

1. Introduction

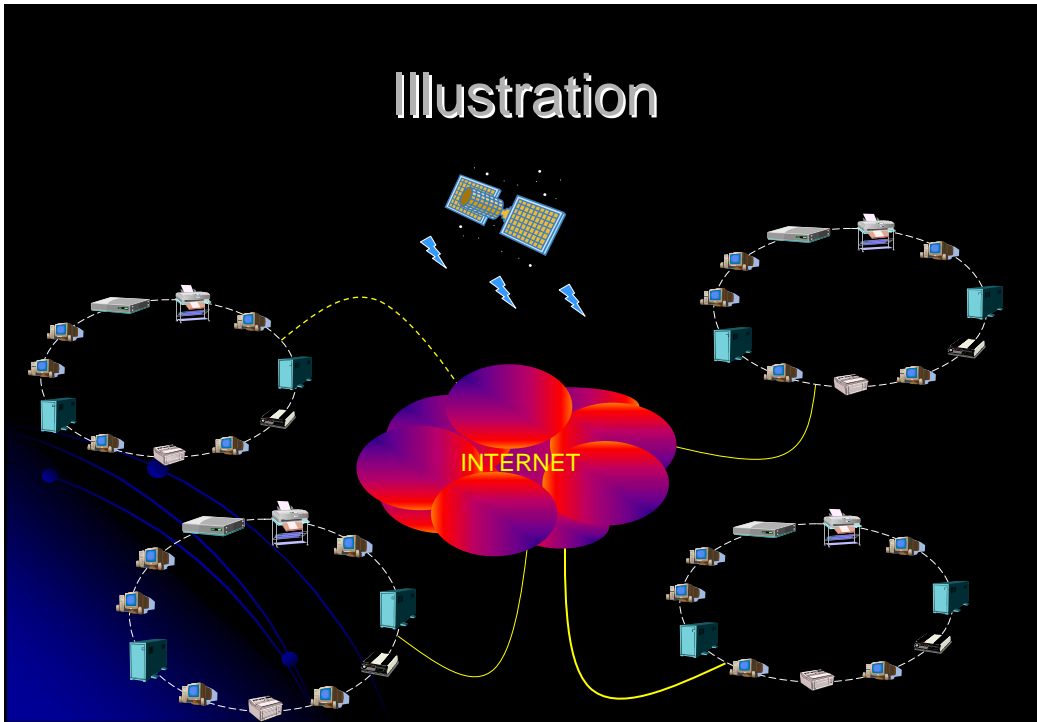
- Qu'est-ce qu'un réseau ?
 - *Fonction*
 - transporter des données d'une machine terminale vers une autre machine terminale
 - machine terminale = ?
 - *Support*
 - utilisation de câbles, ondes, lumière
 - *Recours à du logiciel*
 - règles, protocoles, ...
 - *Matériel d'acheminement*
 - commutateurs, routeurs, ...



Objectifs

- Les réseaux poursuivent un certain nombre d'objectifs :
 - Le *partage* des ressources
 - L'augmentation de la *disponibilité*
 - La réduction des *coûts*
 - La *centralisation* des données
 - La *sécurité* des données
 - La *communication*

Illustration



Terminologie

- Un certain nombre de 'types' de machines sont interconnectées via un réseau :
 - *serveurs de fichiers*
 - suite office, documents communs, ...
 - *serveurs d'applications*
 - base de données, gestion de planning
 - *serveurs de services*
 - serveur d'impression, de sauvegarde
 - *serveurs d'informations*
 - web, intranet, ...

Authentification

- La connexion à un réseau ou à un serveur distant nécessite de la sécurité :
 - réalisée en général grâce à un système d'authentification
 - système le plus classique :
 - nom de login et mot de passe
 - ou :
 - *passport, certificat, biométrie, ...*
 - gérée par l'ingénieur système :
 - c'est via l'OS que s'opère l'accès au réseau. La vérification lui en incombe donc. OS plus ou moins sûr.

Privilèges

- Liés à la notion de compte utilisateur :
 - possibilité de faire des regroupements logiques
 - autorisent ou empêchent l'utilisation de ressources réseaux :
 - serveurs
 - imprimantes
 - ...
 - définis également par l'ingénieur système

Les différents types de réseaux

- Le réseau est le support de communication entre les clients et les serveurs :
 - *Physiquement*, il est constitué de câblages, de faisceaux hertziens, ...
 - Les différents acteurs du réseau utilisent des *protocoles réseaux* pour communiquer
 - Sur le même support physique, peuvent *cohabiter* plusieurs protocoles différents :
 - *condition* : une couche inférieure commune
 - Nous verrons plus tard une autre différence liée à la topologie

Protocoles réseaux

- Protocole :
 - ensemble de règles qui régissent une communication
 - il existe un grand nombre de protocoles différents
- Pour pouvoir communiquer :
 - le client et le serveur doivent utiliser les mêmes protocoles réseaux
 - les protocoles sont partiellement fonction du support utilisé

Protocoles réseaux

- Parmi les différents protocoles réseaux existants, citons :
 - réseaux Windows (NT, 2000, XP) :
 - initialement NetBeui, maintenant IP
 - Novell :
 - initialement IPX et SPX, maintenant IP
 - TCP/IP :
 - adopté quasi unanimement
 - à la base d'Internet
 - utilisation aussi bien dans les petites que dans les grandes structures

Abus de langage

- **Attention, il ne faut pas confondre le système d'exploitation des serveurs utilisés dans le réseau et le type de réseau**
- Un réseau composé de serveurs NT peut utiliser comme protocole réseau TCP/IP et non pas les protocoles réseaux NT

Caractéristiques physiques

- Pas de classification mais deux critères importants :
 - la **technologie de transmission**
 - la diffusion
 - le point à point
 - la **taille**
 - de 10 centimètres à ... la planète

La technologie de transmission

- La diffusion

chaque message envoyé est reçu par toutes les machines. Il faut donc une adresse identifiant le destinataire réel. La diffusion est *générale* (**broadcast**) ou *restreinte* (**multicast**) : la radio, Ethernet, ...
- Le point à point

chaque message passe par un ensemble de liaisons entre paires de machines (**unicast**). Il en découle potentiellement un grand nombre de 'routes' possibles. Le choix optimal devient crucial : le téléphone, les liens internodes, ...

La technologie de transmission

- De manière générale :
 - **petit** réseau
 - si le réseau est géographiquement peu étendu, c'est la diffusion qui est utilisée
 - **grand** réseau
 - si le critère géographique est plus important, on recourt aux réseaux point à point

La taille

< 20 cm	<i>bus d'ordinateur</i>
de 1 à 10 m	<i>réseau personnel</i>
de 10 à 1.000 m	<i>réseau local (bâtiment, site, campus) : LAN</i>
10 Kms	<i>réseau métropolitain : MAN</i>
100 à 1.000 Kms	<i>réseau longue distance (pays, continent) : WAN</i>
10.000 Kms	<i>la planète (Internet)</i>

Techniques de transfert

- L'acheminement de données informatiques se fait généralement en procédant à une **découpe** en 'blocs' de l'information initiale. Ces *blocs* sont ensuite acheminés selon différentes techniques. Nous exposerons les 3 premières :
 - *la commutation de circuits*
 - *le transfert de messages*
 - *le transfert de paquets*
 - mixte circuits-paquets
 - commutation multicircuits

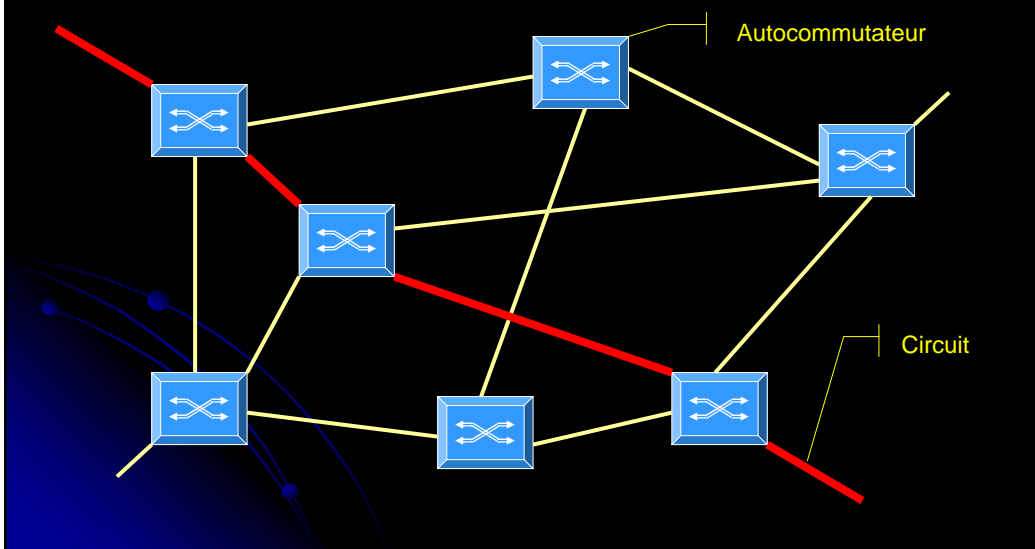
Commutation de circuits

- Première technique à apparaître
 - le réseau le plus connu :
 - réseau téléphonique
- Caractéristiques :
 - avec cette commutation, un circuit physique est **réellement construit** entre l'émetteur et le récepteur (mobilisation de ressources)
 - cette matérialisation est **exclusive** à ces deux parties

Commutation de circuits

- le circuit **doit** d'abord être établi **avant** tout transit. C'est donc un mode connecté (délai \leftrightarrow sécurité)
- si aucune donnée n'est échangée, la liaison reste active d'où un gaspillage de ressources de bande passante
- il est également possible de regrouper plusieurs communications sur une même liaison (multiplexage) afin d'utiliser au mieux la bande passante
- malheureusement cela alourdit la gestion

CC : illustration



Transfert de messages

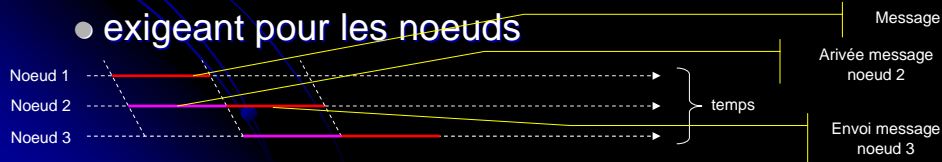
- Développé pour optimiser l'utilisation de la bande passante des supports :
 - un message est une suite d'informations formant, tant pour l'émetteur que pour le destinataire, **un tout logique** (fichier, ligne de commande, ...)
 - ce type de réseau se compose d'un ensemble de noeuds. Le message est véhiculé de noeud en noeud

Transfert de messages

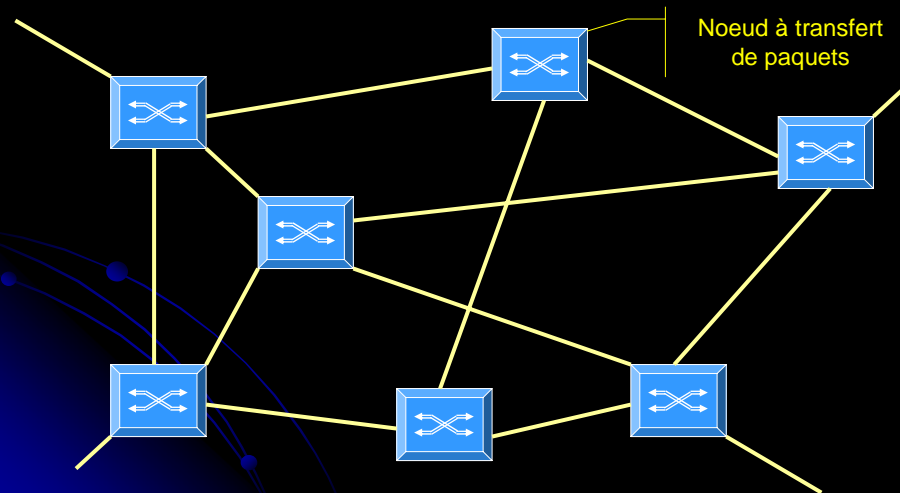
- le message est intégralement mémorisé dans chaque noeud avant d'être passé au suivant (store and forward)
- le temps de réponse est donc assez élevé car c'est au moins la **Σ des temps** de réception transmission à chaque noeud
- ce mode de transfert nécessite également des acquittements/demandes de retransmission en cas de bonne réception ou non (conservation ou non du message dans le noeud)

Transfert de messages

- Avantages sur la commutation :
 - en cas de panne ou de rupture d'un lien, un chemin alternatif peut-être (automatiquement ou non) trouvé
 - pas de ressource exclusivement mobilisée
- Inconvénients :
 - temps de réponse
 - exigeant pour les nœuds

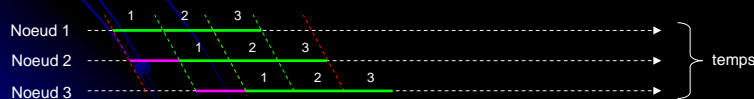


Transfert de messages



Transfert de paquets

- Amélioration du transfert de messages :
 - les messages sont **découpés** en paquets (< 2.000 bits)
 - cela allège et facilite la tâche des noeuds car les paquets (taille réduite) arrivent plus vite et sont donc retransmis plus rapidement
 - ceci est particulièrement vrai en cas de reprises sur erreurs



Transfert de paquets

- Inconvénient :
 - la découpe en paquets induit la nécessité de :
 - ré assembler (<> *découpe*)
 - réordonner (*ordre d'arrivée* <> *ordre arrivée*) les paquets
- Internet est un exemple de type de réseau à transfert (ou plus exactement) routage de paquets

Transfert de paquets

- D'autres protocoles se fondent sur des paquets de taille encore plus réduite (des **cellules**)
- En outre, ils imposent aux cellules de toujours suivre la même route ... virtuelle
- Les performances augmentent, la perte de séquence est évitée, mais la gestion de ces protocoles se complexifie. **ATM** en est un bon exemple

Le routage

- Corollaire :
 - le routage peut être **statique** :
la route est toujours la même car prédéfinie
 - plus simple, moins de complexité, moins de travail pour les noeuds
 - MAIS, si rupture d'une ligne, le transfert échoue
 - le routage peut être **dynamique** :
la route est calculée, pour chaque paquet, par les noeuds.
Cela accroît leur travail mais garantit mieux des pannes

Logiciels de réseaux

- Afin de réduire la complexité des logiciels de réseaux, une organisation en **strates** a été adoptée :
 - on appelle **couche** ou *niveau* une partie spécialisée de ces logiciels.
 - toute partie est chargée de **rendre des services** à la couche immédiatement supérieure
 - toute partie peut **utiliser les services** de la couche immédiatement inférieure
- OSI : Modèle de référence

Couches, services et protocoles

- Ce modèle comprend 7 couches
- Concept décrit dans ISO 7498-1
- Cette architecture en couche requiert, pour un niveau **N**, la définition de 3 objets :
 - le **service** N
 - le **protocole** N
 - les points d'accès au niveau N, c'est-à-dire les **N-SAP** (service Access Point)

Couches, services et protocoles

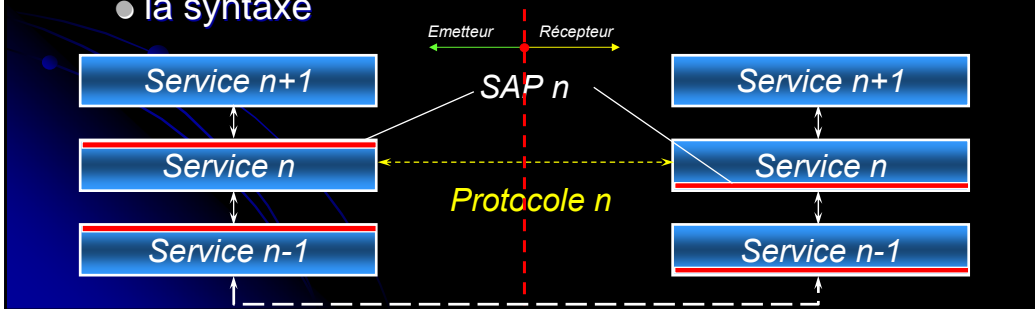
- Le service de la couche N :
consiste en une série d'*actions* (événements et primitives) *devant être accomplies par la couche N* en vue de réaliser le service fourni à la couche N+1 (même machine) : **notion verticale**
- Le protocole de la couche N :
définit un ensemble de *règles nécessaires à la réalisation du service de la couche N*. Ces règles définissent les mécanismes permettant le transport d'informations d'une couche N vers une autre couche N (machine distante) : **notion horizontale**

Couches, services et protocoles

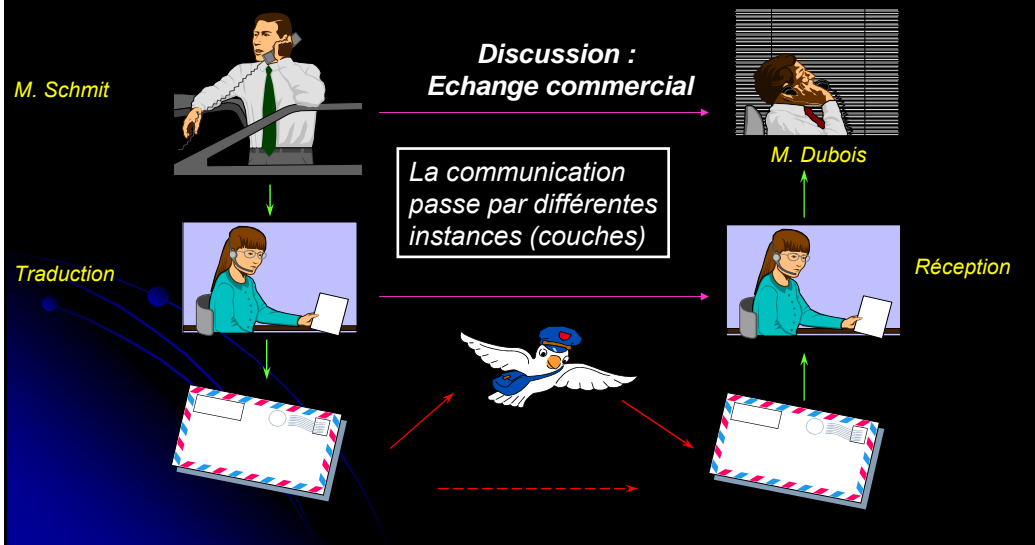
- Les points d'accès au service N :
 - situés à la frontière (N+1)-(N)
 - c'est là que s'échangent les paramètres
 - c'est là également qu'est associée une adresse permettant de définir le service N d'une machine donnée. En effet, une même machine peut offrir plusieurs services différents à un même niveau

Couches, services et protocoles

- Un service et un protocole sont déterminés par 3 attributs :
 - la sémantique d'association
 - la sémantique de fonctionnalité
 - la syntaxe



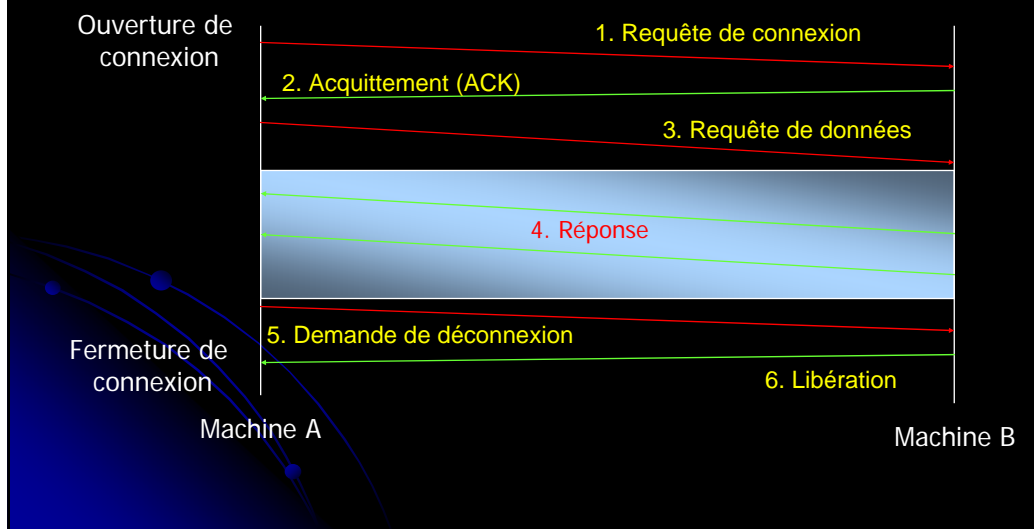
Couches, services et protocoles



Couches, services et protocoles

- La sémantique d'association :
 - Chaque couche N peut offrir 2 types de service à la couche N+1:
 - le mode **connecté** (// téléphone)
 - le mode **sans connexion** (// poste)
 - En mode connecté, 3 phases temporelles se succèdent :
 - établissement de la connexion
 - échange des données
 - fermeture de la connexion

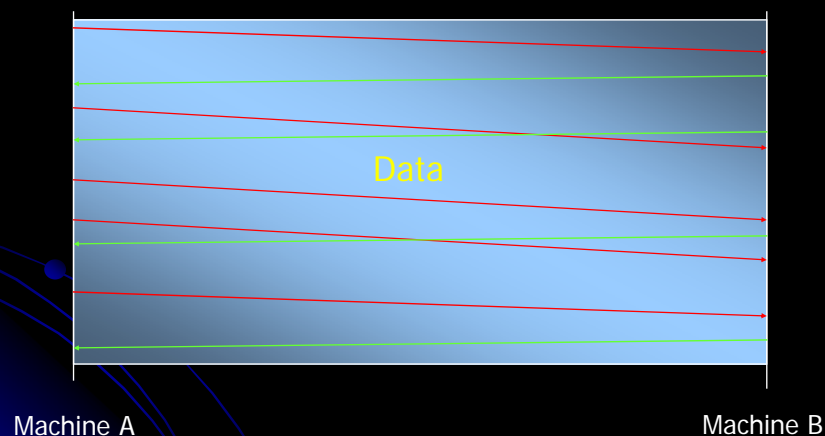
Couches, services et protocoles



Couches, services et protocoles

- Le mode sans connexion :
 - possède ses propres avantages :
 - rapidité à l'initialisation
 - il est particulièrement utilisé dans le cas de transferts courts (beaucoup de petits blocs)
 - certaines applications le réclament :
 - le mail. En effet, un utilisateur ne doit pas être connecté en permanence pour recevoir son courrier
- Le service non fiable et sans connexion est appelé **service datagramme**
 - non fiable = ?
 - l'e-mail en est un exemple

Couches, services et protocoles



Couches, services et protocoles

- Chacun de ces services peut être caractérisé par une QoS :
 - **certains sont fiables** : pas de perte de données (accusés de réception, ...). Si cela est fiable, c'est également très lourd.
 - FTP en est un exemple, ...
 - **d'autres ne le sont pas**. Dans d'autres cas (video, voix, ...), on peut tolérer quelques pertes en évitant les retards liés aux acquittements
 - mail, Video On Demand, ...

Primitives de services

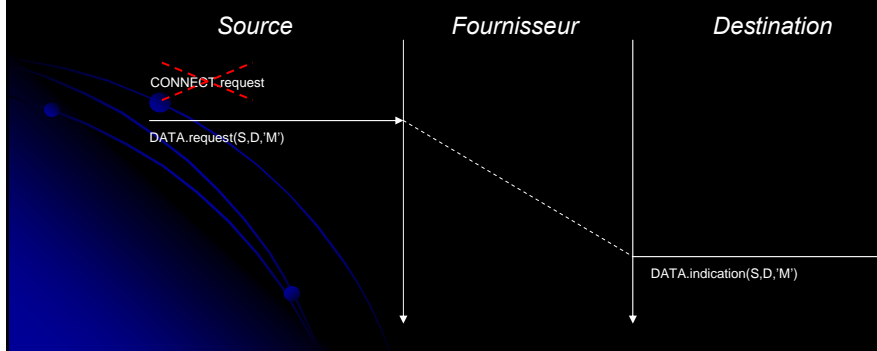
- Une primitive est une représentation abstraite d'une interaction entre un utilisateur et son fournisseur de service :
 - **X.request**
 - sens : utilisateur → fournisseur
 - **X.indication**
 - sens : fournisseur → utilisateur, liée à un X.request distant
 - **X.response**
 - sens : utilisateur → fournisseur, en réponse à un X.indication
 - **X.confirm**
 - sens : fournisseur → utilisateur, liée à un X.response distant

Primitives de services

- Le **X** :
 - les primitives de services s'appliquent aux :
 - CONNECT
 - DATA
 - DISCONNECT

'QoS' : service sans connexion (1)

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



'QoS' : service sans connexion (2)

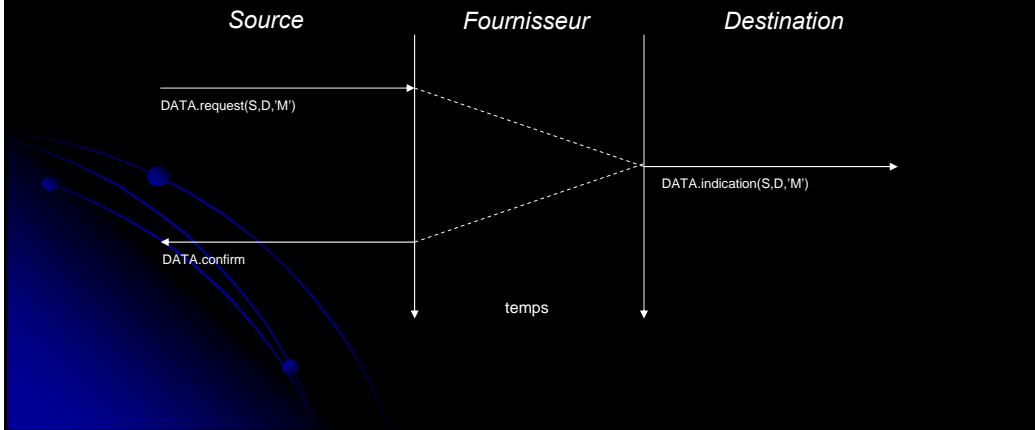
- Caractéristiques du service :
 - confirmation :
 - primitive DATA.confirm pour confirmer l'envoi d'un message
 - fiabilité :
 - le service garanti ou non l'absence d'erreurs et de pertes
 - protection contre les erreurs de transmissions :
 - le service évitera, détectera ou non les erreurs

'QoS' : service sans connexion (3)

- protection contre les erreurs de messages :
 - le service évitera ou non les pertes de messages
- respect de la séquence
 - l'absence de mode orienté connexion induit, à l'origine, une absence de garantie du respect de la séquence
 - MAIS, la QoS permet :
 - de demander le respect de la séquence des messages d'une même source
 - de demander le respect global pour toutes les sources

'QoS' : service sans connexion (4)

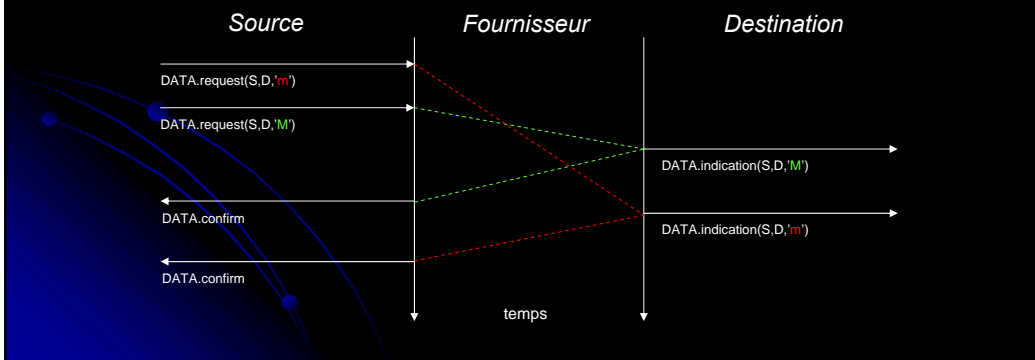
- Service **confirmé** :



'QoS' : service sans connexion (5)

- Problème :

- comment un utilisateur peut-il savoir à quel DATA.request un DATA.confirm correspond ?



'QoS' : service orienté connexion

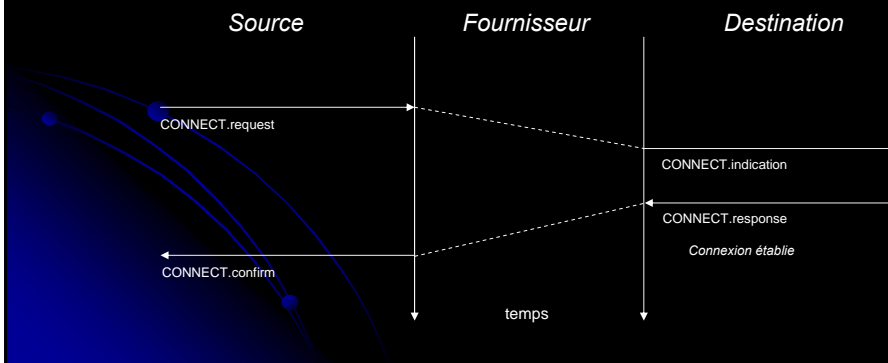
- Caractéristiques du service :
 - transmission bidirectionnelle :
 - une fois la connexion ouverte, les deux utilisateurs peuvent l'exploiter
 - service fiable :
 - respect de la séquence
 - pas de perte de donnée
 - pas d'erreur sur les données
 - mode message ou mode stream

'QoS' : service orienté connexion (2)

- fermeture de connexion :
 - abrupte lorsqu'elle est provoquée par le fournisseur
 - ordonnée ou abrupte lorsqu'elle est provoquée par un utilisateur

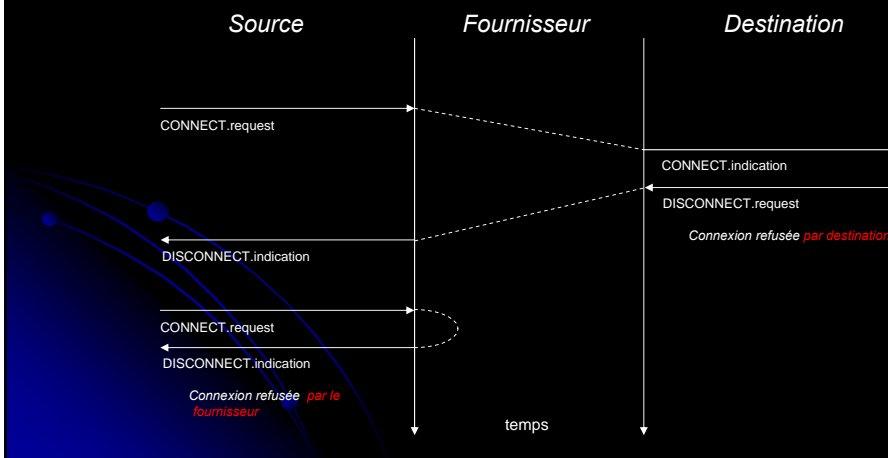
'QoS' : service orienté connexion (1)

- Ouverture d'une connexion :
 - *CONNECT.request*, *CONNECT.indication*, *CONNECT.response*, *CONNECT.confirm*



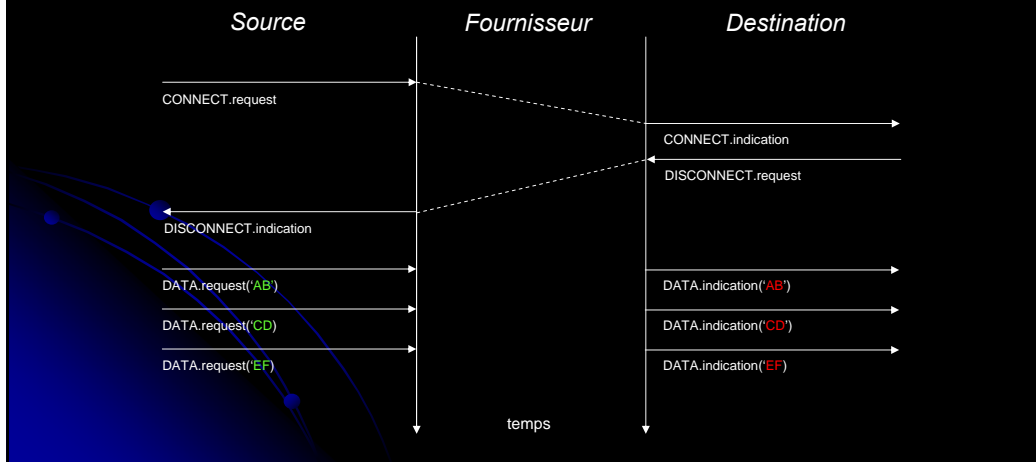
'QoS' : service orienté connexion (2)

- Refus d'ouverture d'une connexion



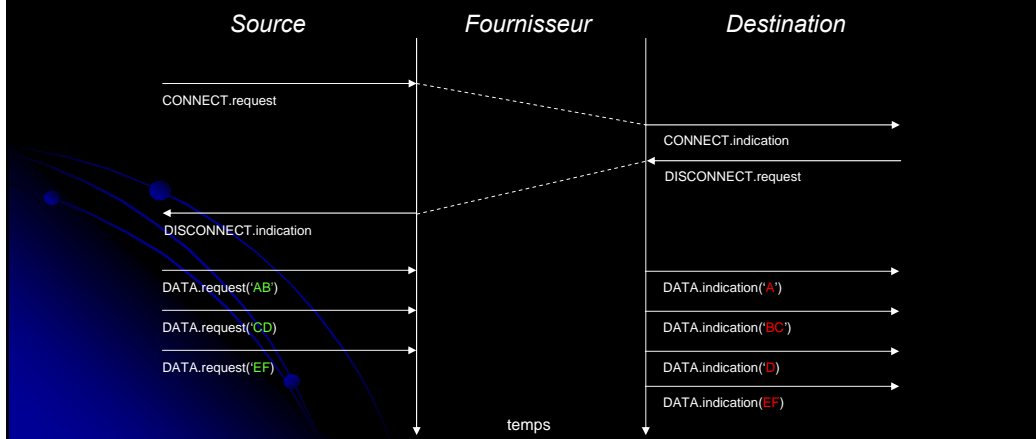
Transfert données, mode message

- un DATA.request → un DATA.indication



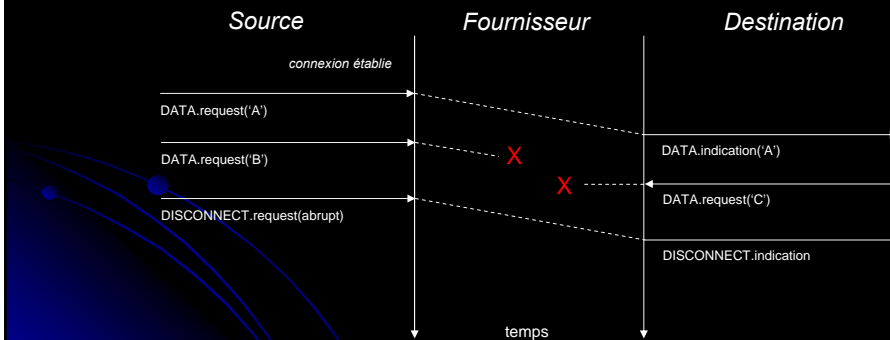
Transfert données, mode stream

- Aucune garantie sur le découpage des données en DATA.indication



Fermeture de connexion

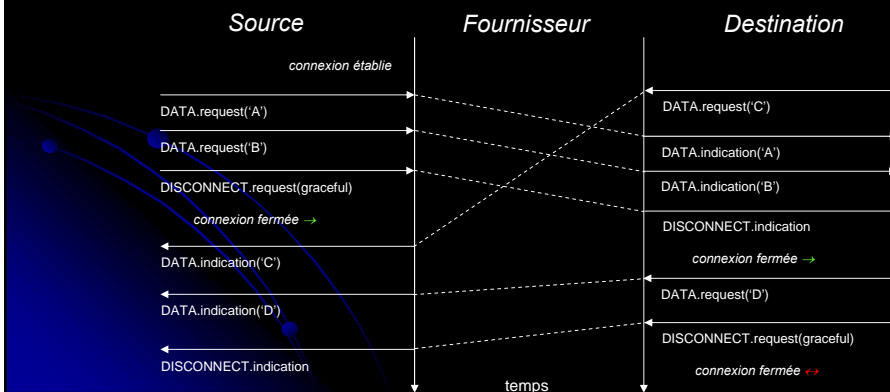
- Déconnexion abrupte :
 - perte de données possible



- déconnexion par un utilisateur ou par le fournisseur

Fermeture de connexion (2)

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



Couches, services et protocoles

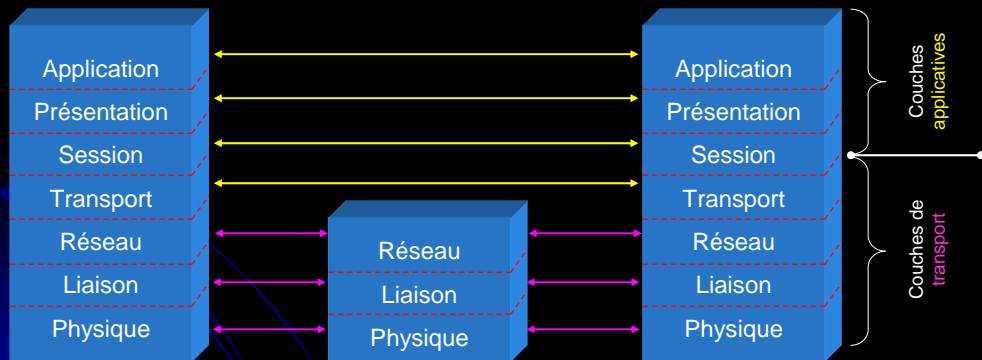
- La sémantique de fonctionnalité :
 - regroupe l'ensemble des procédures qui sont utilisées pendant le transfert de données
 - dans le cas d'une association en mode connecté, nous trouvons :
 - fragmentation réassemblage
 - concaténation séparation
 - données exprès
 - remise en séquence
 - réinitialisation
 - contrôle de flux
 - contrôle d'erreur

OSI : modèle de référence

- Open Systems Interconnection
 - développé par l'ISO (International Organization for Standardization)
 - 1995
 - Open = ouvert à la communication avec d'autres systèmes
 - décomposé en 7 couches afin de 'faciliter' et de 'simplifier'

OSI : modèle de référence

- L'architecture en couches



OSI : modèle de référence

- Principes décomposition en couches :

- une couche par niveau d'abstraction
- une couche = une fonction spécialisée
- chaque fonction doit tendre vers une normalisation des protocoles au niveau international
- limitation des informations à passer via les interfaces
- le # de couches doit être suffisamment grand pour ne pas regrouper dans une même couche des fonctions distinctes
- le # de couches doit être suffisamment limité pour ne pas rendre l'architecture trop complexe

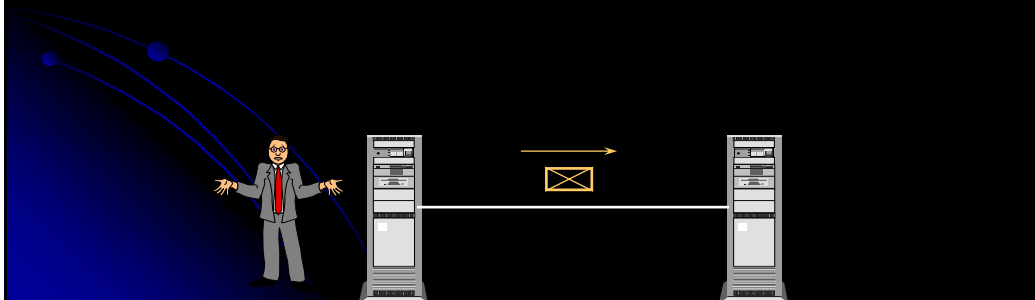
- César : DIVISER pour REGNER

OSI : couche physique

- Couche 1: physique
 - transmission de bits (état brut) sur un canal non fiable
 - doit s'assurer qu'un bit envoyé à 1 arrive bien comme 1
 - d'où :
 - quelle tension pour 1 et quelle tension pour 0 ?
 - durée (ns) d'un bit ?
 - possibilité d'émettre dans les deux sens en même temps
 - nombre de broches des connecteurs et rôles
 - ...

OSI : couche physique

- concerne surtout les interfaces mécaniques, électriques et le support physique
- réalise une **communication non fiable** entre 2 machines du **même réseau**



OSI : couche liaison de données

- Couche 2 : liaison de données
 - doit rendre fiable le moyen de communication brut (support physique). En d'autres termes, elle réalise une **communication fiable** entre 2 machines du **même réseau**
 - pour cela :
 - décomposition des données en trames de bits
 - délimitation (début et fin) de ces trames
 - envoi en séquence
 - contrôle du flux

OSI : couche liaison de données

- dans le cas d'une QoS fiable, il faut en outre :
 - utiliser des trames d'acquittement
 - gérer les tampons pour le maintien de la séquence
- une combinaison de régulation de flux et de contrôle d'erreurs (CRC) est fréquente
- Difficulté supplémentaire pour les systèmes à diffusion :
 - l'accès au canal partagé (sous-couche MAC)

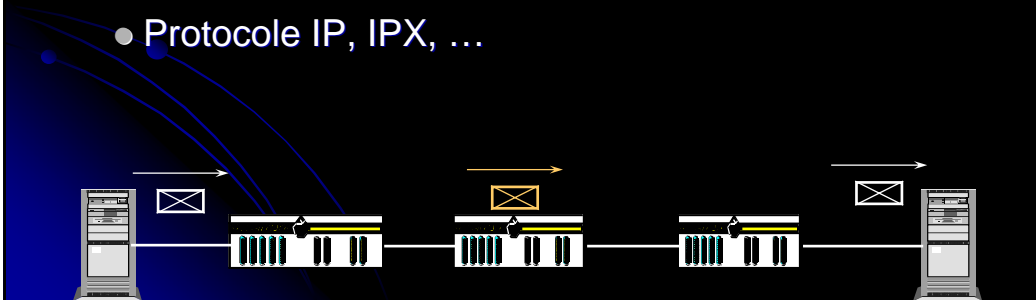


OSI : couche réseau

- Couche 3 : réseau
 - joue un rôle essentiel dans la façon dont les 'paquets' sont routés (statiques, dynamiques)
 - Notion d'adressage pour identifier le destinataire
 - gère également :
 - le flux
 - la qualité de service
 - les infos de coûts
 - ...

OSI : couche réseau

- transformation des paquets lors de passage sur des réseaux hétérogènes
- réalise une **communication non fiable** entre 2 machines distantes
- inexistant sur des réseaux à diffusion car ...
- Protocole IP, IPX, ...

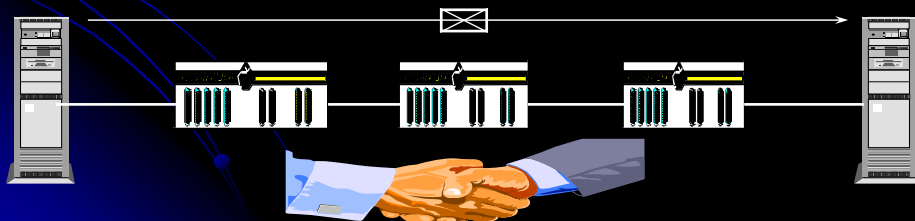


OSI : couche transport

- Couche 4 : transport
 - prend les données de la couche 5, les divise et les communique à la couche 3
 - réalise une **communication fiable** entre 2 machines distantes
 - offre une vraie notion de service de bout en bout car un programme sur la machine source dialogue 'directement' (transparent) avec son homologue sur la machine de destination

OSI : couche transport

- détermine le type de services à fournir à la couche session :
 - le canal point à point exempt d'erreur
 - remise de message isolé avec ou sans garantie d'ordre
 - multiplexage de connexions réseaux
 - diffusion de messages à plusieurs destinataires

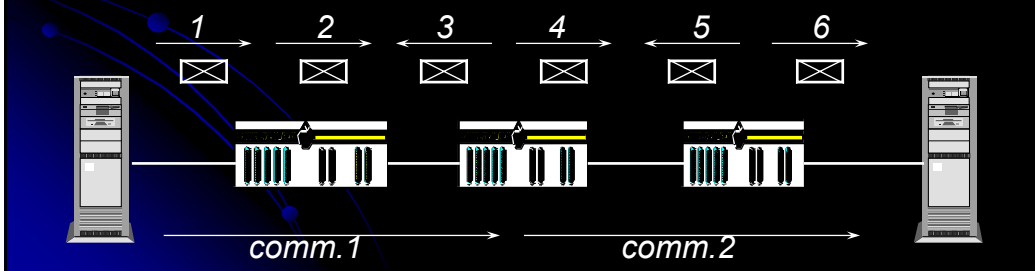


OSI : couche session

- Couche 5 : session
 - plus aucune notion de réseau :
 - le réseau, l'hétérogénéité, le routage, ... deviennent totalement transparent
 - vrai dialogue de bout en bout
 - subdivision de la communication en parties indépendantes

OSI : couche session

- une session permet :
 - la gestion du dialogue
 - la gestion du jeton (à l'alternance)
 - la synchronisation et les points de reprises pour le maintien de la connexion niveau 5 en l'absence des couches inférieures (rupture)

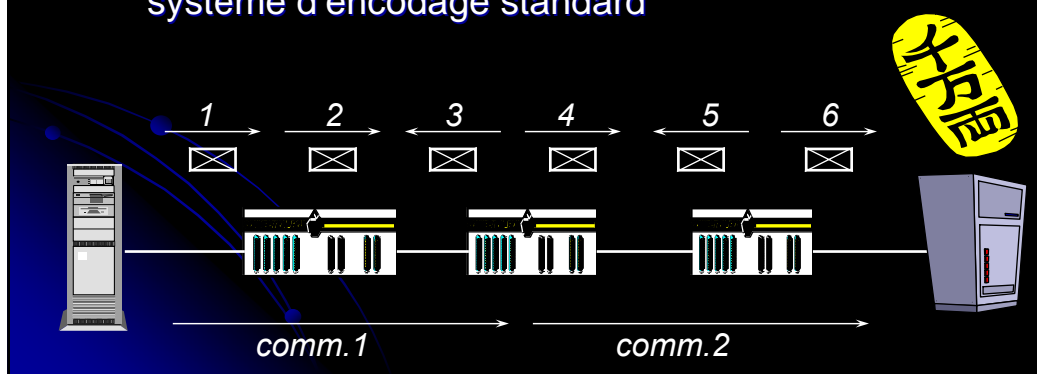


OSI : couche présentation

- Couche 6 : présentation
 - alors que les couches précédentes sont essentiellement concernées par le déplacement des bits, la couche 6 s'attache à :
 - la syntaxe
 - la sémantique
 - en effet, les ordinateurs ne travaillent pas tous avec la même représentation des données :
 - bit poids fort, poids faible, code ASCII, UNICODE, ...

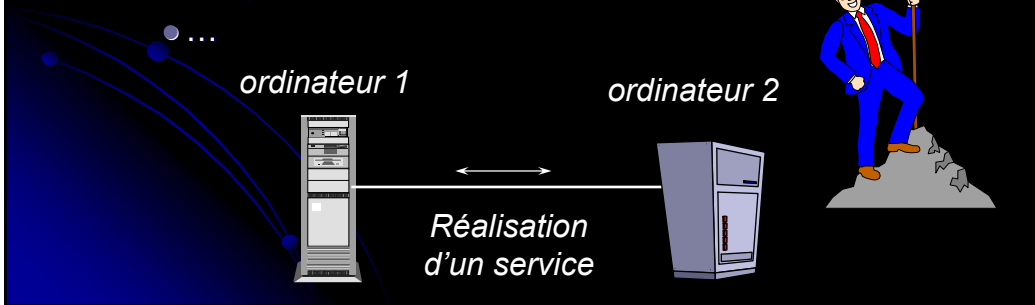
OSI : couche présentation

- la couche 6 définit donc une représentation méta (abstraction) des données.
- cette représentation est ensuite associée à un système d'encodage standard

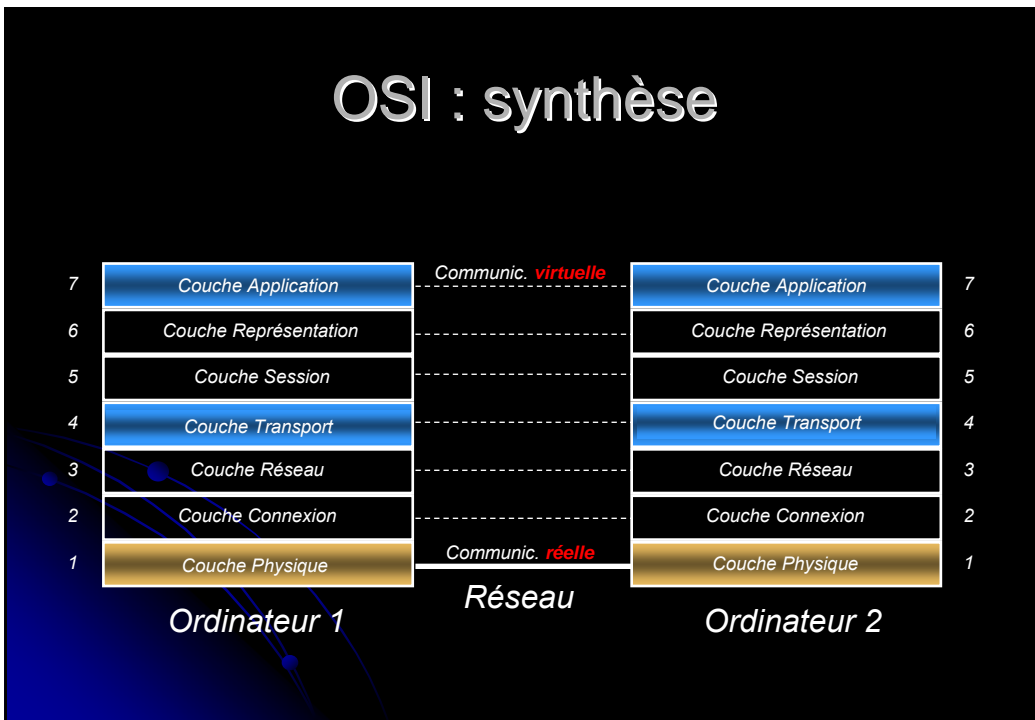


OSI : couche application

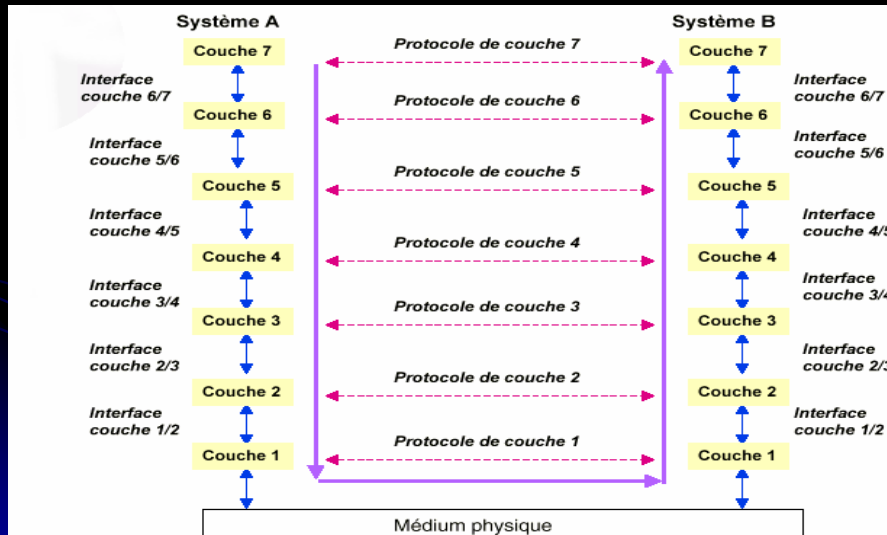
- Couche 7 : application
 - regroupe un ensemble de protocoles destinés aux utilisateurs :
 - HTTP, FTP, TELNET
 - mail, news
 - ...



OSI : synthèse



OSI : synthèse

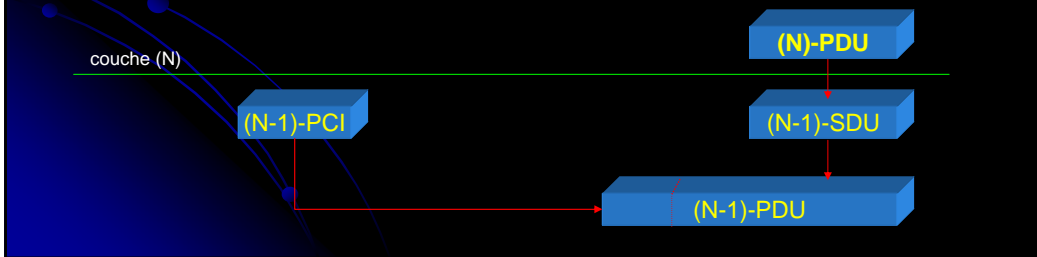


Les unités de données

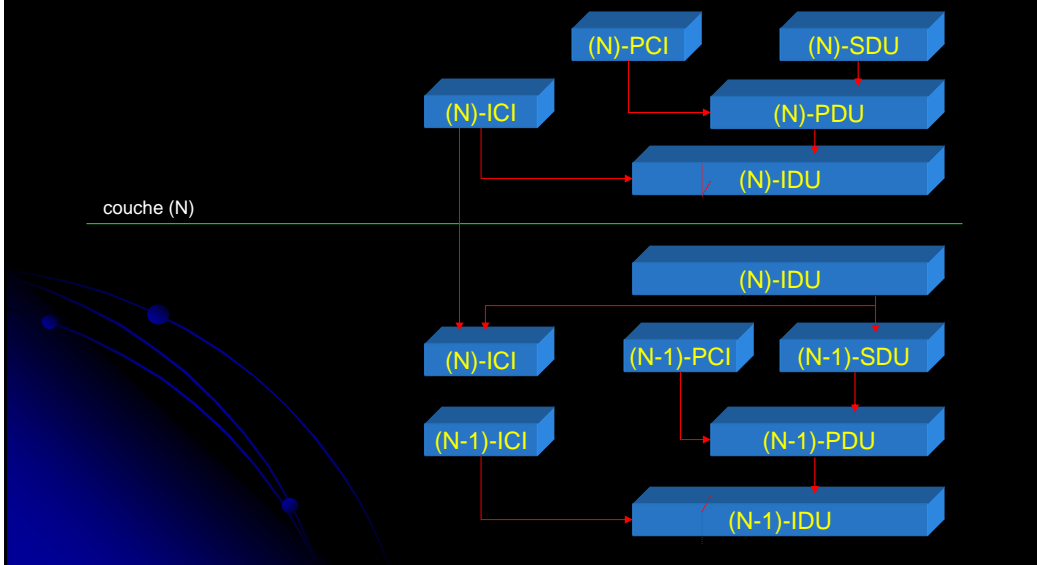
- Après les notions de services et de protocoles, voyons la structure de ce qui s'échange :
 - on appelle **N-SDU** : *Service Data Unit*
 - les unités de données du service N. C'est un ensemble de données provenant de l'interface avec la couche N
 - on appelle **N-PCI** : *Protocol Control Interface*
 - les informations de contrôle du protocole N. Elles proviennent d'entités N et sont ajoutées aux SDU sur une connexion (N-1)
 - on appelle **N-PDU** : *Protocol Data Unit*
 - les unités de données du protocole N

Les unités de données

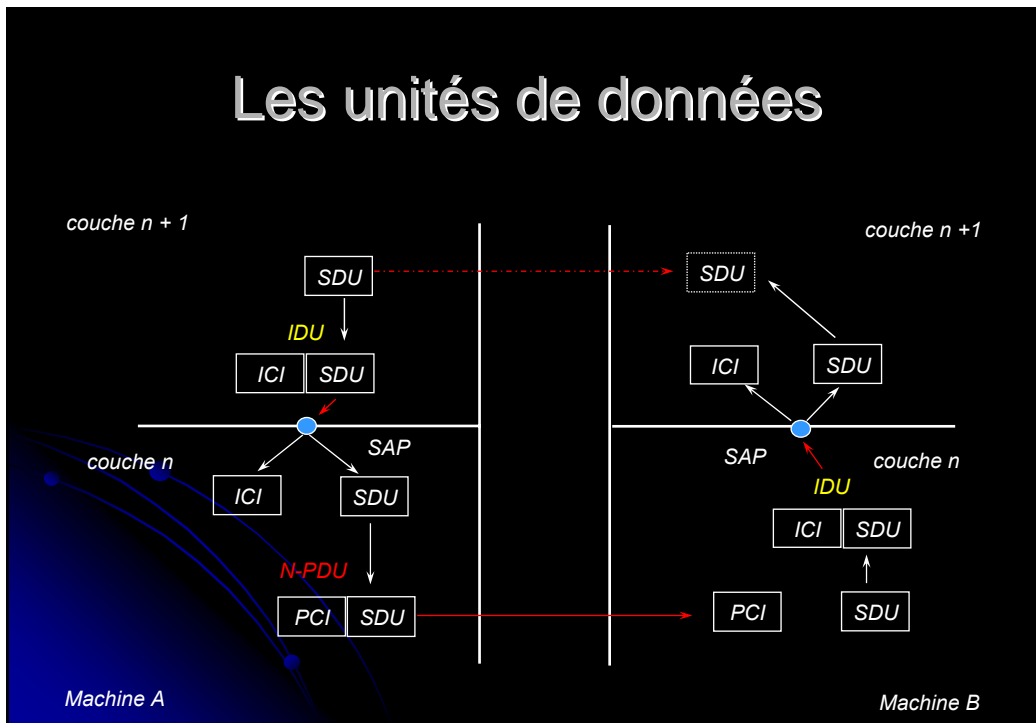
- Les (N)-PDU :
 - sont spécifiés par le protocole N. Il s'agit :
 - d'informations de contrôle du niveau N
 - d'informations provenant d'une ou plusieurs unité(s) de données de service
 - les PCI coordonnent, à un niveau donné, le travail d'échange



Les unités de données



Les unités de données



Fonctions appliquées aux UD

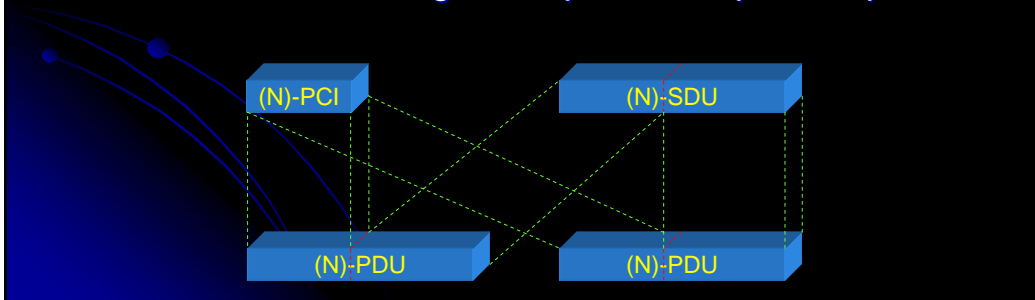
- Trois grandes catégories de **fonctions** peuvent être appliquées aux Unités de Données :
 - segmentation réassemblage
 - groupage dégroupage
 - concaténation séparation
- Il appartient au réseau de **découvrir** la meilleure taille à appliquer aux UD afin de fonctionner de manière optimale
 - les fonctions citées ci-dessus permettent d'atteindre la taille choisie

Fonctions appliquées aux UD

- segmentation réassemblage

mise en correspondance d'une unité de données du service N avec plusieurs unités de données du protocole N.

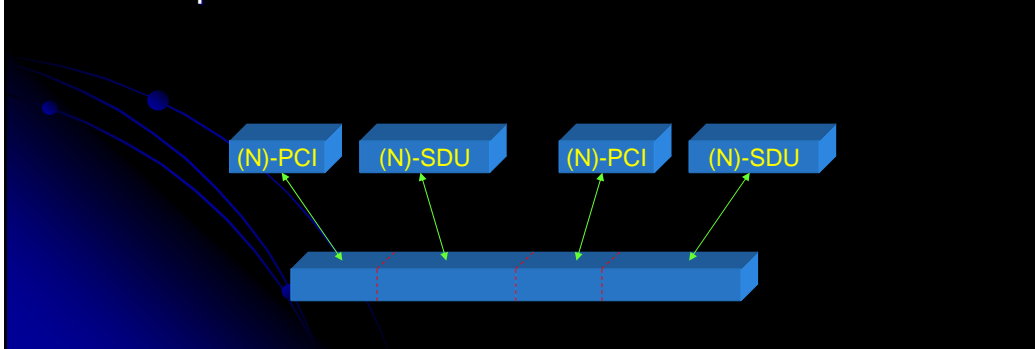
Le nombre de fragments peut être quelconque



Fonctions appliquées aux UD

- groupage dégroupage

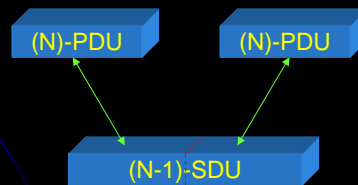
mise en correspondance de plusieurs unités de données du service N avec une unité de données du protocole N



Fonctions appliquées aux UD

- concaténation séparation

mise en correspondance, par l'entité N, de plusieurs unités de données de protocoles avec une unité de données du service (N-1)



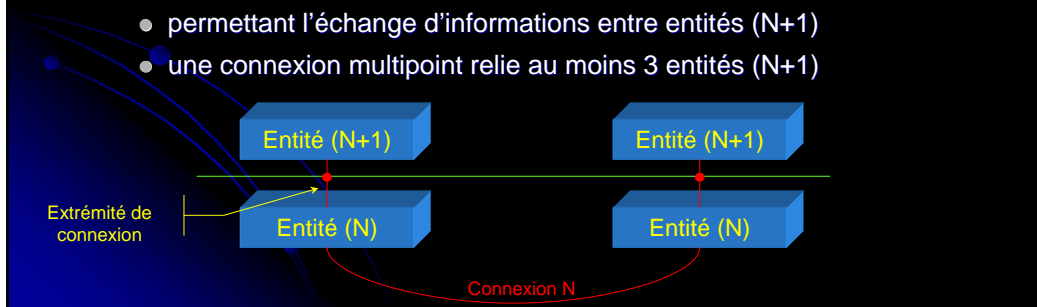
Éléments nécessaires au fonctionnement d'une couche

- Parmi les fonctionnalités permettant le traitement des entités qui traversent les couches, nous exposons :
 - les connexions
 - le contrôle de flux
 - le contrôle d'erreur

Les connexions

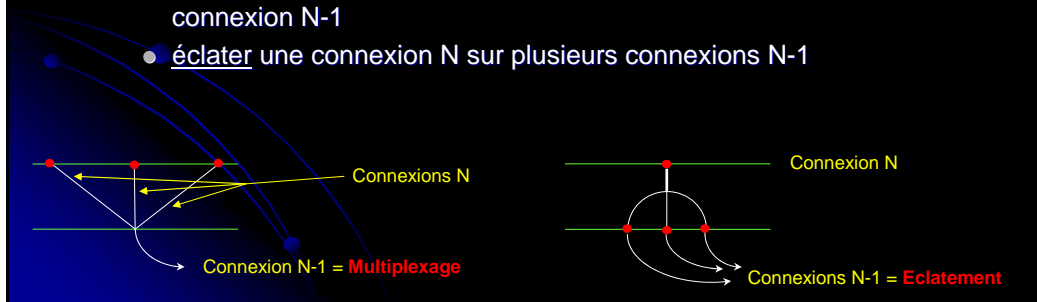
Une connexion N est une association établie pour permettre la communication entre au moins deux entités (N+1) identifiées par leur adresse N. Une connexion est :

- un service offert par la couche N
- une connexion N possède au moins 2 extrémités N qui associent les entités
- permettant l'échange d'informations entre entités (N+1)
- une connexion multipoint relie au moins 3 entités (N+1)



Les connexions

- A une extrémité de connexion N correspond une adresse N. Pour communiquer, il faut que les entités disposent :
 - des mêmes éléments de protocoles
 - d'une connexion N-1
- Pour optimiser une connexion, il est possible de :
 - multiplexer plusieurs connexions N sur une seule et même connexion N-1
 - éclater une connexion N sur plusieurs connexions N-1



Le contrôle de flux

- Rôle :
 - cadencer l'envoi des PDU afin de permettre à l'entité homologue de les récupérer à une vitesse lui convenant
 - notons qu'il est possible d'effectuer un contrôle de flux à l'interface entre deux couches

Le contrôle d'erreur

- Rôle :
 - erreurs sur la connexion
 - erreurs dues aux protocoles traversés
 - pertes d'informations
- Moyen :
 - utilisation d'accusés de réception inclus dans les (N)-PCI

Modèle TCP/IP

- Origine :
 - 1970, DoD
 - dans le but d'unifier un ensemble de machines utilisant des protocoles différents et incompatibles
- 2 parties :
 - **IP** : *Internet Protocol* : niveau réseau, service sans connexion
 - **TCP** : *Transmission Control Protocol* : niveau transport, service fiable avec connexion

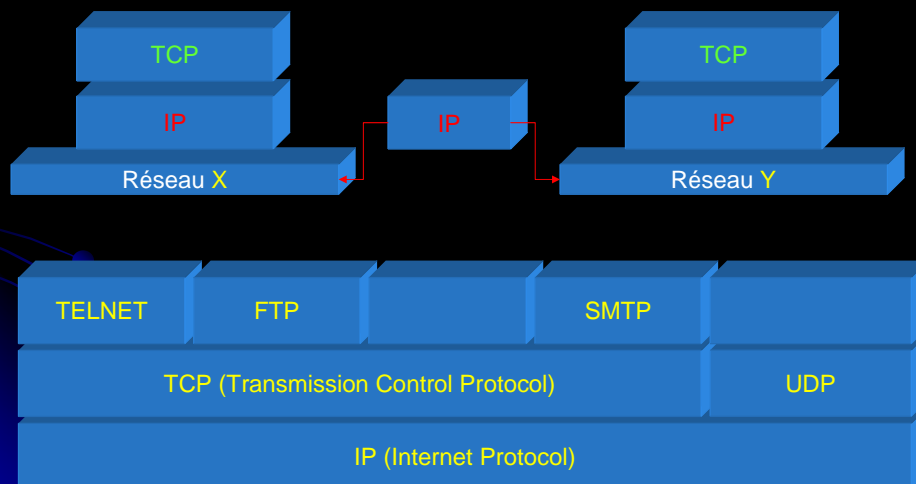
Modèle TCP/IP

- Architecture en couches (// OSI)
 - fondé sur le protocole IP, correspond au niveau 3 (paquet) de OSI
 - partiellement :
 - IP est conçu comme protocole d'interconnexion
 - c'est un bloc de données dont le format est bien défini et contenant une adresse
 - mais aucune autre fonctionnalité en version IPv4
 - plusieurs nouveautés en IPv6 :
 - sécurité, adressage, compression, QoS, ...

Modèle TCP/IP

- les paquets sont indépendants les uns des autres et sont routés par des routeurs.
- la QoS de IP est très faible :
 - pas de détection de pertes
 - pas de possibilité de reprise sur erreur
- la QoS de TCP peut être beaucoup plus grande :
 - il peut résoudre tous les cas d'erreurs des niveaux inférieurs

Modèle TCP/IP



Comparaison OSI / TCP

- Les modèles OSI et TCP/IP ont beaucoup de points communs :
 - reposent sur le concept de pile de protocoles indépendants
 - leurs couches sont fonctionnellement très proches
 - distinction couches inférieures/supérieures
 - les couches supérieures offrent un service de transport indépendant du réseau
 - ...

Comparaison OSI / TCP

- Les différences sont également nombreuses :
 - OSI (modèle de référence) a clairement distingué les concepts de :
 - services, interfaces, protocoles
 - chaque couche rend des services à la couche au-dessus d'elle.
 - La définition du service indique **ce que fait** la couche, **pas comment** elle le fait, **ni comment** les couches supérieures y accèdent
 - le service spécifie donc la sémantique d'une couche

Comparaison OSI / TCP

- L'interface d'une couche indique aux processus de la couche supérieure **comment accéder** à la cette couche mais pas comment elle fait sont travail
- Le protocole N est strictement privé à la couche N. Tant qu'elle accompli sa tâche (service), la couche N utilise le protocole qu'elle veut.

Comparaison OSI / TCP

- TCP/IP n'a jamais clairement distinguer ces 3 concepts.
- Historiquement, OSI est **arrivé avant** les protocoles. Il a donc l'avantage d'être très général.
- Pour TCP/IP, l'histoire fût exactement l'inverse. Il est arrivé après les protocoles. En fait, TCP/IP s'est contenté de décrire l'existant.

Comparaison OSI / TCP

- Autre différence :
 - OSI a 7 couches (modèle fortement détaillé)
 - TCP/IP en compte 5 (modèle très pragmatique)
 - alors qu'OSI a voulu tout détaillé, TCP/IP est allé à l'essentiel en évitant les lourdeurs excessives
- Citons encore :
 - le fait qu'OSI autorise les deux types de communication dans la couche réseau mais un seul (mode connecté) dans la couche transport, alors que TCP/IP inverse cela

Critique du modèle/protocoles OSI

- Quatre causes principales :
 - moment inopportun
 - les protocoles de TCP/IP existaient déjà (dans les univ.) quand le modèle OSI s'est présenté. Les premières réalisations ont donc été faites pour TCP. Aucune ne le fût pour OSI car les fabricants n'ont pas voulu se lancer sur une seconde pile en même temps
 - mauvaise technologie
 - OSI n'était pas exempt d'imperfections. Le choix des 7 couches a surtout été politique. 2 d'entre elles (session et présentation étaient quasi vides).
 - de plus, certaines fonctions (contrôle d'erreurs, adressage, ...) sont présentes à plusieurs niveaux

Critique du modèle/protocoles OSI

- mauvaise implémentation
 - l'énorme complexité du modèle et des protocoles a engendré une très grande lourdeur des premières implémentations. Pour TCP, qui était gratuit de surcroît, ce fût tout le contraire
- mauvaise stratégie
 - TCP à été quasi immédiatement associé, dans l'esprit des gens, à UNIX et au monde universitaire
 - par contre, OSI à été assimilé aux administrations européenne et américaine
 - la réticence pour un système « technocrate » à donc joué en faveur de TCP

Critique du modèle/protocoles TCP

- TCP et ses protocoles ont aussi leurs problèmes :
 - rappel : la **non différenciation** claire entre les services, les interfaces et les protocoles. La spécification n'est pas clairement distincte de l'implémentation. TCP ne peut pas facilement servir de base à la spécification de nouveaux réseaux basés sur des technologies neuves
 - le **modèle** TCP ne décrit correctement que l'implémentation ... TCP. S'en servir pour décrire Bluetooth est impossible

Critique du modèle/protocoles TCP

- les couches physique et liaison de données ne sont même **pas évoquées**. Elles sont pourtant radicalement différentes l'une de l'autre
- si les protocoles IP et TCP ont été soigneusement définis et implémentés, le développement des autres protocoles (couche applicative, ...) est nettement plus brouillon. Ce fût souvent le fruit de travaux de doctorat dont les implémentations gratuites furent largement diffusées. C'est pour cette raison qu'elles existent toujours en l'état :
 - ex : telnet : conçu pour un terminal Télétype à 10 caractères par seconde !!!

Résultat OSI vs TCP

- Le **modèle** OSI :
 - hormis les couches 5 et 6, s'est révélé très utile pour l'analyse des réseaux. Par contre, les **protocoles** n'ont jamais fait l'objet d'une large acceptation
- Pour **TCP/IP** ce fût tout le contraire :
 - le modèle n'existe pour ainsi pas alors que les protocoles sont déployés partout