

# Chapitre 4 : Architecture générale des réseaux informatiques

/udd/bcousin/Pages-web-Armor/Enseignement/Reseaux-generalites/Cours/4.recover.fm - 1 juin 2001 18:28

## Plan

- 1. Introduction p95
- 2. Le modèle de référence OSI de l'ISO p98
- 3. Quelques fonctions p112
- 4. Autres exemples d'architecture de réseaux p118
- 5. Normalisation des réseaux p120
- 6. Conclusion p123
- 7. Quelques commentaires p124s

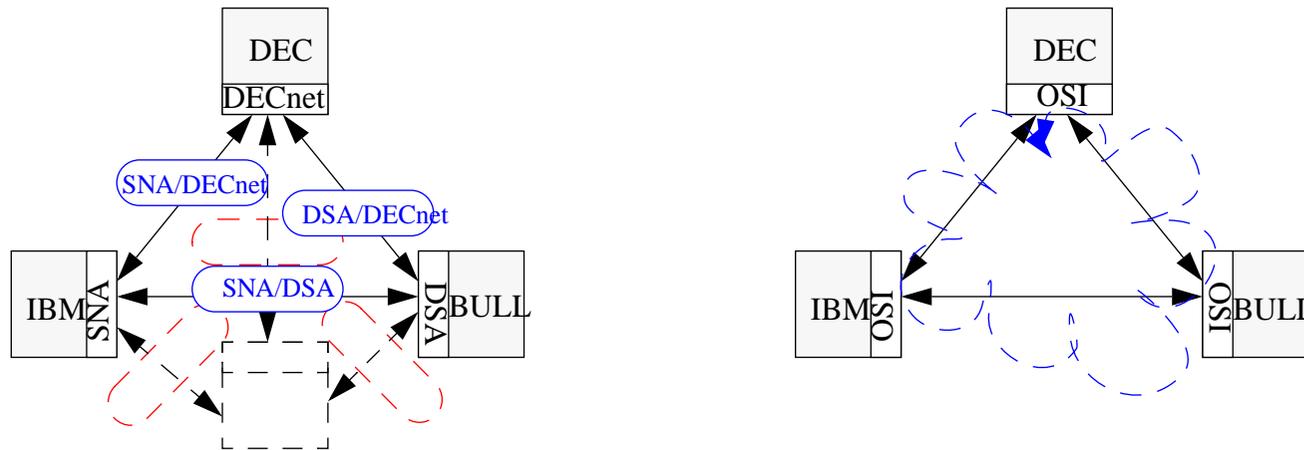
## Bibliographie

- Reference model for open systems interconnection : ISO 7498, UIT-T.200
- G. Pujolle, Les réseaux, Eyrolles, 1995. Chapitre 3.
- H. Nussbaumer, Téléinformatique, Presses Polytechniques romandes, 1987. Chapitre 1.3.
- A.Tanenbaum, Réseaux, 3<sup>ème</sup> éd., InterEditions, 1997. Chapitre 1.

## 1. Introduction

Les réseaux informatiques doivent permettre à des applications informatiques de coopérer sans avoir à tenir compte de l'hétérogénéité des moyens et procédés de transmission (par exemple : de la topologie, des méthodes d'accès, des caractéristiques des équipements ou des supports, etc.).

- . Adapter la technologie de transmission au support de communication
  - . Masquer les phénomènes altérant la transmission
  - . Maintenir la qualité demandée
  - . Offrir l'interopérabilité (1)
  - . Optimiser l'utilisation des ressources (2)
  - . Assurer la pérennité des choix (3)
- (1) + (2) + (3) ==> **Normalisation**



## 1.1. Objectif

Réduire la complexité de conception des réseaux informatiques.

Principes :

- démarche analytique : recensement des fonctions nécessaires
- démarche synthétique : classement des fonctions
- démarche simplificatrice et constructive :
  - . regroupement en sous-ensembles pour simplifier la compréhension des fonctions (frontières précises, concises et utiles)
  - . décomposition hiérarchique de l'ensemble des mécanismes à mettre en œuvre en une série de **couches** (ou niveaux).

Remarque :

**Le nombre de couches, leurs noms et leurs fonctions** varient selon les types de réseaux

Exemples :

le modèle de référence de l'OSI : 7 couches

LAN : 2 + 2 sous-couches; ATM : 2 + 1 + 3 sous-couches, Internet : 3 ou 4 couches

## 1.2. Principes de base de la décomposition en couches

Une couche doit être créée lorsqu'un nouveau niveau d'abstraction est nécessaire

Chaque couche exerce une fonction bien définie

Les fonctions de chaque couche doivent être choisies en pensant à la définition des protocoles normalisés internationaux

Le choix des frontières entre couches doit minimiser le flux d'informations aux interfaces

Le nombre de couches doit être :

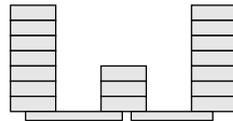
- suffisamment grand pour éviter la cohabitation dans une même couche de fonctions très différentes, et
- suffisamment petit pour éviter que l'architecture ne devienne difficile à maîtriser.

## 2. Le modèle de référence OSI de l'ISO

### 2.1. Présentation

Norme de description de l'architecture générale des réseaux informatiques:

- L'OSI = "Open Systems Interconnection : reference model"
- modèle en 7 couches



Les noms de la norme :

- ISO : IS 7498
- CCITT : X200 (nouvellement IUT-T)
- AFNOR : NF.Z.70.001

## 2.2. Notion de couche, de protocole et de service

Une **couche** est spécialisée dans un ensemble de fonctions particulières. Elle utilise les fonctionnalités de la couche inférieure et propose ses fonctionnalités à la couche supérieure.

Un **système** est un ensemble de composants formant un tout autonome.

Une **entité** est l'élément actif d'une couche dans un système.

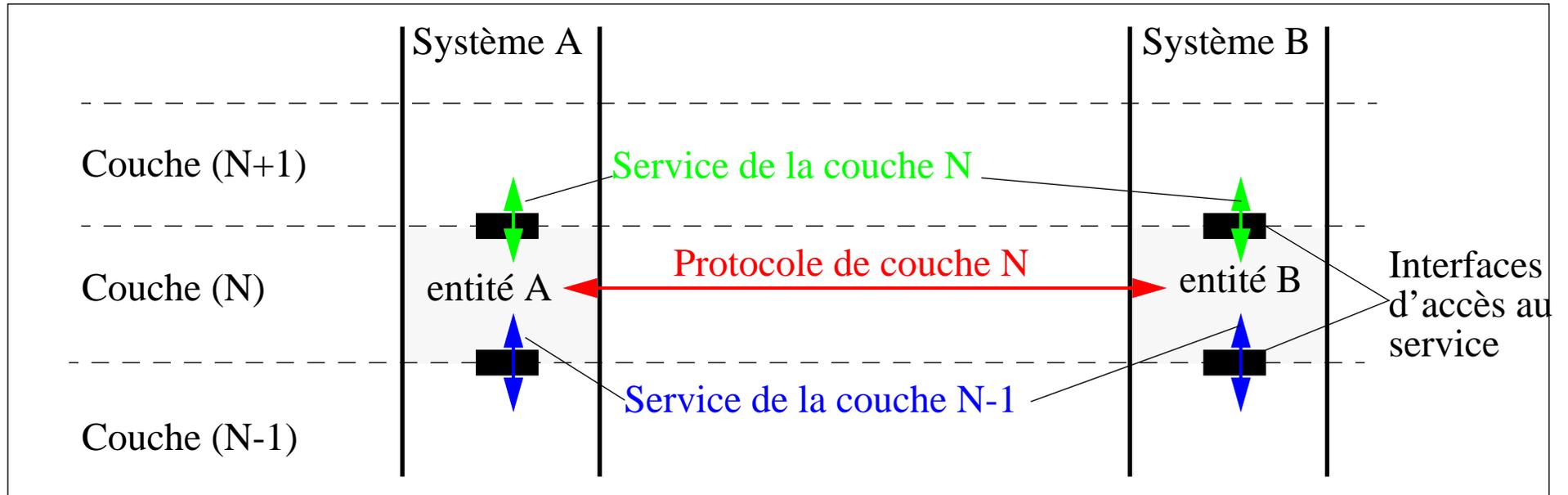
- entités homologues (paires) : entités de même couche situées dans des systèmes distants

Le **protocole** d'une couche N définit l'ensemble des **règles** ainsi que les **formats** et la signification des objets échangés, qui régissent la communication entre les entités de la couche N.

Le **service** d'une couche N définit l'ensemble des fonctionnalités possédées par la couche N et fournies aux entités de la couche N+1 à l'interface N/N+1.

**Notation** : on note  $N\_X$  (ou encore  $X(N)$ ) l'objet de type X appartenant à la couche N.

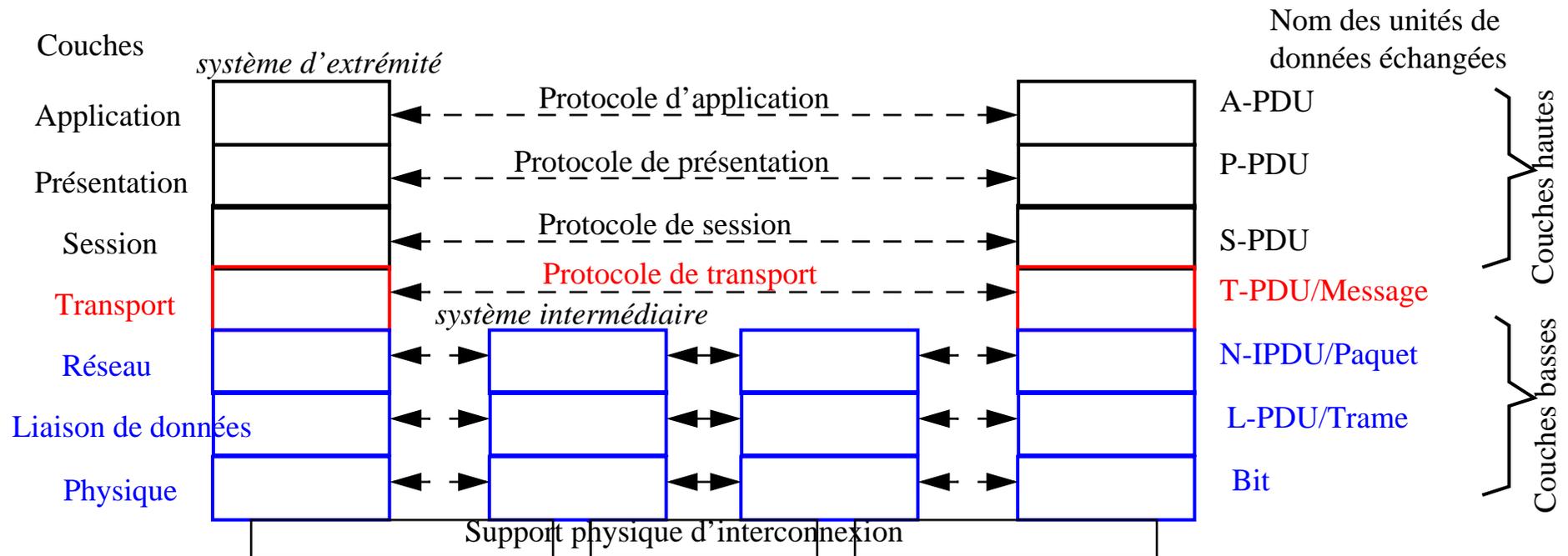
- ex : N-entité ou entité(N)



L'architecture d'un réseau est définie par l'ensemble des couches et la description des protocoles et des services de chacune d'elles.

### 2.3. Architecture générale du modèle OSI

Le modèle OSI possède sept couches



Le modèle décrit simplement ce que chaque couche doit réaliser (le service), les règles et le format des échanges (le protocole), mais pas leur implantation.

## 2.4. Les sept couches du modèle OSI

### 2.4.1 La couche Physique (couche 1)

Fournit les moyens mécaniques, optiques, électroniques, fonctionnels et procéduraux nécessaires à l'activation, au maintien et à la désactivation des connexions physiques nécessaires à la transmission de trains de bits.

Note : les systèmes sont interconnectés réellement au moyen de supports physiques de communication. Ces derniers ne font pas partie de la couche Physique.

### 2.4.2 La couche Liaison de données (couche 2)

Assure la transmission d'informations entre (2 ou plusieurs) systèmes immédiatement **adjacents**. Détecte et corrige, dans la mesure du possible, les erreurs issues de la couche inférieure. Les objets échangés sont souvent appelés **trames** ("frames").

### 2.4.3 La couche Réseau (couche 3)

Achemine les informations à travers un **réseau** pouvant être constitué de systèmes intermédiaires (routeurs). Les objets échangés sont souvent appelés **paquets** ("packets").

#### 2.4.4 La couche Transport (couche 4)

Assure une transmission de **bout en bout** des données. Maintient une certaine qualité de la transmission, notamment vis-à-vis de la fiabilité et de l'optimisation de l'utilisation des ressources. Les objets échangés sont souvent appelés **messages** (de même pour les couches supérieures).

#### 2.4.5 La couche Session (couche 5)

Fournit aux entités coopérantes les moyens nécessaires pour **synchroniser** leurs dialogues, les interrompre ou les reprendre tout en assurant la cohérence des données échangées.

#### 2.4.6 La couche Présentation (couche 6)

Se charge de la **représentation** des informations que les entités s'échangent. Masque l'hétérogénéité de techniques de codage utilisées par les différents systèmes.

#### 2.4.7 La couche Application (couche 7)

Donne aux processus d'application les moyens d'accéder à l'environnement de communication de l'OSI. Comporte de nombreux protocoles **adaptés** aux différentes classes d'application.

Note : les fonctionnalités locales des applications proprement dites sont hors du champ de l'OSI donc de la couche Application !

## 2.4.8 Conclusion

Les trois premières couches constituent les couches basses où les contraintes du réseau sont perceptibles. Fonctions élémentaires spécialisées dans la transmission.

La couche Transport est une couche charnière, d'adaptation ou intermédiaire, associée le plus souvent aux couches basses.

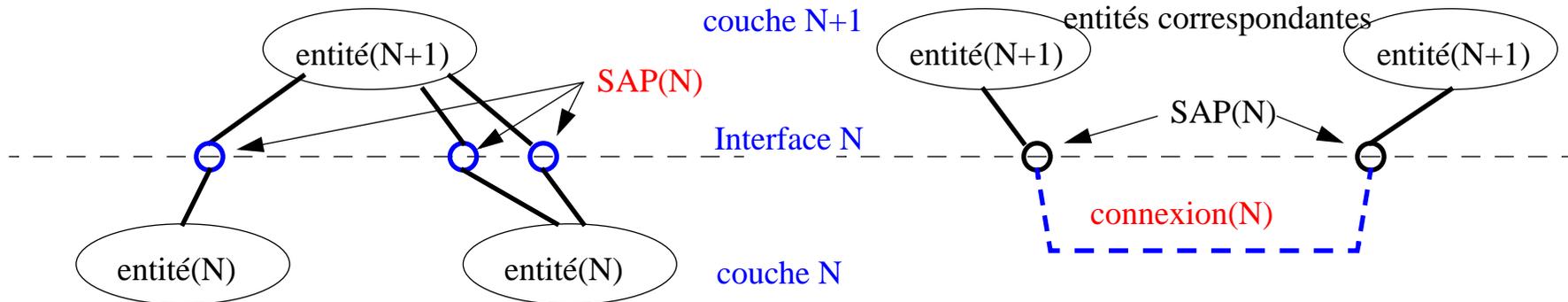
Les trois dernières couches constituent les couches hautes où les contraintes de l'application sont perceptibles. Fonctions complexes et variables adaptées aux traitements applicatifs.

**Attention** : La norme stipule clairement qu'il s'agit d'un modèle de référence et par conséquent, suivant le contexte dans lequel on se trouve et les besoins de communication, certaines fonctionnalités de certaines couches peuvent ne pas être utilisées (protocoles alternatifs, classes de protocole, options, etc.).

## 2.5. Connexion

**SAP(N)** : “service access point”

- point identifié où les services sont fournis par l’entité(N) à une entité(N+1)
- . ex : adresse



**Connexion(N)** : association d’entités homologues pour le transfert de données

- entités correspondantes : entités associées par la même connexion

Extrémité de connexion(N) : terminaison d’une connexion(N) à un SAP(N).

- connexion **bipoint** : connexion comportant exactement 2 extrémités.
- connexion **multipoint** : connexion comportant plus de 2 extrémités.

## 2.6. Les modes de communication

On peut distinguer deux grands modes de communication:

- communication en **mode connecté** (appelé aussi “with connection”)
- communication en **mode non connecté** (appelé aussi “connectionless” ou par abus de langage “datagramme”)

### 2.6.1 Le mode non connecté

- 1 seule phase (ou 0!) :
  - . le transfert de données
- chaque unité de transfert de données est acheminée indépendamment
- les entités communicantes ne mémorisent rien (“memoryless”).
- les messages échangés sont auto-suffisants (“self-content”)

## 2.6.2 Le mode connecté

- 3 phases :
  - . phase d'**établissement** de la connexion
  - . phase de **transfert** de données
  - . phase de **libération** de la connexion
- un **contexte** (réparti) est partagé par les membres de la connexion :
  - . par exemple : le numéro du paquet
- permet (facilite) le contrôle et la gestion du transfert de données :
  - . contrôle d'erreur, contrôle de flux, maintien en séquence, etc.
- les messages échangés comportent des informations qui ne sont utilisables que grâce à la connaissance de ce contexte :
  - . par exemple : le numéro de paquet / la largeur de la fenêtre coulissante

## 2.7. Les primitives de service

Le protocole(N) s'appuie sur le service(N-1), donc la description du service est nécessaire à la compréhension du protocole.

Attention, le service n'est accessible qu'à l'intérieur d'un système : la façon d'accéder au service **ne doit pas** être normalisée !

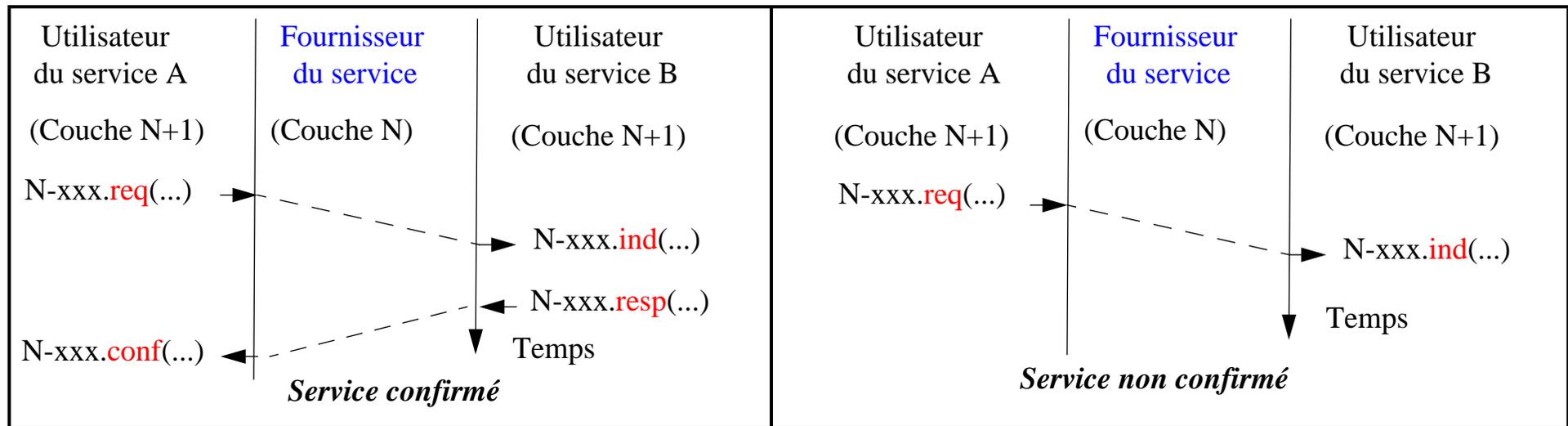
Les primitives spécifient le service :

- Ce sont des objets abstraits (puisque le service est **abstrait** !)
- Echangées à travers un interface idéal (sans perte et sans délai)
- Leur représentation ressemble à une procédure avec des paramètres
  - . préfixe : l'initiale de la couche
  - . suffixe : le type de primitive
  - . Exemples :
    - . T\_Data.req(T\_SDU)
    - . LLC\_Data.req(ad-locale, ad-distante, L\_SDU, classe-de-service)

Il existe quatre types de primitives :

- **Request** : Une entité sollicite un service
- **Indication** : Une entité est informée d'une demande de service
- **Response** : Une entité a rendu le service, si possible
- **Confirmation** : Une entité est informée que le service a été rendu

Par exemple :



De très nombreuses variantes d'enchaînement des types de primitives de service [1 à 4] existent !

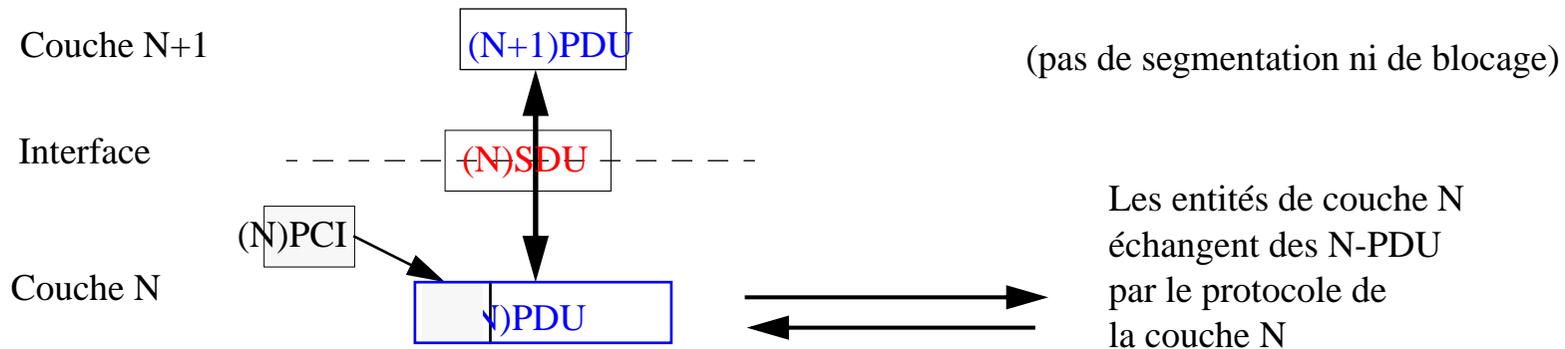
## 2.8. Les unités de données

### SDU(N) :

- unité de données spécifique au **service(N)**, dont l'intégrité est préservée d'une extrémité à l'autre d'une connexion. Mais pas forcément entre !
  - . potentiellement de taille quelconque.
  - . exemple : un fichier à transmettre

### PDU(N) :

- unité de données spécifique au **protocole(N)**, adaptée à la transmission, constituée par les informations de contrôle du protocole (PCI(N)) et éventuellement par des portions de données issues du SDU(N)
  - . par ex. : une trame, un paquet.



### IDU(N) :

- unité d'information transférée en une seule interaction à l'**interface** de 2 couches, constituée d'information de contrôle d'interface (ICI(N)) et tout ou partie d'une SDU(N).
  - . dépend du système d'accueil (notamment leur format).
  - . dépendant de l'implantation
  - . par ex. : les tampons (buffers) utilisés pour charger le fichier

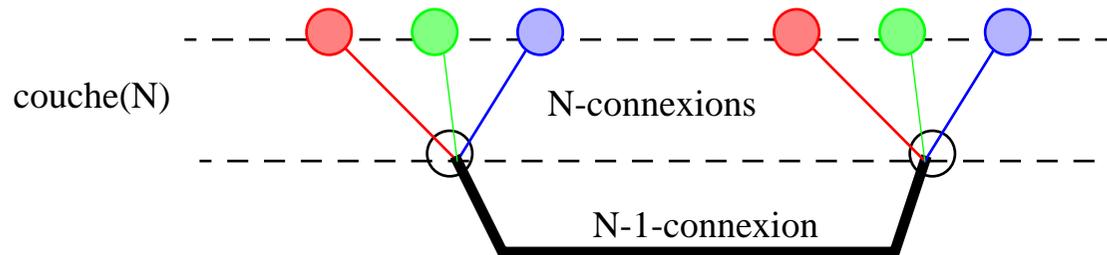
### 3. Quelques fonctions

Ces fonctions peuvent être présentes dans n'importe quelle couche.

#### 3.1. Multiplexage/démultiplexage

Fonction d'une couche(N) permettant de prendre en charge plusieurs connexions(N) sur une seule connexion(N-1)

- Optimise l'utilisation de la connexion(N-1)
- Permet l'établissement simultané de plusieurs connexions(N) alors qu'une seule connexion(N-1) existe
- Problème : identification des connexions, contrôle des différents flux

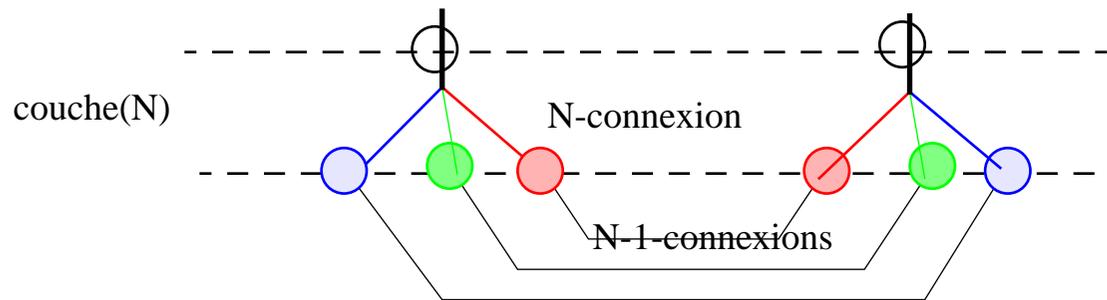


Exemple : multiplexage de plusieurs connexions de Transport sur une seule connexion de Réseau

### 3.2. Eclatement/recombinaison

Fonction d'une couche(N) permettant d'utiliser plusieurs connexions(N-1) pour prendre en charge une connexion(N).

- Amélioration de la fiabilité grâce à la redondance
- Amélioration des performances malgré l'usage de connexions(N-1) peu performantes
- Problème : dé-séquencement



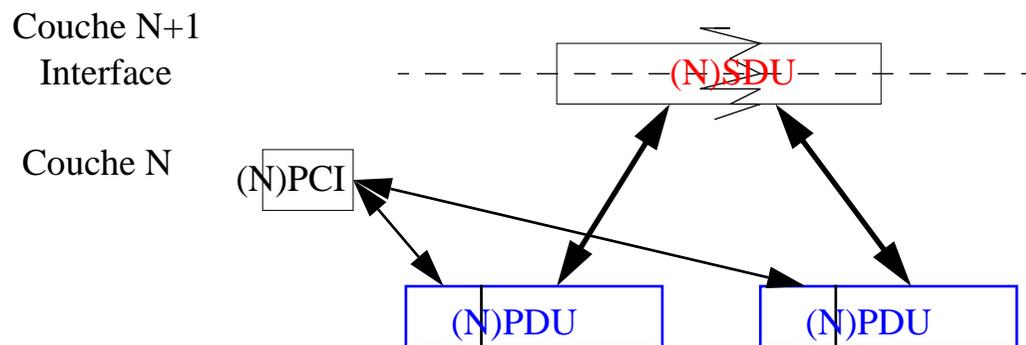
- Appelé aussi multiplexage aval.

Exemple : HDLC multiliasion (“multilink”)

### 3.3. Segmentation/réassemblage

Fonction d'une couche(N) mettant en correspondance une SDU(N) avec plusieurs PDU(N)

- Adaptation de la taille des données (N-SDU) aux caractéristiques de transmission (N-PDU)
- Problème : identification des PDU transportant les données constituant la SDU.

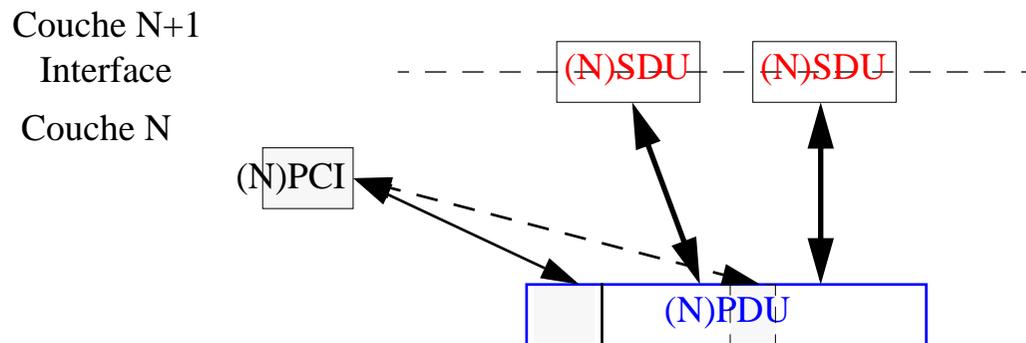


Exemple : les fragments d'un datagramme IP

### 3.4. Groupage/dégroupage

Fonction d'une couche(N) mettant en correspondance plusieurs SDU(N) avec une seule PDU(N)

- Adaptation de la taille des données (N-SDU) aux caractéristiques de la transmission (N-PDU)
- Diminution du surcoût (overhead)
- Problème : identification des SDU transportées.

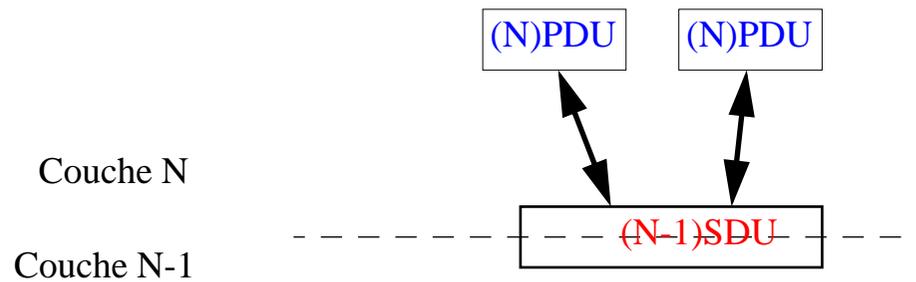


Exemple : la “bufferisation” des données sous TCP

### 3.5. Concaténation/séparation

Fonction d'une couche (N) mettant en correspondance plusieurs PDU(N) avec une seule SDU(N)

- Adaptation de la taille des données aux caractéristiques du service (N-SDU)
- Diminution du surcoût (overhead)
- Problème : identification des PDU transportées

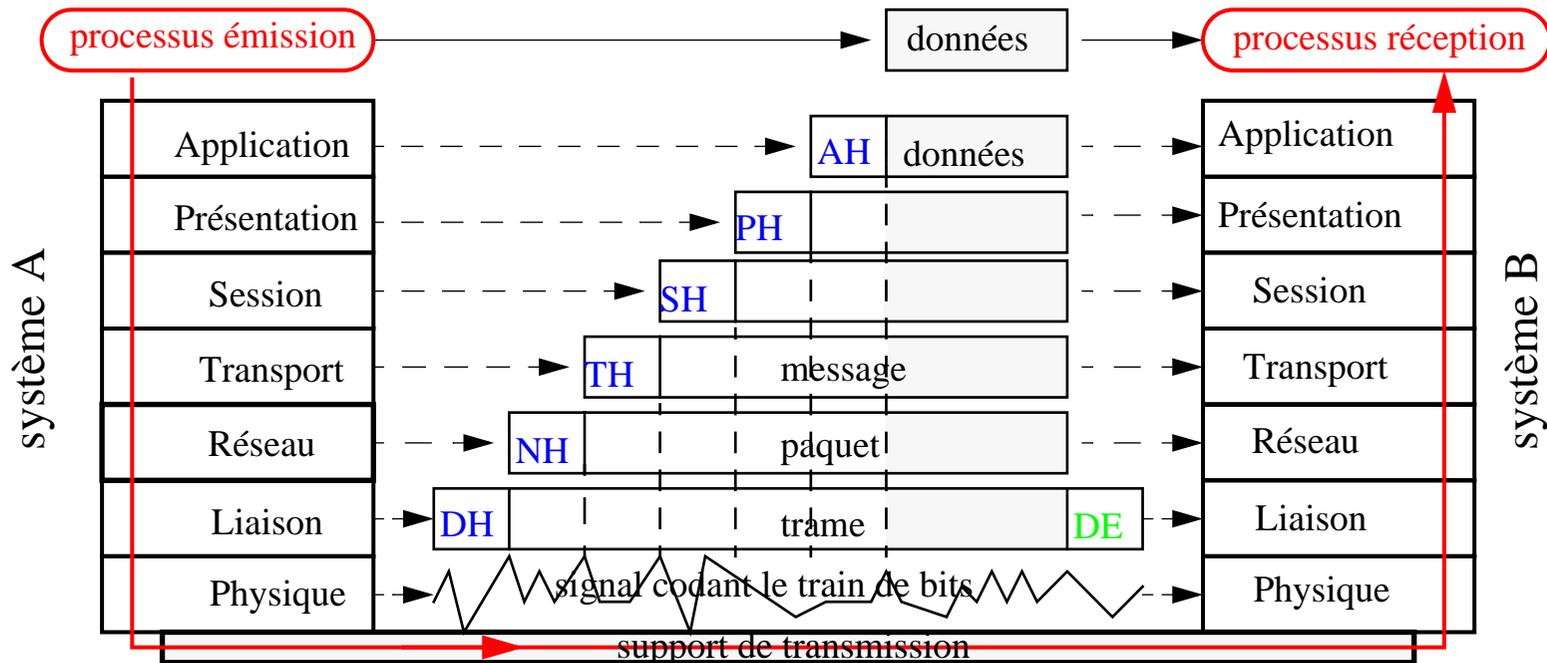


Exemple : Concaténation des commandes et des données de la couche Transport.

### 3.6. Transmission de données

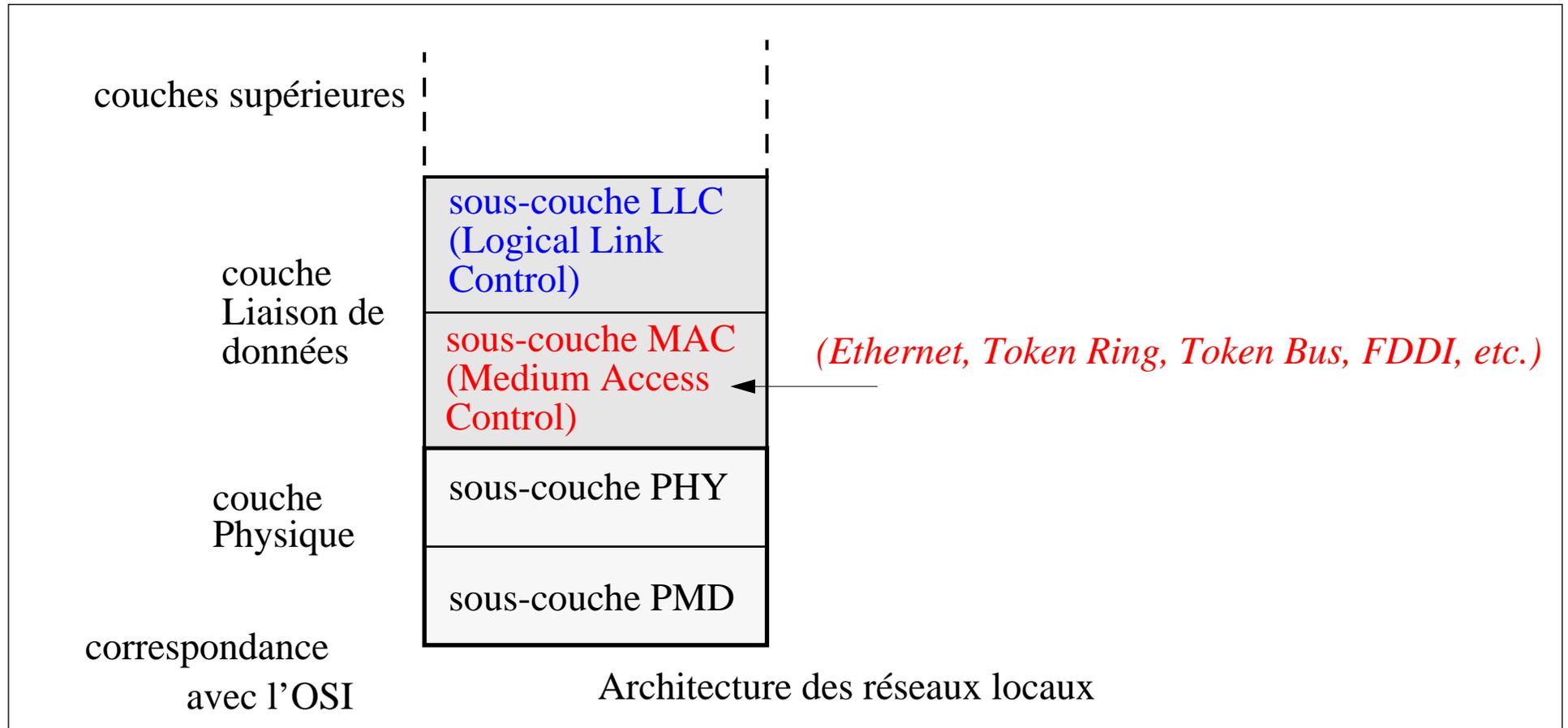
L'encapsulation :

- Les données d'une couche sont encapsulées dans une unité de données de la couche inférieure.
- Par ex. : la lettre dans l'enveloppe dans le sac postal dans le train postal.

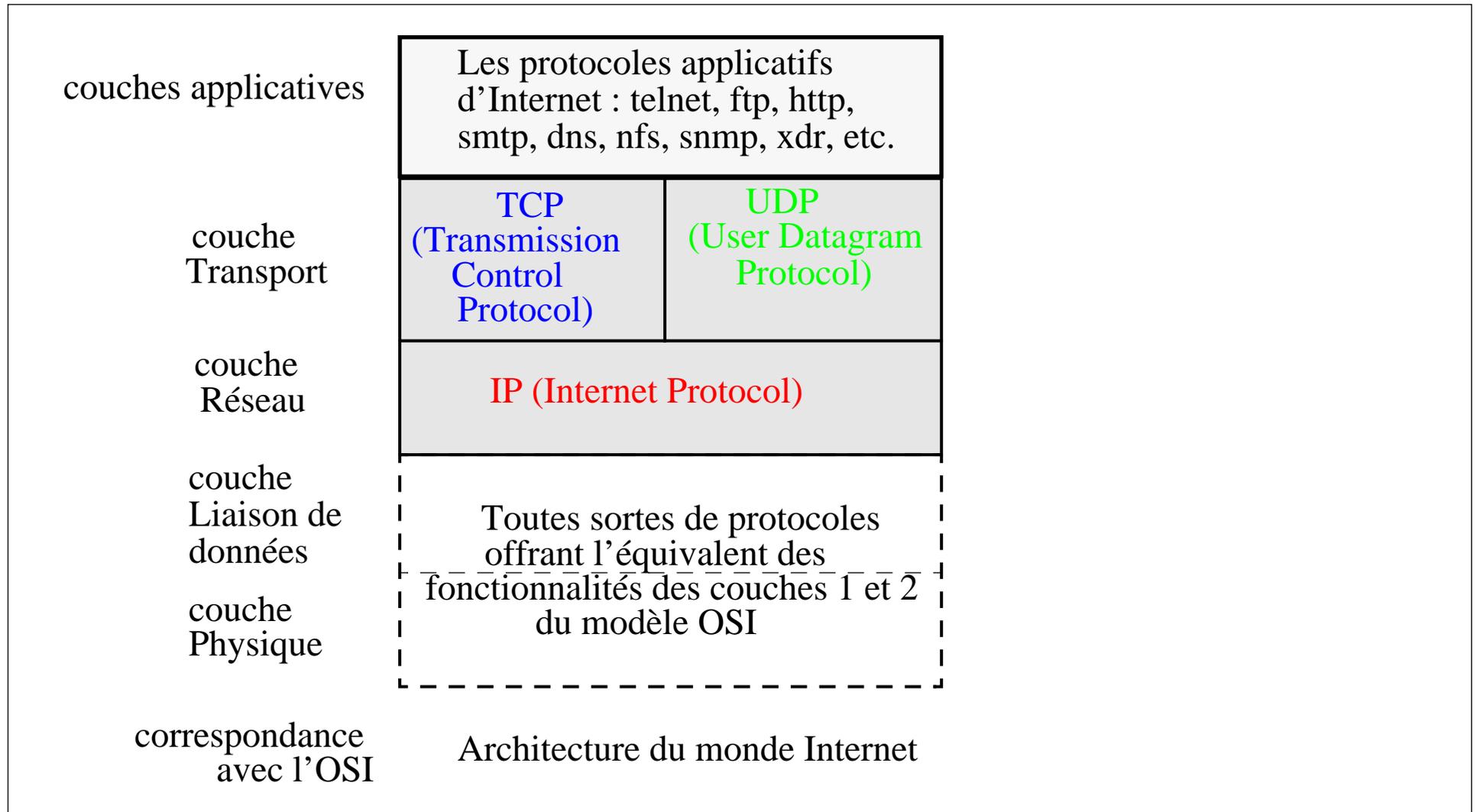


## 4. Autres exemples d'architecture de réseaux

### 4.1. Architecture des réseaux locaux

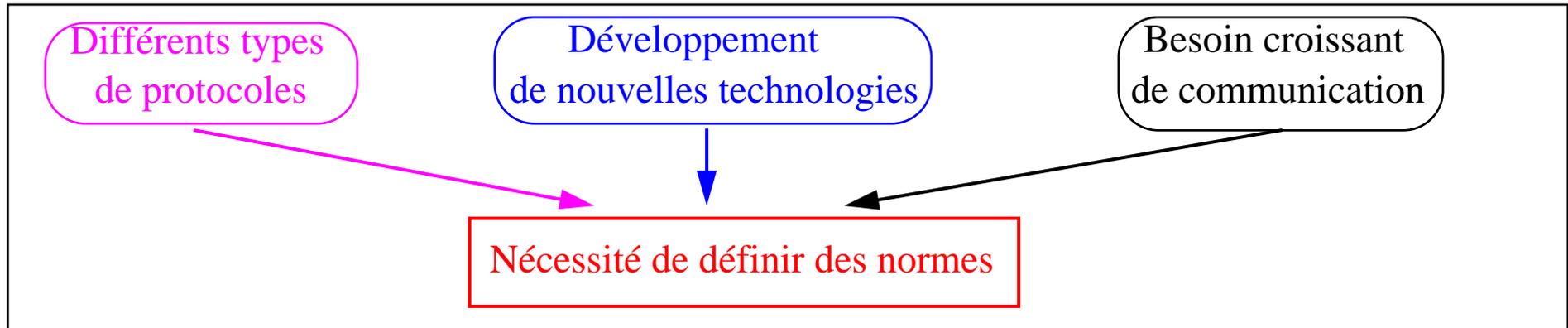


## 4.2. Architecture générale d'Internet



## 5. Normalisation des réseaux

La normalisation garantit l'interfonctionnement et une certaine pérennité, diffusion ou efficacité ou de l'objet produit à partir de ces normes.



### 5.1. Les principaux organismes de normalisation, dans le domaine des réseaux numériques

- **ISO** (International Standardization Organization)
- **IUT-T** (International Union of Telecommunication - section Telecommunication) (ex-CCITT)
- **IEEE** (Institute of Electrical and Electronic Engineers)
- **IETF / IRTF** (Internet Engineering/Research Task Force)
- ANSI, ECMA, AFNOR, etc.

## 5.2. Identification des normes et exemples de normes

La dénomination d'une norme doit tenir compte d'un ensemble de critères :

- son **origine** (ISO, IEEE, etc)
- son **domaine d'application** (réseaux publics/privés/locaux/, téléphone, etc)
- sa **zone d'application** (européenne, internationale, etc)

Exemples de normes :

Les normes ISO sont préfixées par IS (International Standard)

- . **ISO / IS 8208** (=X.25/L3), **ISO / IS 8802.3** (=Ethernet), etc.

Les normes IUT-T sont nommées à l'aide d'une lettre suivie d'un point et d'un numéro :

- . **IUT-T / X.25**, **IUT-T / X.400** (Messagerie), **IUT-T / V.24** (Jonction pour la transmission de données numériques sur lignes téléphoniques), etc.

Les noms des normes IEEE :

- . **IEEE 802.5 (Token Ring)**, etc.

Les normes de l'IETF/IRTF sont appelées des RFC ("Request For Comments") :

- . **RFC 791** (Internet Protocol), **RFC 768** (UDP), **RFC 793** (TCP), ...

### 5.3. Cycle de vie d'une norme

Une norme est généralement développée au sein d'un groupe de travail ad hoc.

- production de "draft"
- expérimentation
- la norme est généralement la (meilleure) réponse technique à un problème en fonction des connaissances de l'époque.

Une norme est adoptée par des instances officielles

- après un vote formel
- la norme est publique et publiée
- la norme répond a des compromis économiques ... et politiques

Une norme est généralement révisée :

- périodiquement
- une norme peut être écartée et même disparaître

## 6. Conclusion

Modèle de référence pour l'interconnexion des systèmes informatiques hétérogènes :

- en 7 couches (P, LdD, N, T, S, P, A)

Mais ce ne doit pas être un dogme immuable, ni la vérité révélée !

Nombreuses variantes ou extensions :

- Piles multi-protocoles (par ex. : UDP/TCP)
- Notion de sous-couches (par ex. : PMD/PHY)
- Adaptation à l'évolution des techniques (réseaux locaux, ATM, RNIS, etc.)
- Interconnexion de réseaux issus de familles protocolaires différentes

Le modèle est cependant pratique pour avoir un langage commun et fournir un repère.

- par ex. : service /protocole, SDU/PDU, multiplexage/segmentation

Nous allons explorer les différentes couches (de bas en haut) en étudiant leurs protocoles, leurs fonctions, leurs mécanismes et le format des objets qu'elles échangent.

## 7. Quelques commentaires

**Note 1** : les normes ont une durée de vie limitée

- Elles naissent, elles vivent, elles meurent !
- ex : IPv4 -- IPv6

**Note 2** : une norme répond à des besoins spécifiques

- Une réponse à un problème, à un instant !
- ex : HDLC n'est pas parfaitement adapté aux réseaux locaux

**Note 3** : les normes spécifient juste ce qu'il faut

- Ce n'est pas un dossier de conception
- Notamment, les normes protocolaires spécifient le comportement des systèmes ouverts dans leurs échanges avec les autres systèmes, mais pas leur fonctionnement interne et propre.
- ex : un langage est défini par sa syntaxe et ses règles opératoires, mais pas par l'implantation de son compilateur !
- ex : de même un protocole informatique est défini par le format de ses messages, la procédure qui régit ces messages (le protocole) et son service, mais pas par une de ses implantations.

**Note 4** : Un concept est abstrait

- Les concepts introduits pour décrire les systèmes ouverts constituent une abstraction, en dépit de similitude avec les objets du monde réel :
- Parfois une nouvelle terminologie spécifique est introduite qui masque l'objet réel
  - . peut rendre difficile l'appréhension du concept.
  - . ex : N-PDU (“Network layer Protocol Data Unit”) == datagramme ou paquet
- Parfois une terminologie usuelle est utilisée qui fait référence (trop) clairement à l'objet réel
  - . peut rendre difficile l'abstraction ou entraîne une certaine confusion
  - . ex : connexion

**Note 5** : Le modèle OSI ne doit pas être un dogme

- Il n'est pas nécessaire que les systèmes soient implantés suivant la description du modèle de référence
- Le système n'est pas forcément organisé en 7 modules ou tâches assurant chacune les fonctions de l'une des 7 couches !
- Le modèle doit s'adapter au cours du temps aux innovations :
  - . les bons concepts, les bonnes techniques perdurent naturellement

Note 6:

- Une couche peut exister et être vide car une couche peut regrouper un ensemble de fonctions qui ne sont pas mises en oeuvre !
- Dans ce dernier cas, soit le service n'est pas nécessaire à la couche supérieure, soit il est déjà rendu par la couche inférieure.

Note 7 :

- Le domaine de normalisation de l'OSI est restreint aux problèmes soulevés par la communication des données entre applications informatiques distantes.
  - . Ce n'est pas le domaine des traitements locaux spécifiques à ces applications tel que les techniques d'utilisation des ressources locales : gestion locale de fichiers, partage du processeur, synchronisation locale des tâches, accès aux données locales, etc.