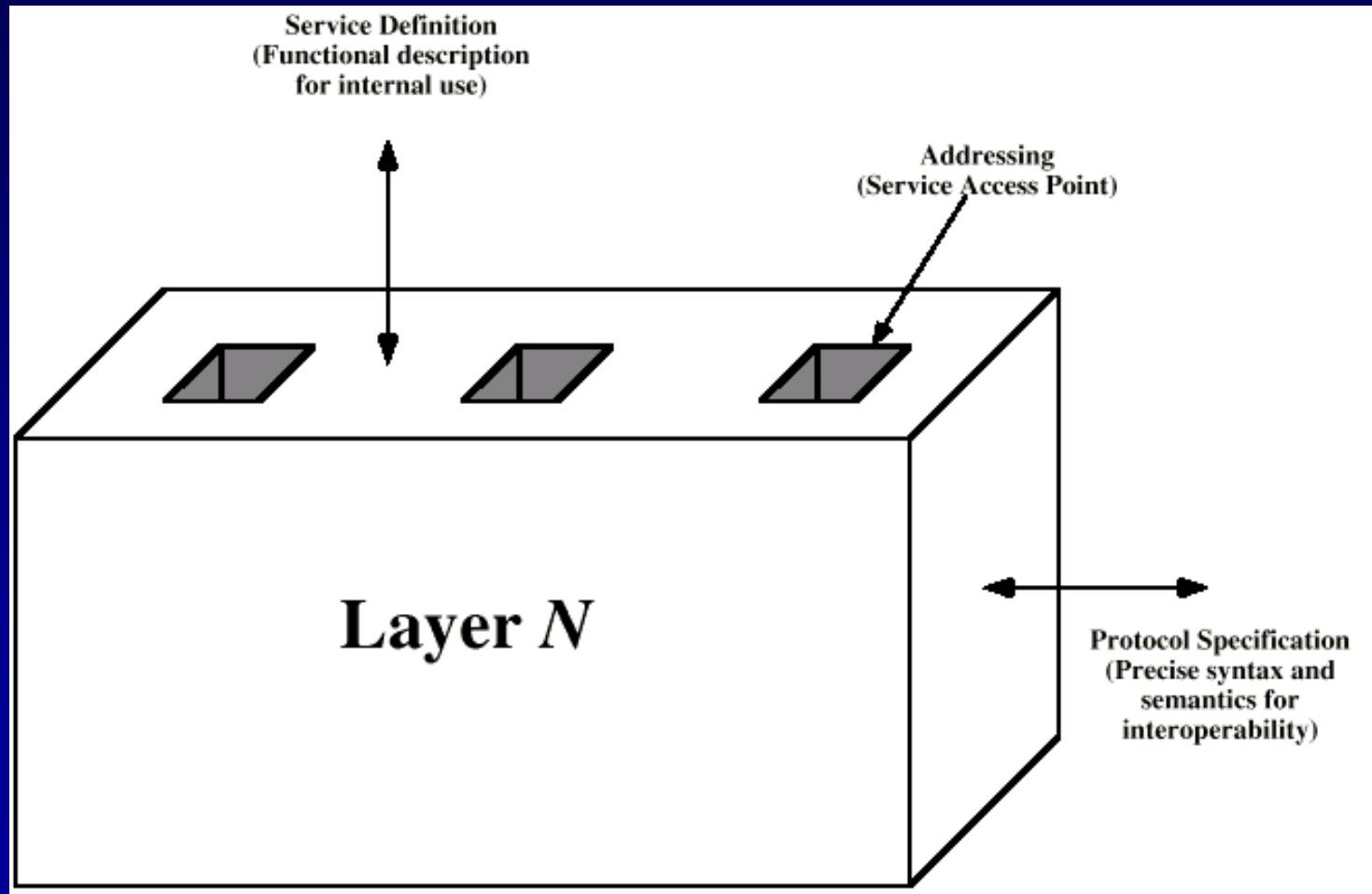
**Entités:** éléments actifs (processus)

**Entités pairs:** les entités de la même couche sur différentes machines (peer entities)

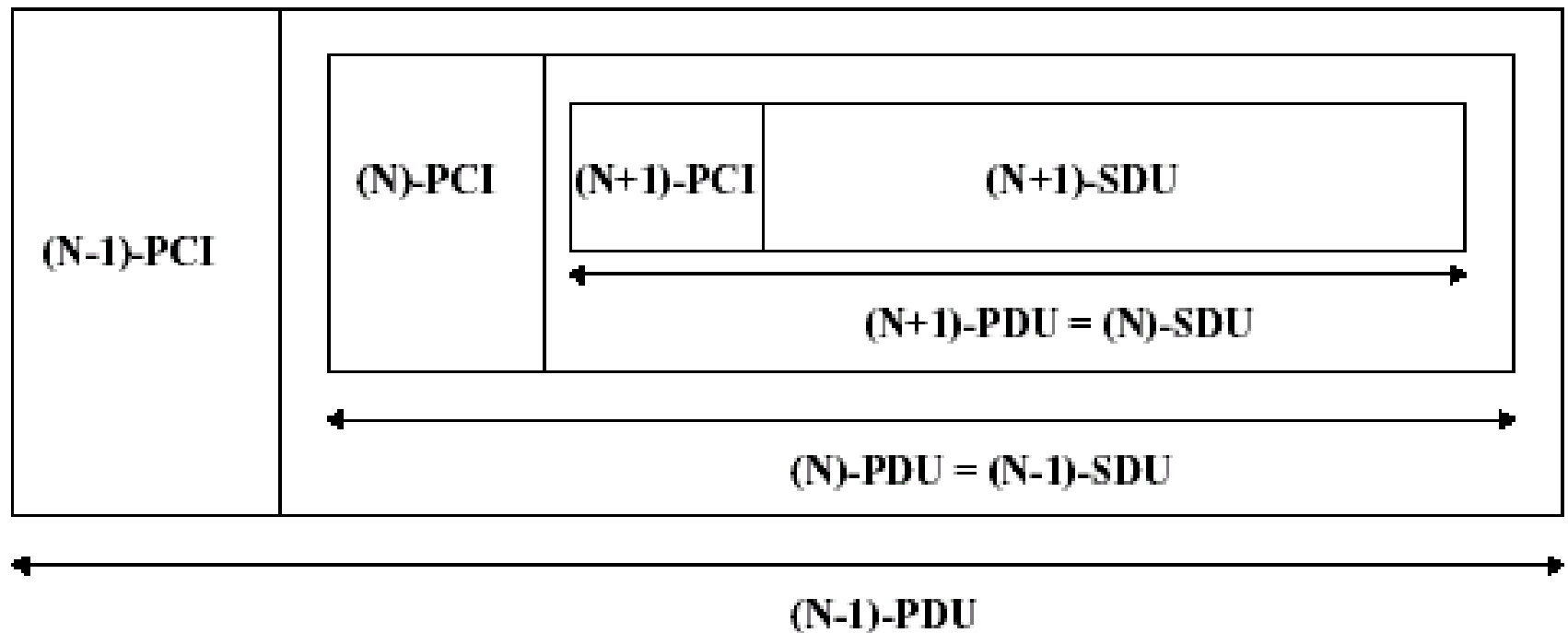
**Protocole:** règles de communication entre entités paires

**Communication logique ou virtuelle:** communication aux couches supérieures de la couche physique

# Une Couche



# PDU de Différentes Couches





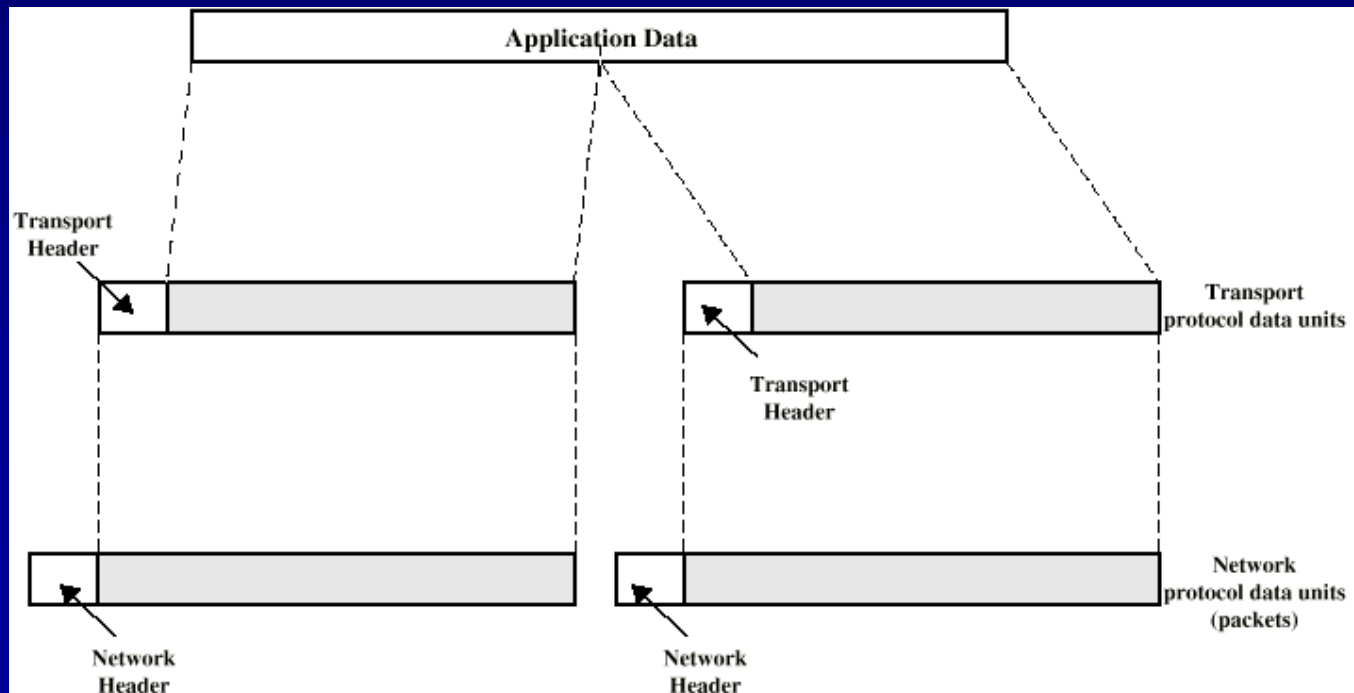
Fonctions?

# Fonctions

- Encapsulation
- Segmentation et assemblage
- Contrôle de connexion
- livraison Ordonnée
- Contrôle de flux
- Contrôle d'erreur
- Adressage
- Multiplexage
- Services de Transmission

# Encapsulation

- Ajout d'information de contrôle
  - Adresses
  - Code de détection d'erreurs
  - Etc.



Grand paquet vs. petit paquet?

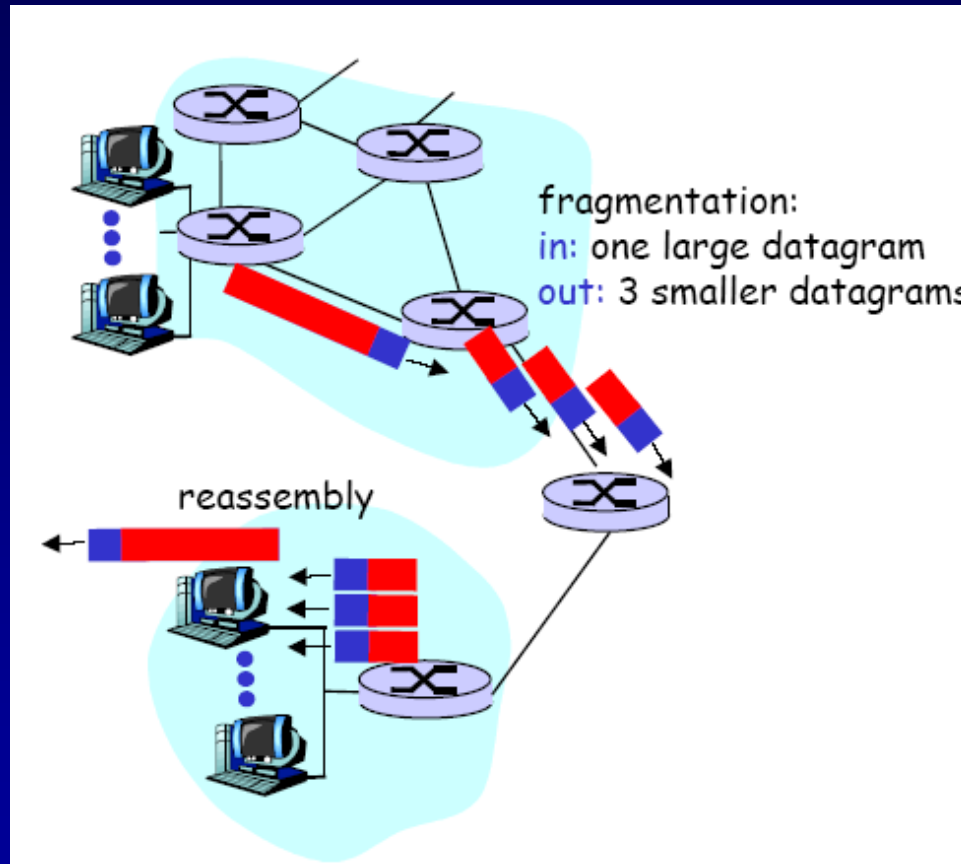
# Segmentation (Fragmentation)

- Les blocs de données sont d'une taille déterminée
- Les messages de la couche application peuvent grands
- Les paquets réseaux peuvent être petits
- Segmentation consiste à couper des blocs de données qui sont grands en plus petits blocs
  - Fragmentation dans le modèle TCP/IP
  - Les trames Ethernet ne peuvent pas dépasser 1526 octets

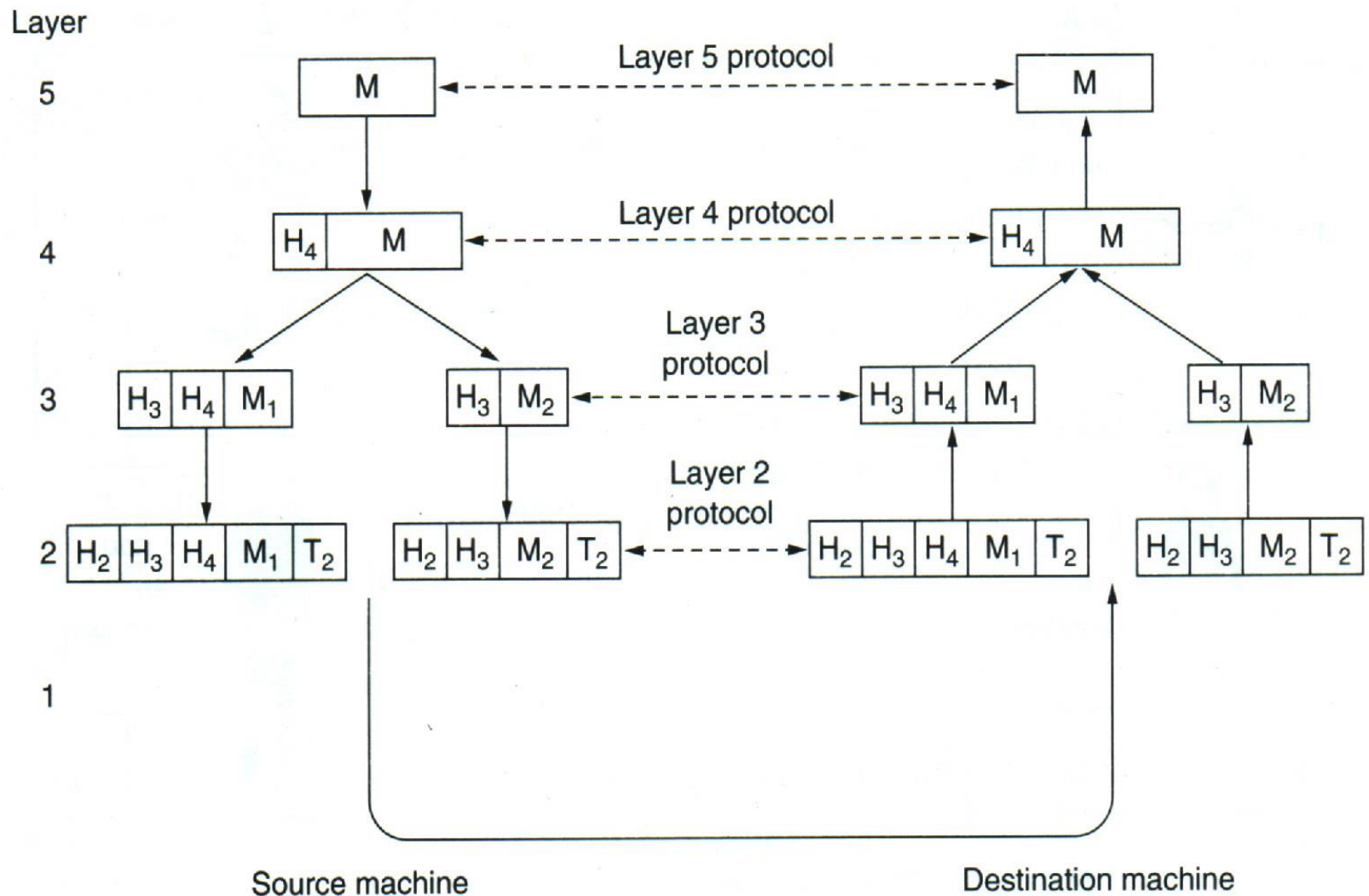
# Segmentation (Fragmentation): Exemple

Protocole de liaison de données	MTU (octets)
Ethernet	1500
IEEE 802.3	1492
Bus à jeton	8191
Anneau à jeton	5000 (typique)
FDDI	4500
PPP	1500 par défaut (négociable)

# Segmentation (Fragmentation): Example



# Encapsulation, Fragmentation, et Assemblage: Exemple





# Contrôle de Connexion

- Établissement de connexion
- Transfert de données
- Terminaison de connexion

# Livraison Ordonnée

- PDUs peuvent traverser des chemins différents de la source à la destination
- PDUs peuvent ne pas arriver en ordre
- Comment?

# Livraison Ordonnée

- Numérotter les PDUs pour permettre la livraison ordonnée

# Contrôle de flux

- Fait par le receveur
- Limiter la quantité ou le taux de données
- Exemples de techniques?

# Contrôle de flux

- Techniques
  - Arrêter et attendre (stop and wait)
  - Systèmes à crédit
    - » E.g., Fenêtres coulissantes

# Contrôle d'erreur

- Protéger contre les pertes et les dommages
- Détection d'erreur?
- Détection de pertes?
- Exemples de solutions?

# Contrôle d'erreur

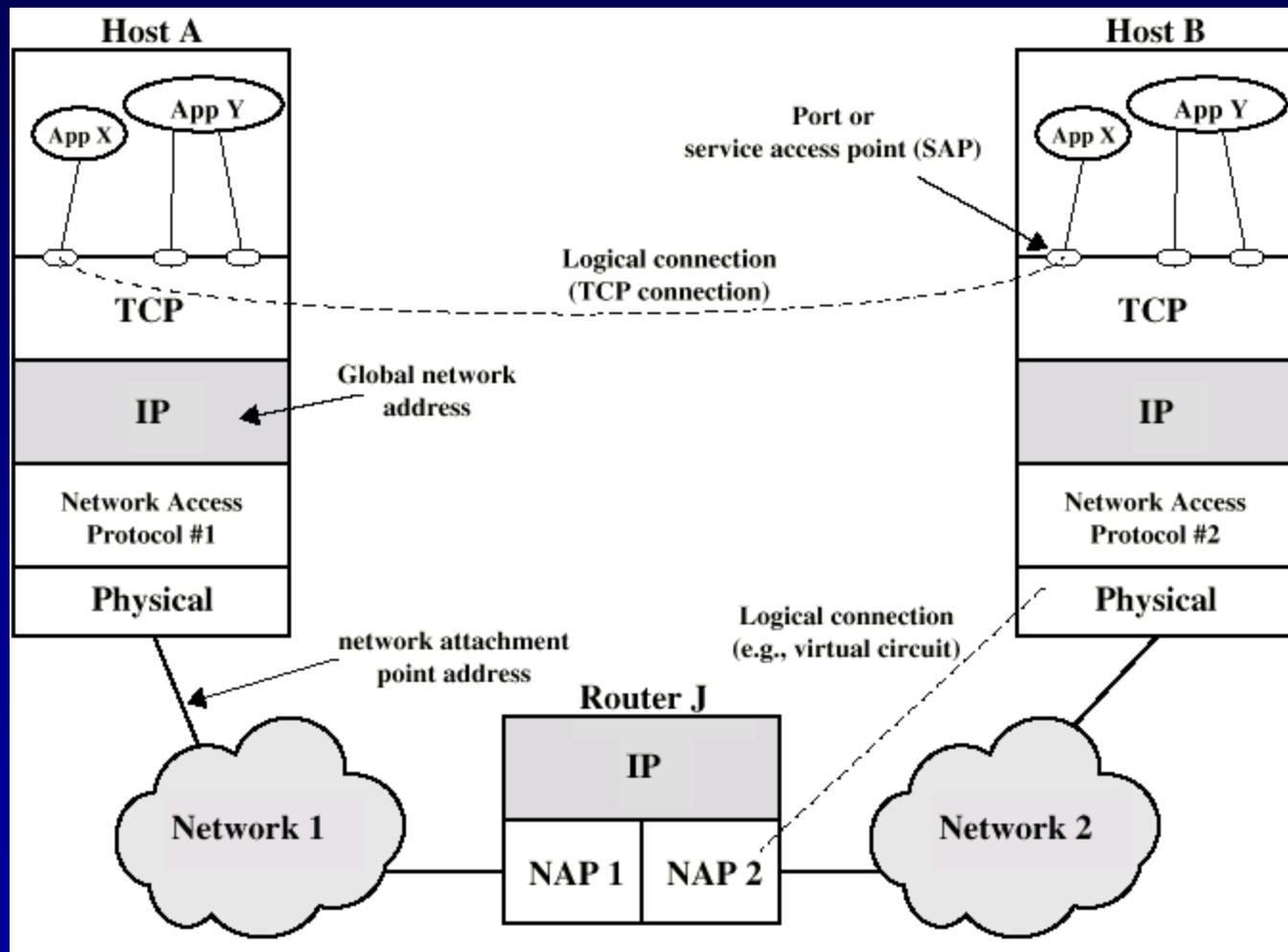
- Détection d'erreur
  - La source insert des bits de détection d'erreur
  - La destination vérifie ces bits
  - Si OK, confirme
  - Sinon, détruire le paquet
- Retransmission
- Fait à différents niveaux

# Adressage

- Une adresse unique pour chaque machine et chaque routeur (par interface)
- Adresse niveau réseau
  - Adresse IP (TCP/IP)
  - NSAP: Network Service Access Point (OSI)
- Processus dans la machine
  - Numéro de port (TCP/IP)
  - Service Access Point (OSI)



# Addressage (Cont.)



Combien de modes d'adressage ?

# Mode d'adressage

- En général une adresse identifie une seule machine
  - Adresse unicast
- Peut adresser toutes les machines dans un domaine
  - Adresse broadcast
- Peut adresser un sous ensemble de machines dans un domaine
  - Adresse multicast

# Multiplexage

- Correspondre plusieurs connexions dans un niveau à une seule connexion dans un autre niveau
  - Plusieurs faisceaux lumineux (lightpaths) dans une seule fibre optique
  - Plusieurs connexions voix sur un câble
  - Etc.

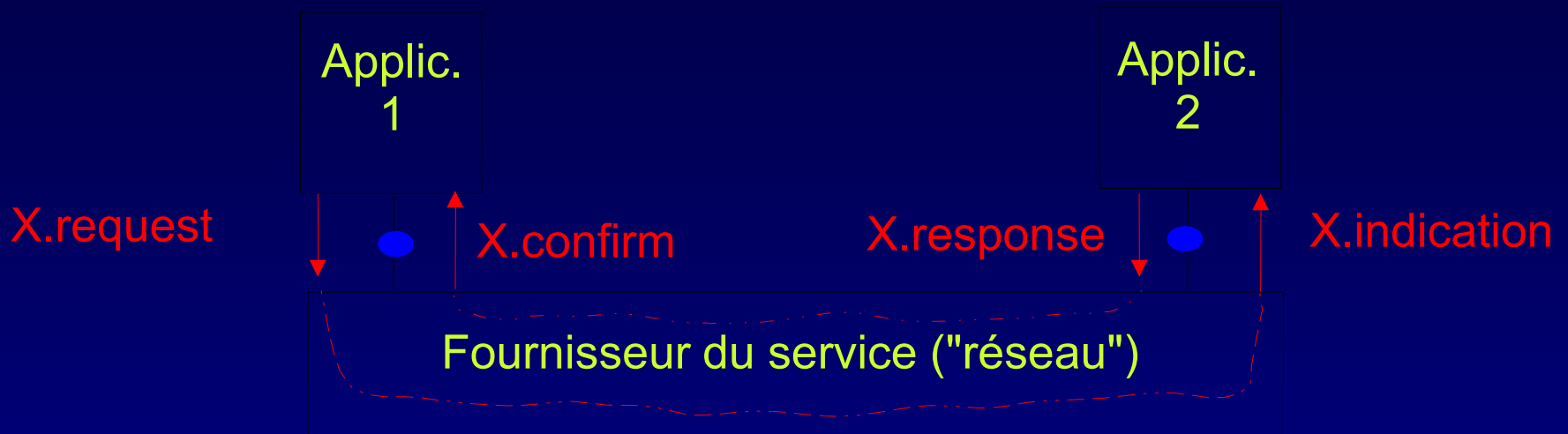
# Service de Transmission

- Priorité
  - P.ex., messages de contrôle
- Qualité de Service
  - délai minimal, maximiser le débit (*throughput*), maximiser la fiabilité
  - Diffserv vs. intserv
- Etc.

# Types de Primitives

- Primitive
  - représentation abstraite d'une interaction entre un utilisateur et son fournisseur de service
  - peut contenir des paramètres tels que
    - source
    - destination
    - message transmis (SDU ou Service Data Unit)

# Types de Primitives (Cont.)



## X.request

requête d'un utilisateur au fournisseur du service

## X.indication

primitive générée par le fournisseur du service à destination d'un utilisateur (souvent liée à une primitive X.request distante)

## X.response

primitive de réponse optionnelle à une primitive X.indication

## X.confirm

primitive générée par le fournisseur du service à destination d'un utilisateur suite à une primitive X.response distante

# Service sans Connexion

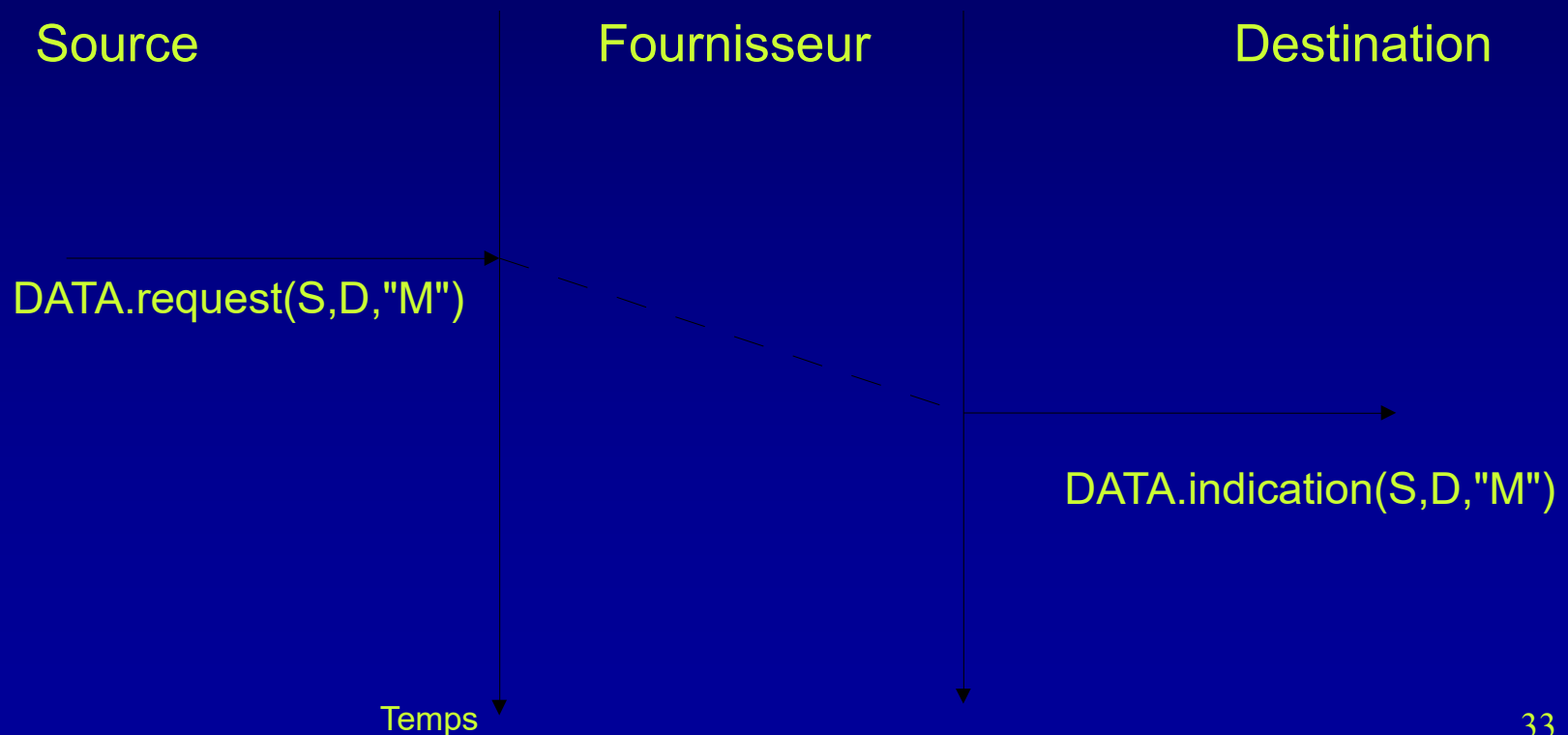
- Objectif
  - permettre à un émetteur d'envoyer rapidement un message à un (ou plusieurs) receveur
- Principe
  - L'émetteur confie son message au fournisseur du service via une primitive DATA.req
  - Le fournisseur du service achemine le message jusqu'au receveur et le délivre via une primitive DATA.ind
- Utilisation
  - utile pour envoyer de petits messages
  - exemple : service postal



# Service sans Connexion (Cont.)

- Primitives

- DATA.request(source, destination, SDU)
- DATA.indication(source, destination, SDU)



# Service sans Connexion (Cont.)

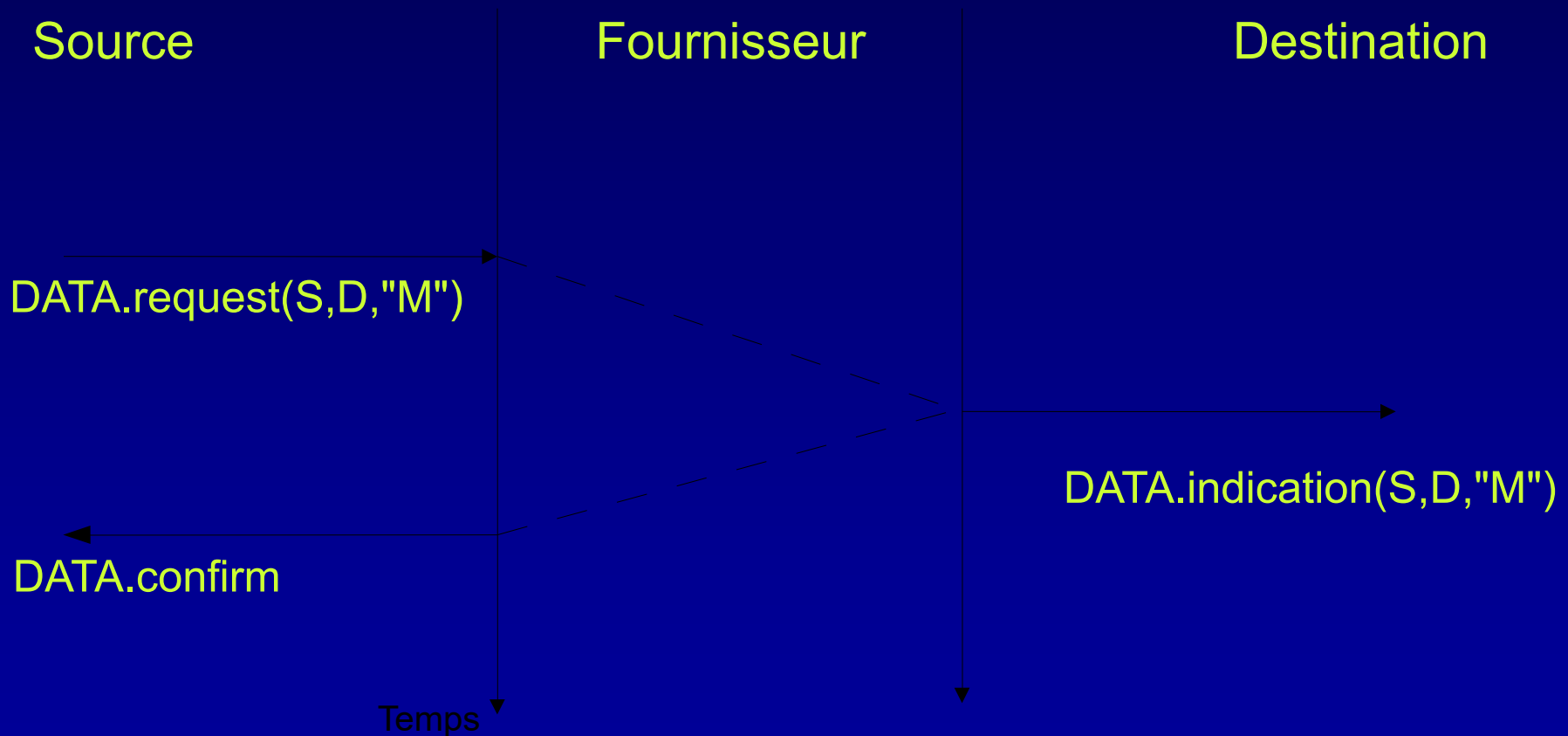
- Caractéristiques du service?

# Service sans Connexion (Cont.)

- Caractéristiques principales du service
  - respect de la séquence
    - aucune garantie
  - Pas d'établissement /termination de connexion
- Service confirmé
  - Envoyer data.confirm

# Service sans Connexion (Cont.)

- Exemple de service confirmé

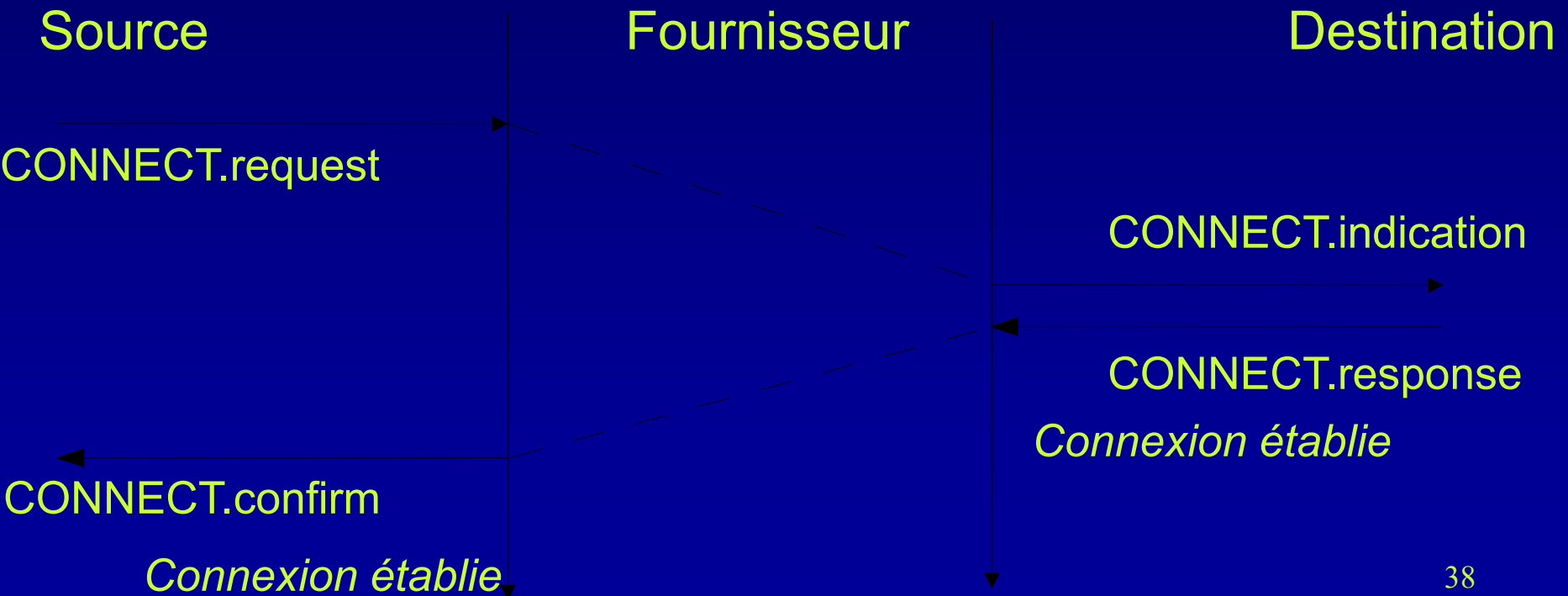


# Service Orienté Connexion

- Objectif
  - établir une association logique (connexion) entre deux utilisateurs afin de permettre l'échange bidirectionnel de messages entre les deux utilisateurs
- Principe
  - Ouverture de la connexion
  - Les deux utilisateurs peuvent envoyer des messages en profitant de cette connexion
  - Fermeture de la connexion
- Utilisation
  - utile pour envoyer de nombreux messages ou lorsqu'un dialogue complexe est nécessaire
  - exemple : service téléphonique

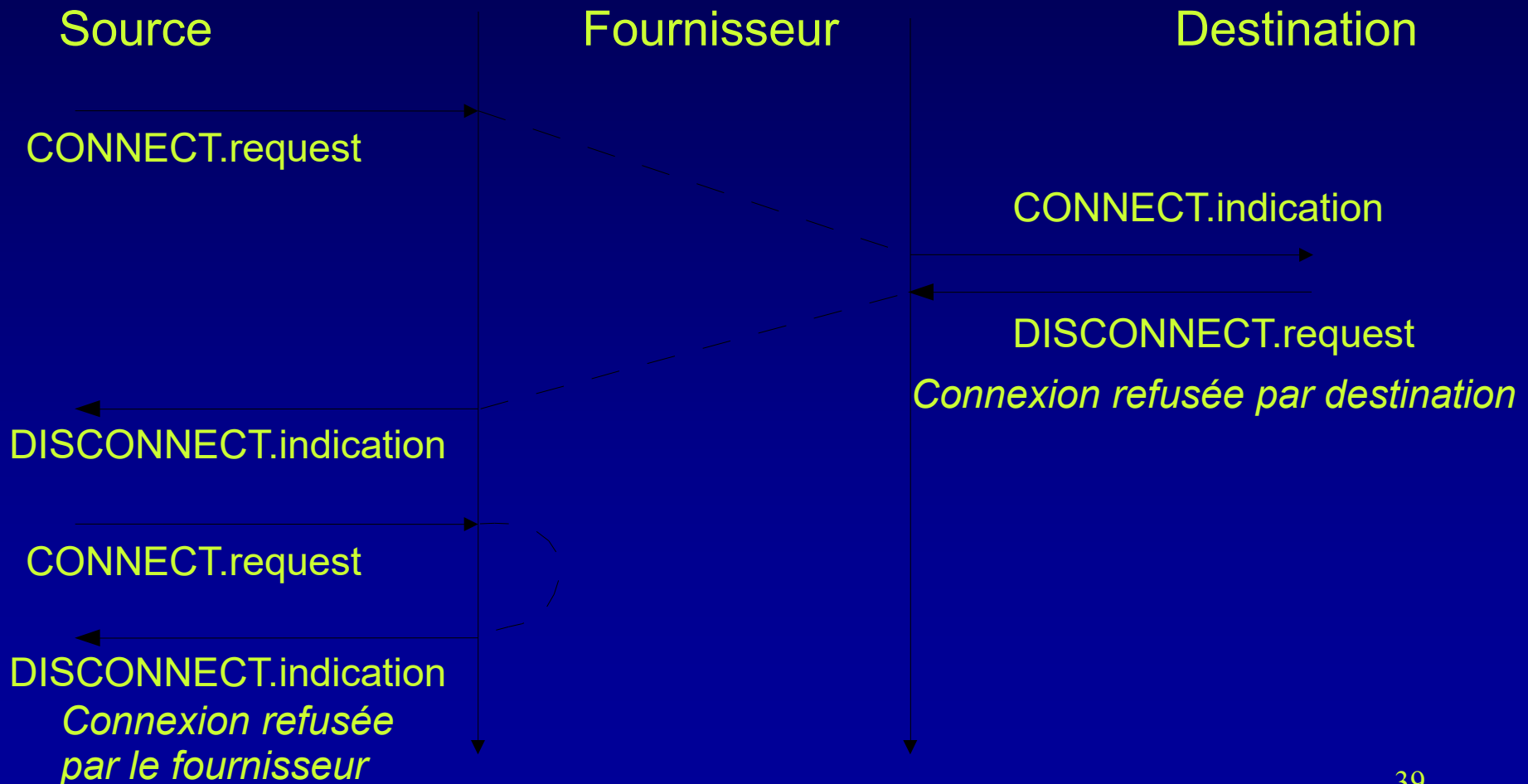
# Service Orienté Connexion (Cont.)

- Ouverture d'une connexion
  - Primitives
    - CONNECT.request
    - CONNECT.indication
    - CONNECT.response
    - CONNECT.confirm



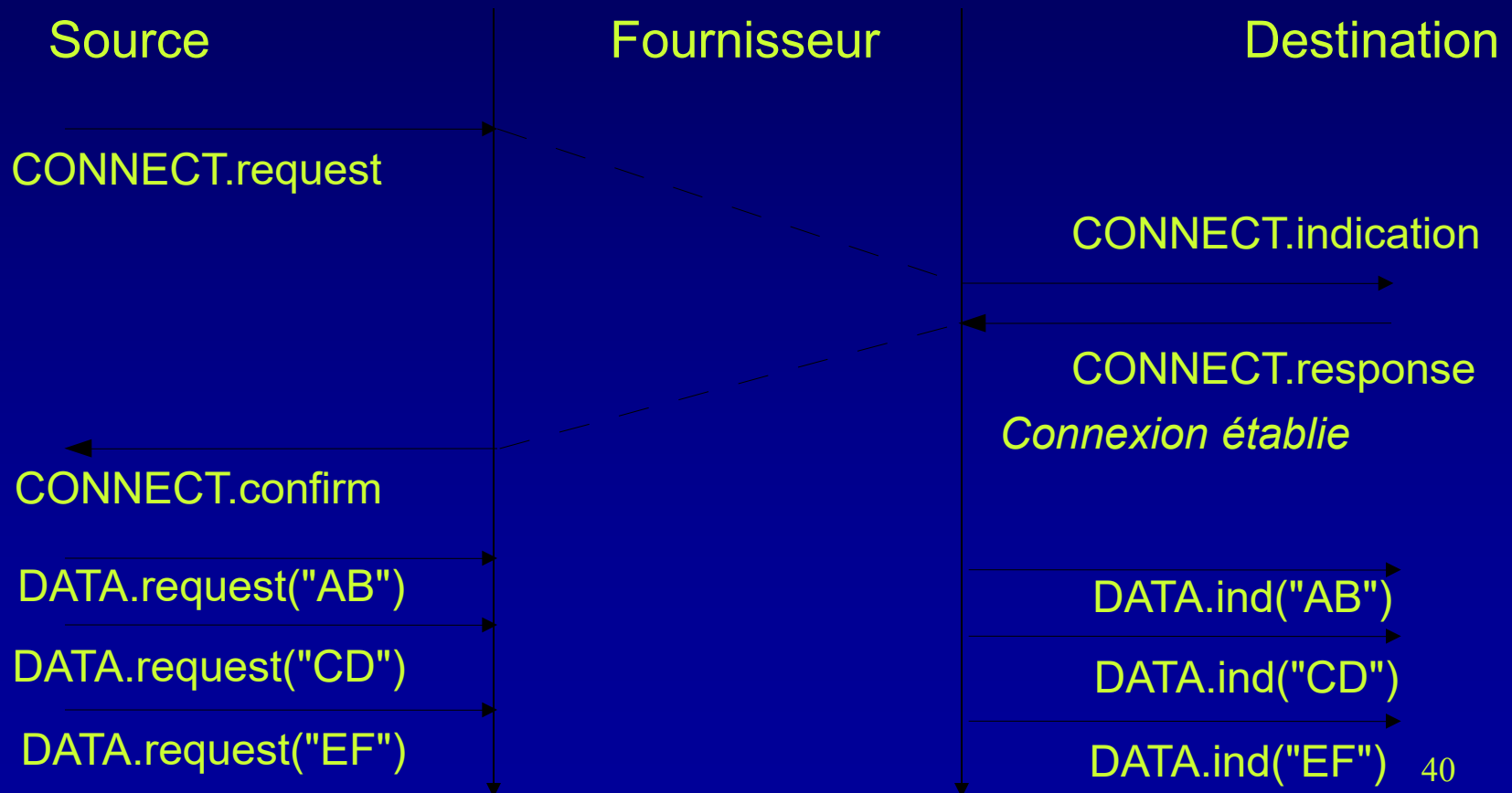
# Service Orienté Connexion (Cont.)

- Refus d'ouverture d'une connexion



# Transfert des Données Mode Message

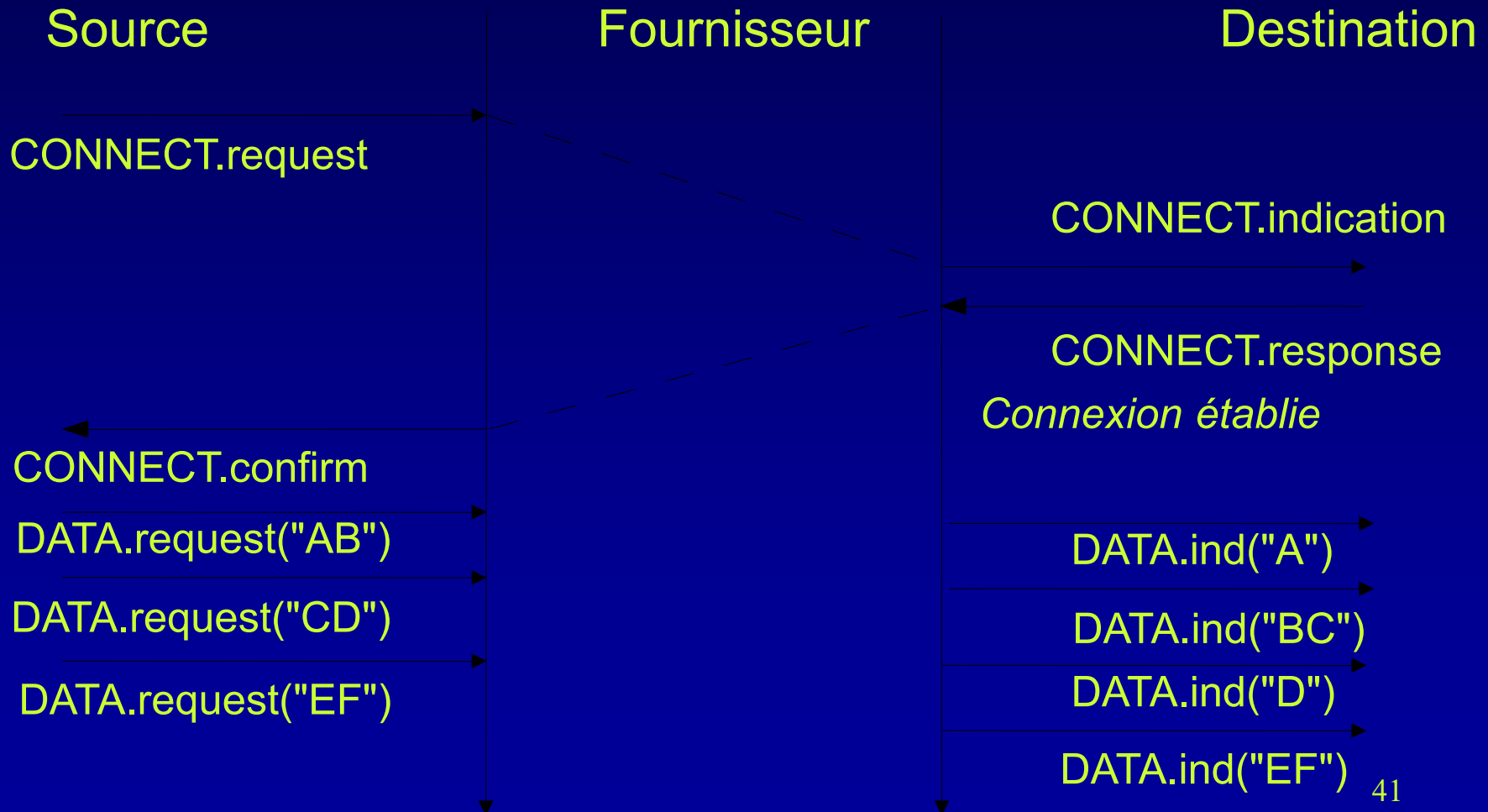
- Le fournisseur de service délivre un DATA.ind pour chaque DATA.req





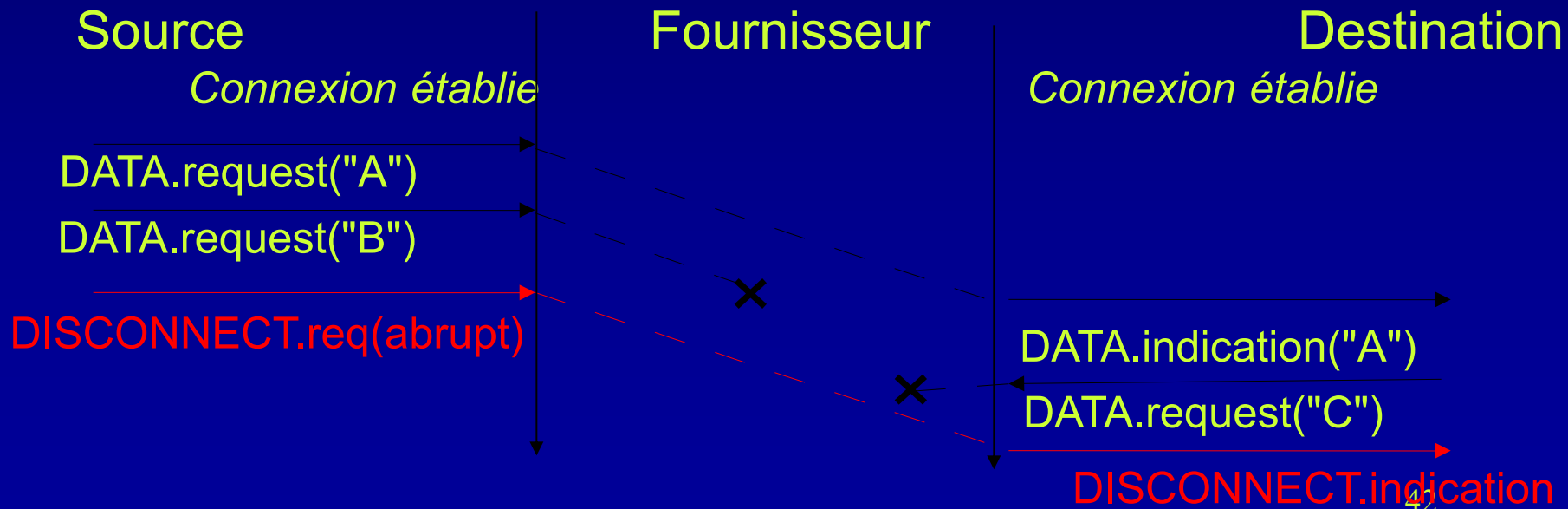
# Transfert des Données Mode Stream

- Pas de contraintes sur la façon dont les DATA.ind seront délivrées par le fournisseur du service



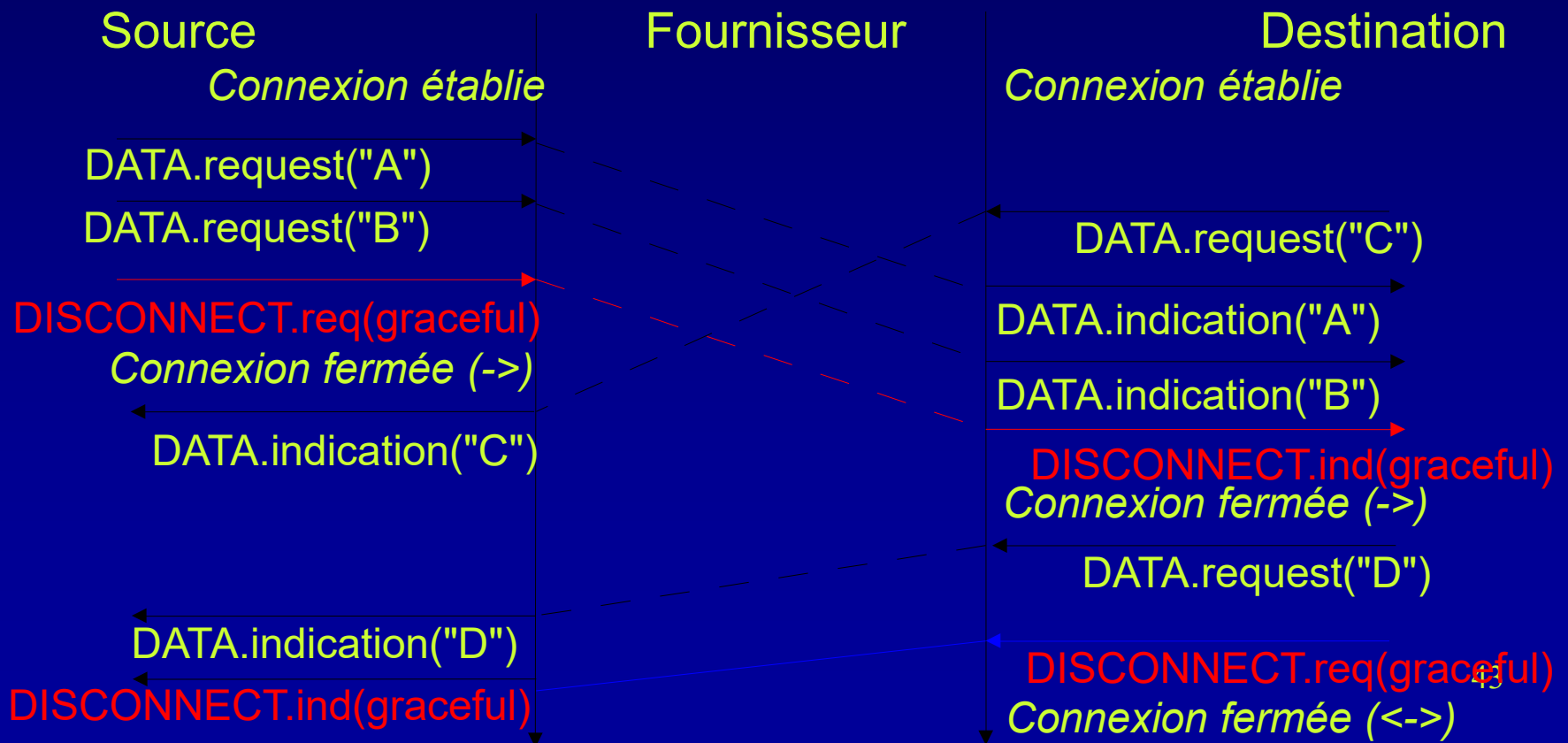
# Fermeture de Connexion

- Déconnexion abrupte
  - perte de données possible
- Une telle déconnexion peut être provoquée par un utilisateur ou (plus souvent) par le fournisseur



# Fermeture de Connexion (Cont.)

- Fermeture ordonnée
  - fermeture d'une direction à la fois
  - pas de perte de données



# Caractéristiques du Service Orienté Connexion

- Caractéristiques générales?

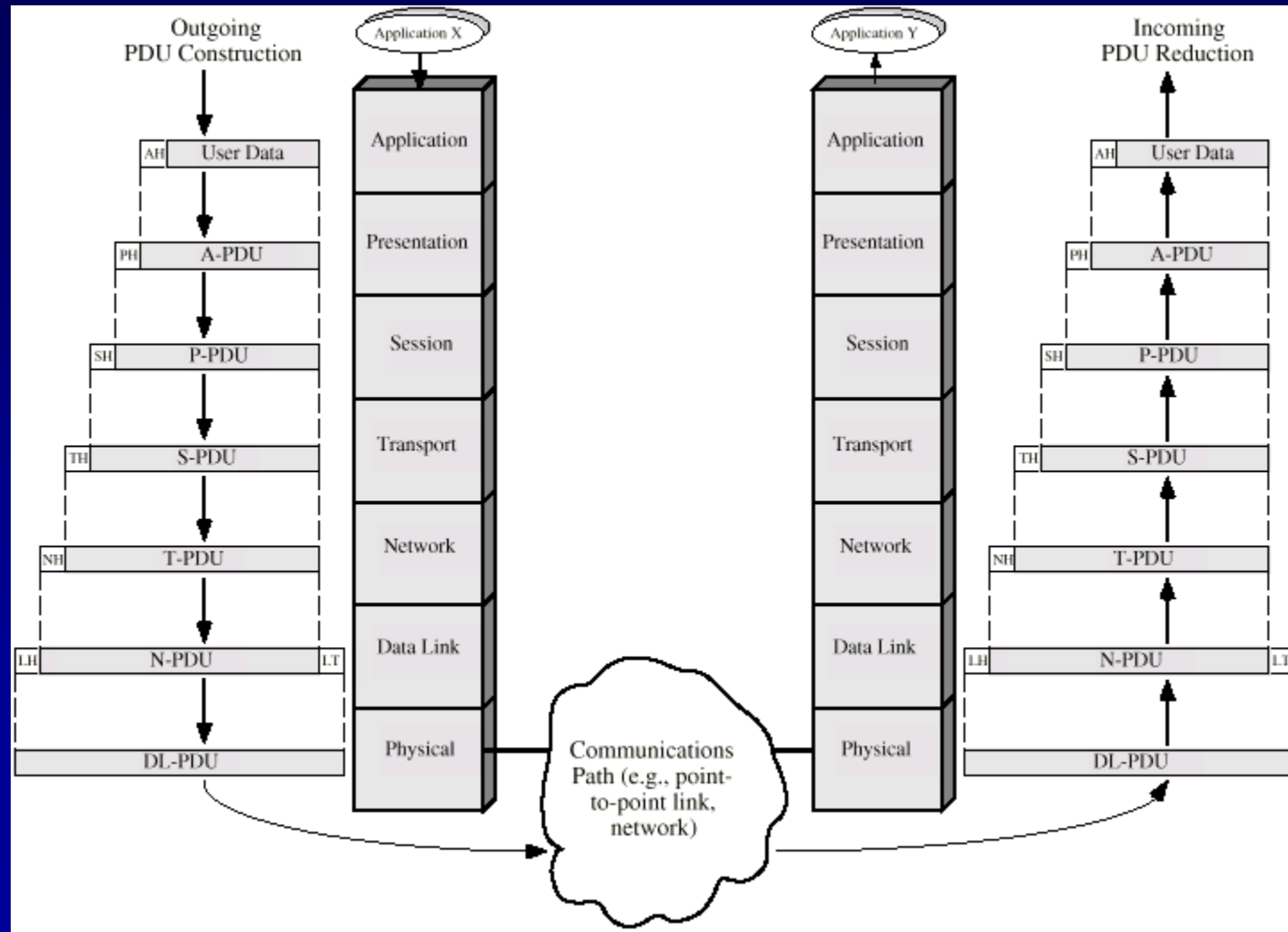
# Caractéristiques du Service Orienté Connexion

- **Caractéristiques générales**
  - transmission bidirectionnelle
    - une fois la connexion ouverte, les deux utilisateurs peuvent l'exploiter
  - respect de la séquence
    - garantie
  - fermeture de la connexion
    - abrupte lorsqu'elle est provoquée par le fournisseur
    - ordonnée (souvent) ou abrupte (parfois) lorsqu'elle est initiée par un utilisateur

# Service Fiable Vs. Service non Fiable

- Service fiable
  - Il n'y a jamais de données perdues
  - Le récepteur acquitte la réception de chaque message
- Service non fiable
  - Il n'y a aucune garantie de réception de données transmises

# Le modèle de référence OSI



# Couches OSI (1)

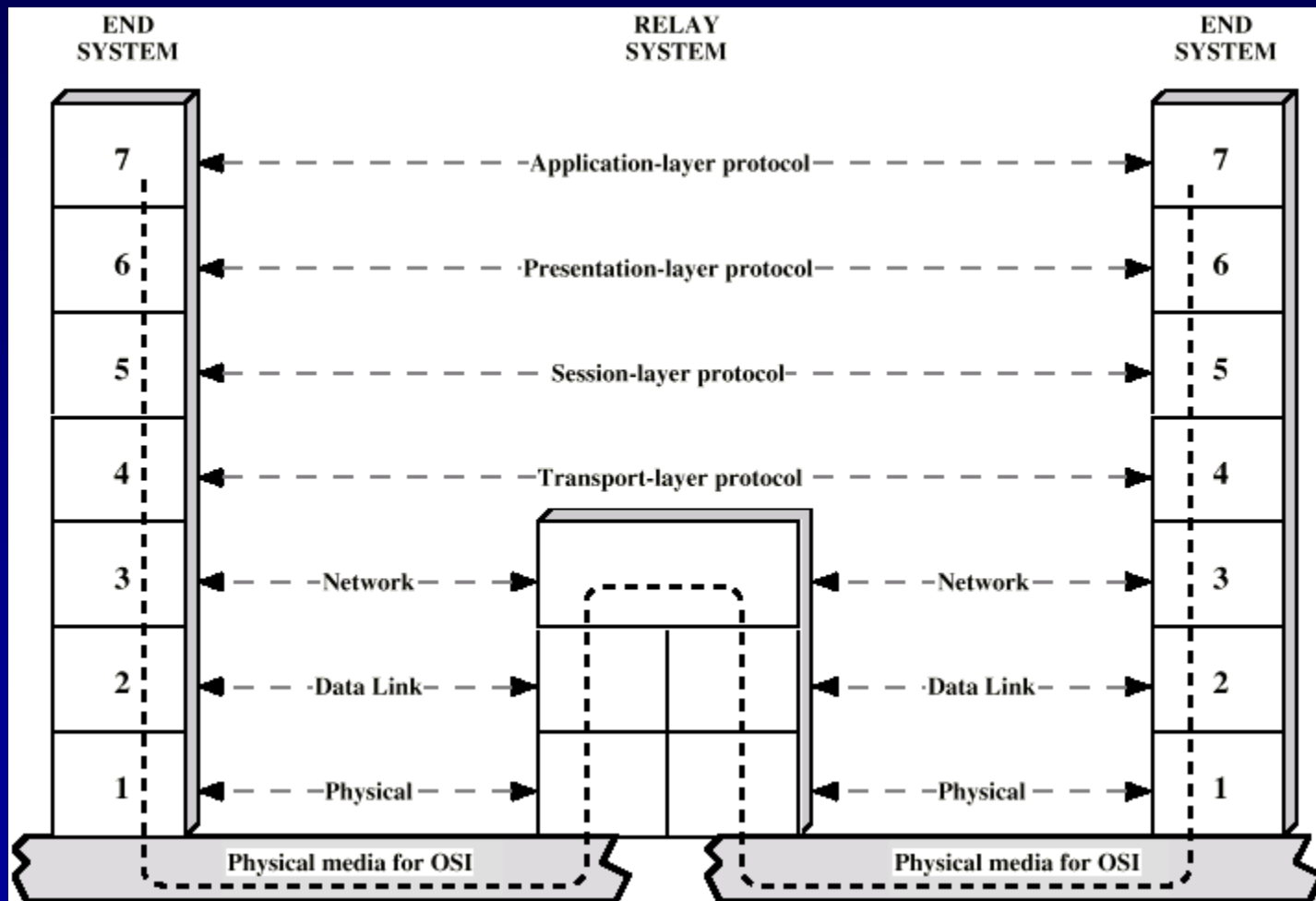
- Physique
  - Signaux
- Liaison
  - Détection et correction des erreurs
  - Contrôle de flux
- Réseau
  - Routage
- Transport
  - Communication de bout en bout
  - Fiabilité et ordonnancement des paquets



# Couches OSI (2)

- Session
  - Contrôle de session
- Présentation
  - Codage et Compression
  - Encryptage
- Application
  - Applications qui ont besoin de communiquer

# Utilisation de relais



# Contrôle d'erreur

- Couche liaison et couche transport?
  - Redondance?

# Le Modèle de Référence TCP/IP

<b>Application</b>
<b>Présentation</b>
<b>Session</b>
<b>Transport</b>
<b>Réseau</b>
<b>Data link</b>
<b>Physique</b>

<b>Application</b>
--------------------

<b>Transport</b>
<b>Internet</b>
<b>Host-to-network</b>

**Telnet - FTP - SMTP - DNS -  
RTP - ...**

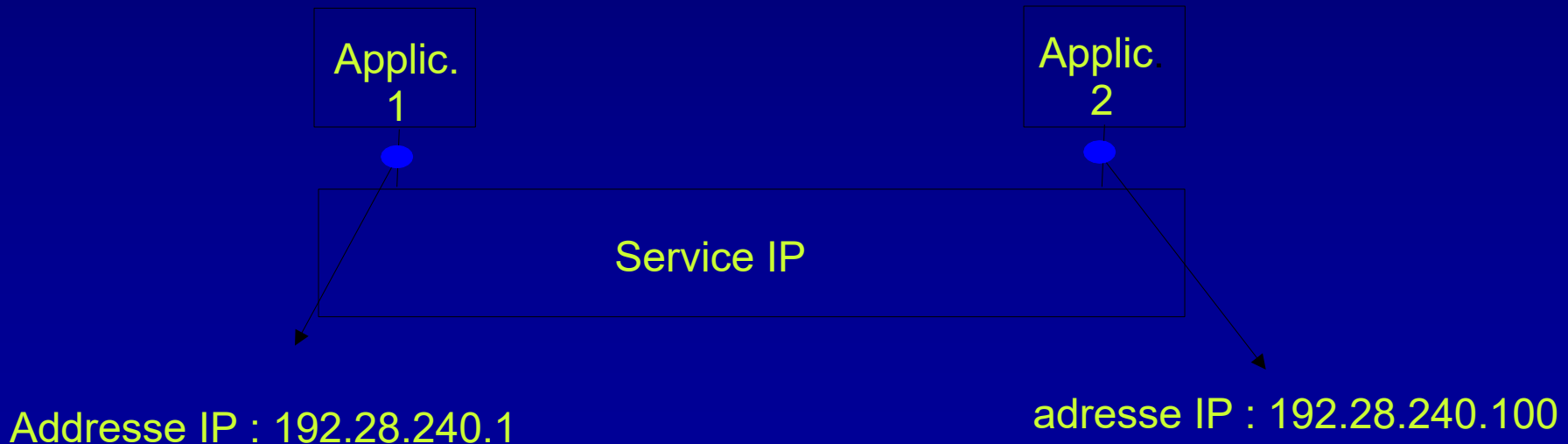
<b>TCP - UDP</b>
<b>IP</b>
<b>Ethernet, WLAN, Token Ring, etc.</b>

# Service Fourni par IP

- Caractéristiques

- service sans connexion non fiable

- taille des messages : maximum 64 Kbytes
    - perte possible de messages
    - non respect de la séquence
    - erreurs de transmission possibles et non détectées



# TCP - UDP

- **Transmission Control Protocol :**
  - Se base sur IP, est un protocole fiable, orienté connexion, permet la transmission sans erreur d'un flux de bytes d'une machine à une autre machine, découpe les messages et réassemble les paquets, contrôle de flux.
- **User Datagram Protocol :**
  - Se base sur IP, est un protocole non fiable, sans établissement d'une connexion.

# TCP-UDP

- Zoom
- Whatsapp
- Netflix
- Jeux videos

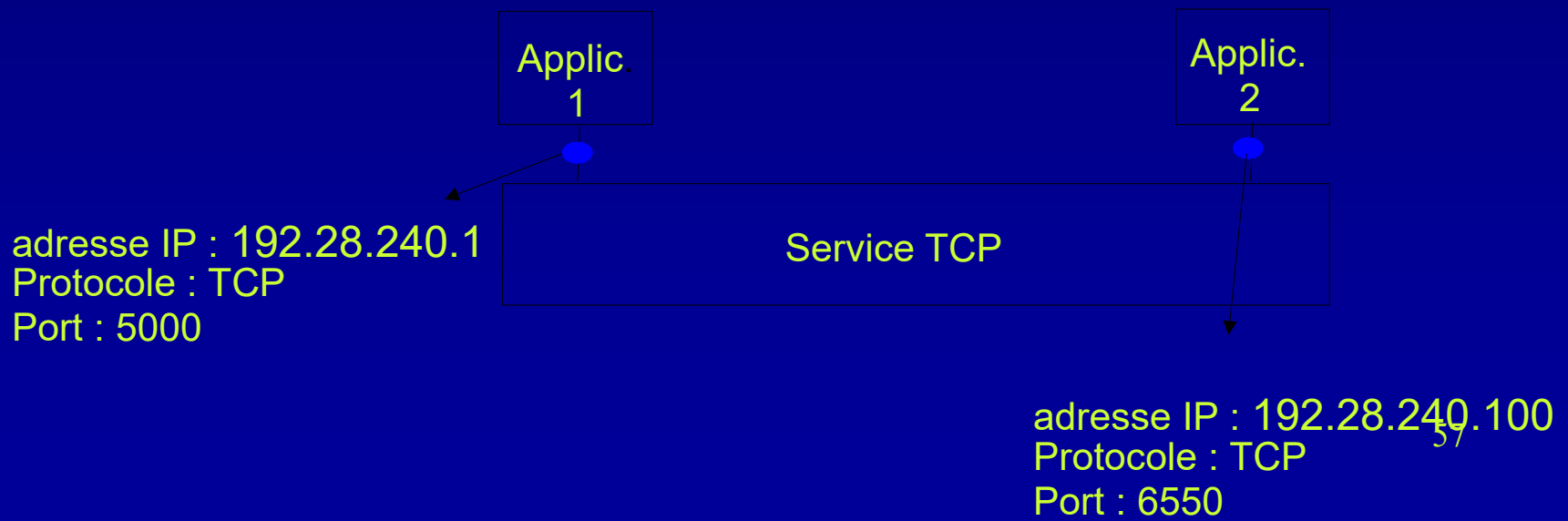
# TCP-UDP

- TCP convient aux applications où la fiabilité est cruciale
  - Possibles retards dus aux retransmissions.
- UDP est mieux adapté aux applications en temps réel
  - la rapidité et la faible latence sont plus importantes que la fiabilité des données.
  - Une légère perte de paquets n'a pas d'impact significatif sur l'expérience utilisateur dans ces scénarios.



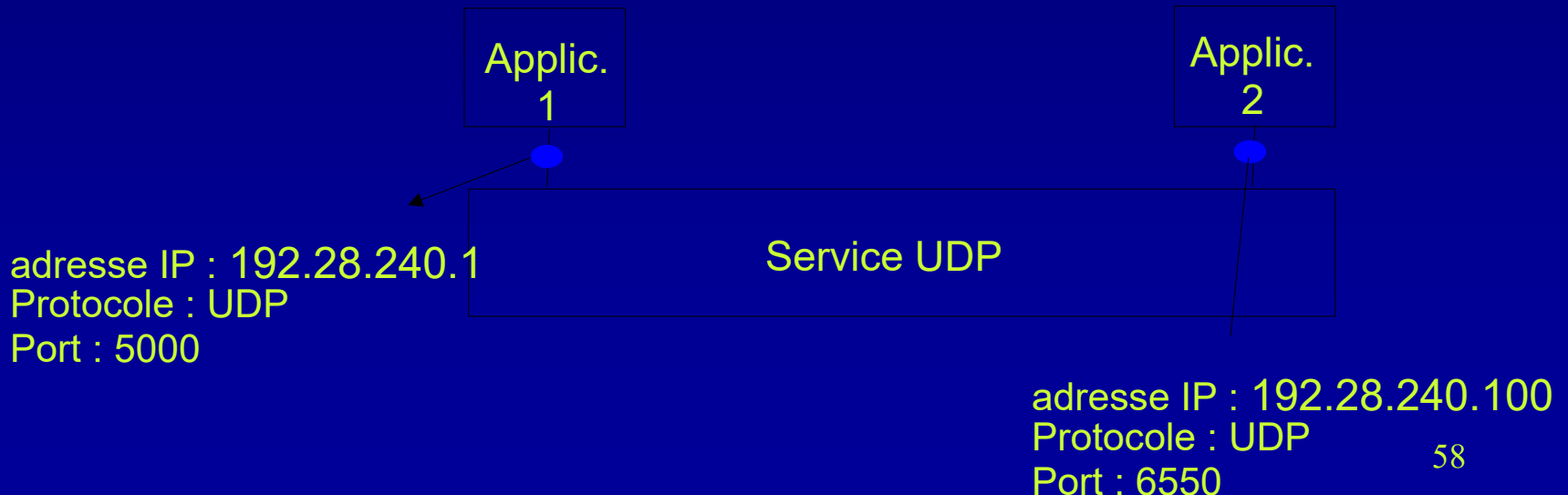
# Service TCP

- Identification d'une application
  - adresse IP + TCP + numéro de port
- Caractéristiques du service TCP
  - service orienté connexion
  - bidirectionnel
  - fiable
  - mode stream
  - fermeture de la connexion: abrupte par le fournisseur, ordonnée ou abrupte par les utilisateurs

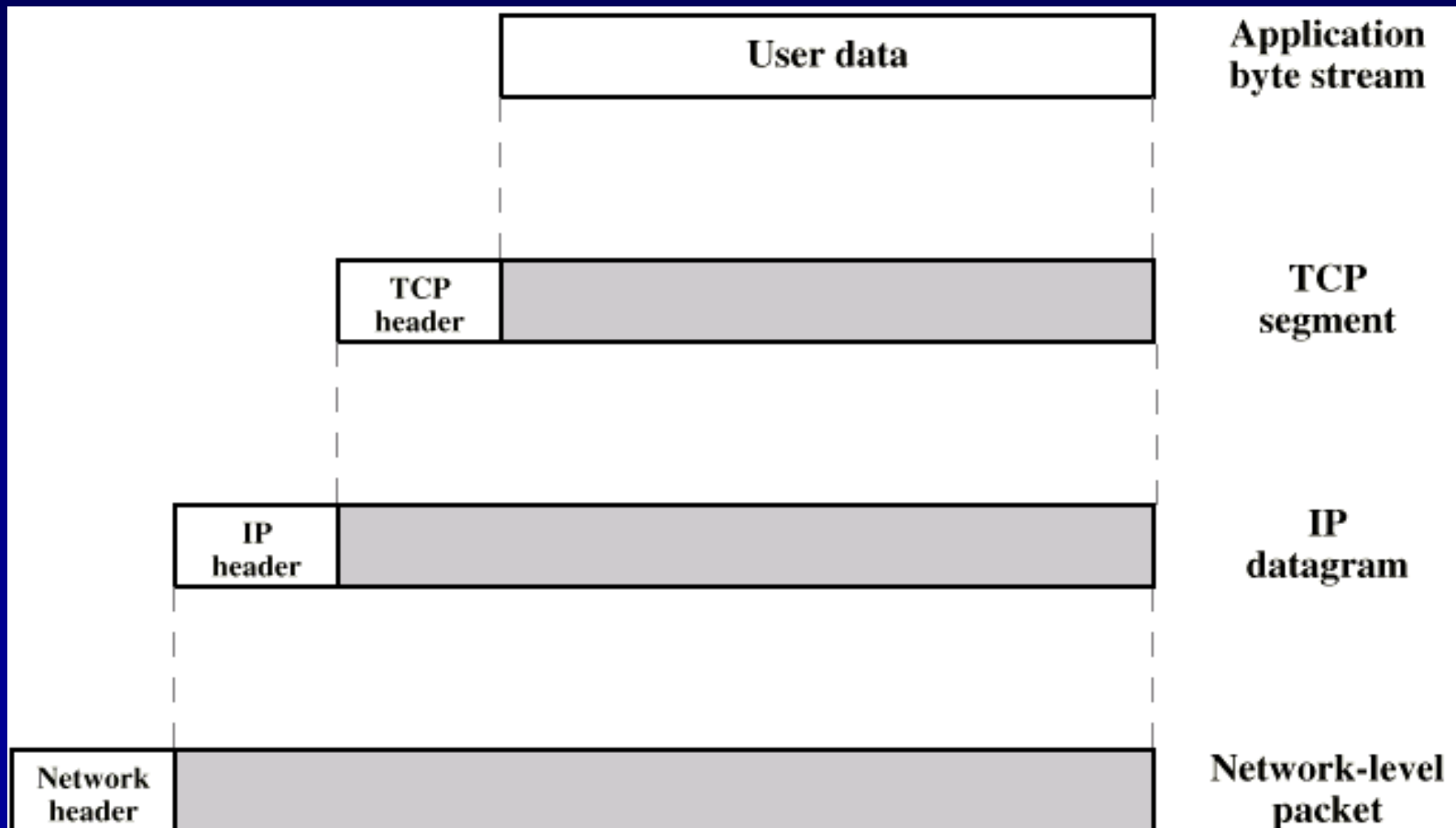


# Service UDP

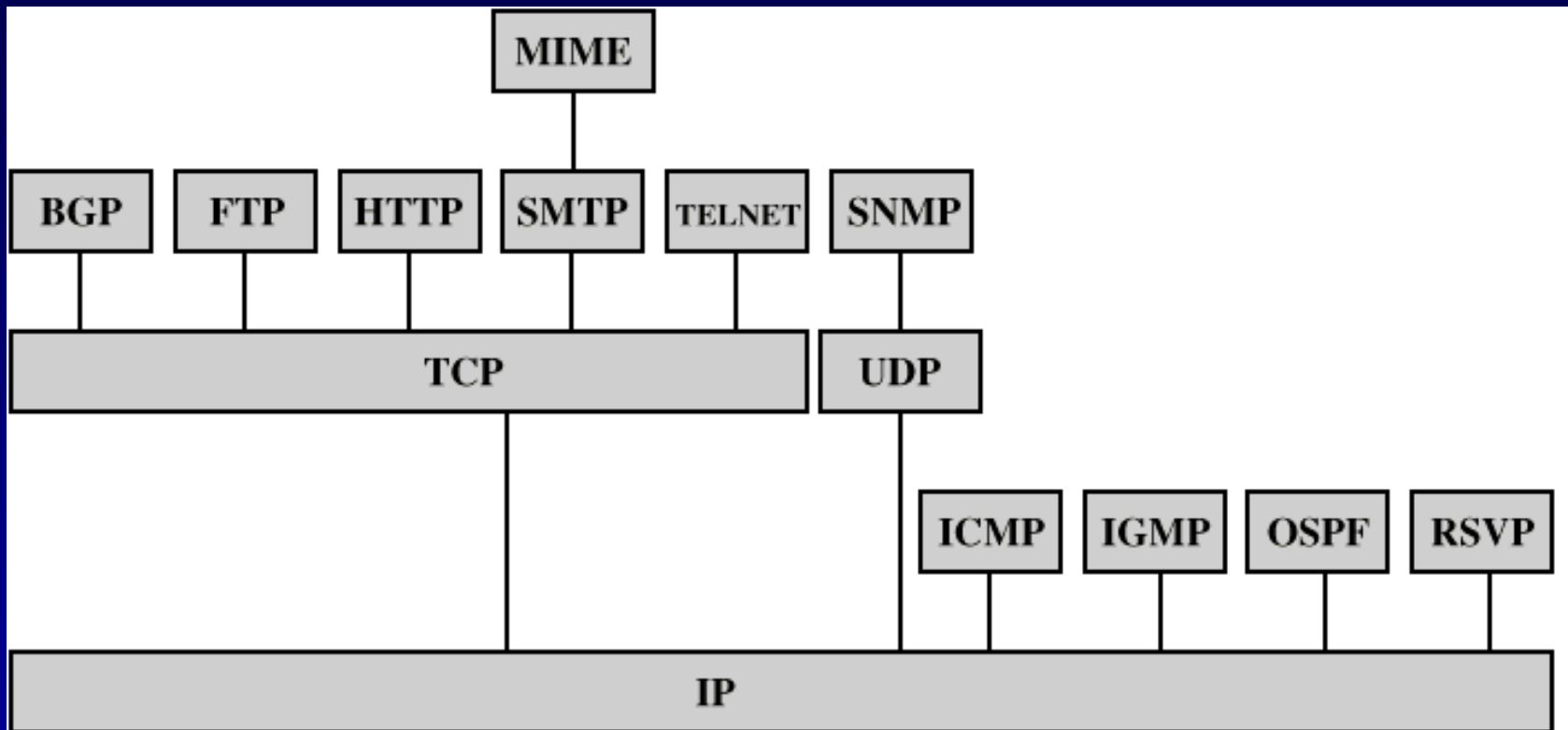
- Identification d'un utilisation
  - adresse IP + UDP + numéro de port
- Caractéristiques du service
  - service sans connexion
  - non fiable
    - pertes de messages possibles
    - détection mais pas correction des erreurs de transmission
    - non respect de la séquence



# TCP/IP: PDU



# TCP/IP: Quelques Protocoles



**BGP** = Border Gateway Protocol

**FTP** = File Transfer Protocol

**HTTP** = Hypertext Transfer Protocol

**ICMP** = Internet Control Message Protocol

**IGMP** = Internet Group Management Protocol

**IP** = Internet Protocol

**MIME** = Multi-Purpose Internet Mail Extension

**OSPF** = Open Shortest Path First

**RSVP** = Resource ReSerVation Protocol

**SMTP** = Simple Mail Transfer Protocol

**SNMP** = Simple Network Management Protocol

**TCP** = Transmission Control Protocol

**UDP** = User Datagram Protocol

# Conclusion

- Principes suivis pour définir un modèle en couches
- Protocoles Vs. Services
  - Protocole est exécuté entre entités paires
    - Ensemble de règles de communication
  - Couche N fournit un service à couche N+1
    - Ensemble de primitives
- Fonctionnalités du modèle en couches
  - Encapsulation
  - Fragmentation et assemblage
  - Contrôle de connexion
  - Livraison en ordre
  - Contrôle de flux/flot
  - Contrôle d'erreurs

# Conclusion (Cont.)

- Adressage
- Multiplexage
- Service de transmission
- Types de primitives utilisées
  - X.request, X.indication, X.response, X.confirm
- Service sans connexion Vs. Service orienté connexion
- Service fiable Vs. Service non fiable
- Modèle de référence OSI
  - 7 couches
- Modèle de référence TCP/IP
  - 5 couches