

Réseaux : RCA - Code RSX103

Comment récupérer le cours au format électronique ?

<http://pierre.sweid1.free.fr/cnam>

Agenda

1. **Introduction** : Rappels Généraux; Concepts de base
2. **Technologies des réseaux locaux** (RLE, 100 Mbps, Giga, ..etc.)/ Câblage
3. **Topologie des équipement d 'interconnexion** et commutation
4. **Routage dans l'Internet**
5. **Réseaux Locaux Sans Fil (WLAN)** : (↔ IEEE 802.11 et ses extensions)
6. **Contrôle de trafic et Qualité de Service** dans les réseaux IP
7. **Services Normalisés (DiffServ & IntServ)**
8. **MPLS** et commutation IP
9. **Routage et transport multimédia (RTP, RTCP)**
10. **Introduction à la Téléphonie sur Internet (VoIP)**
11. **Évolution des technologies de transmission (SDH/SONET, ADSL, ...etc.)**
12. **Réseaux publics** : Frame Relay
13. **Technologie des réseaux publics ATM**
14. **Introduction à l 'administration des réseaux**

Quelques références

1. **Réseaux** : cours et exercices , troisième édition, A. Tanenbaum, Édition DUNOD
2. **Les Réseaux** : Édition 2003, Guy Pujolle, Édition Eyrolles
3. **Architecture des réseaux haut débit**, K-L Thai, V. Vèque, S. Znatt, Edition Hermes
4. **Pratique des réseaux d 'entreprise**, J.L. Montagnier, Edition Eyrolles
5. **Télécoms 1 et 2: de la transmission à l 'architecture des réseaux** , C. Servin , Édition Dunod.
6. **Réseaux locaux et Internet** , des protocoles à l'interconnexion, Laurent Toutain, Édition Hermes
7. **Internet Routing Architecture**, The definitive BGP guide, Sam Halbi, Cisco Presse
8. **Quality of service in IP networks**, Genville Armitage. Macmillan Technical Publications
9. **802.11** et les réseaux sans fil. Paul Muhlethaler. Edition Eyrolles

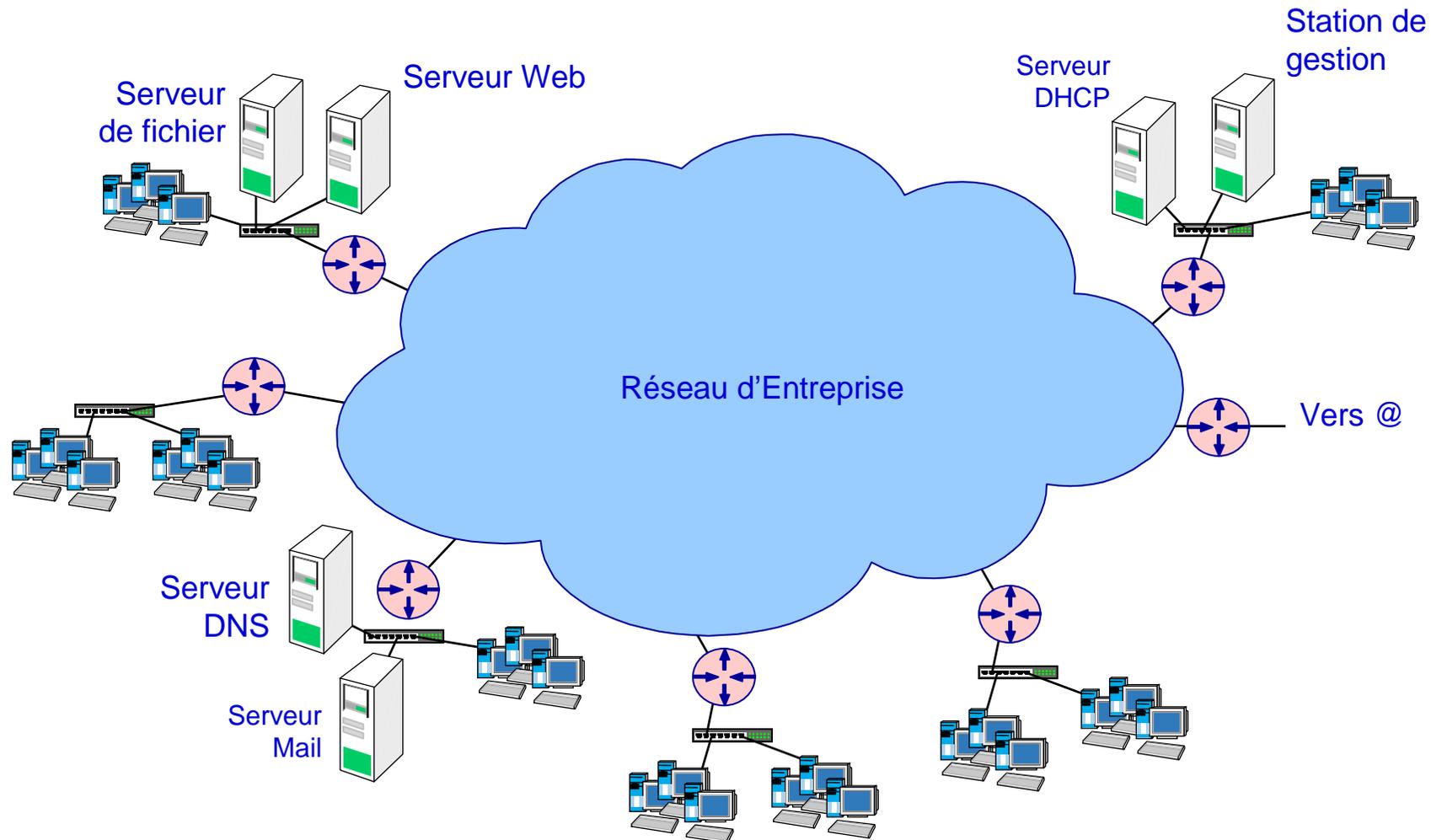
Contenu du chapitre

- Rappels
- L'internet aujourd'hui
- Réseaux d'extrémité
- Le cœur du réseau
- Structure d'Internet et ISPs
- Pertes et délais des paquets
- Structuration des protocoles en couches : modèles et services
- Rappels : OSI /TCP-IP
- Exercice : commutation

Rappels

Qu'est ce qu'un réseau ?

- ◆ Un réseau est constitué d'ordinateurs reliés entre eux à l'aide des **moyens** de communications actuels.



1. Montée en puissance de l'informatisation
 - Tout est informatisé
 - Majorité des applications est du type **client/serveur**
 - Besoins local **et** distant
2. Évolution des technologies de télécommunications
 - Câblage, FO, ...etc.
 - Traitement numérique du signal
 - Nouveaux protocoles
3. Nouveaux besoins
 - Internet
 - Multimédia...
 - etc.

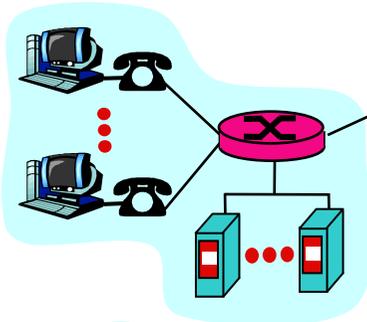
Les nouveaux besoins

1. **Intégration de Service** : Données + Voix + Vidéo
2. **Nouvelles Contraintes** :
 - ⇒ **Différents Débits** : 64 kb/s RNIS-SE, 10 à 100 Mb/s pour les données / 1,5 à 25 Mb/s pour l'image
 - ⇒ **Qualité de Service (QoS)** : Sensibilité aux pertes, Sensibilité aux Délais
 - ⇒ **Trafic** : Continu/Sporadique (Burst)
 - ⇒ **Mode d'exploitation** : Point à point, Multipoint
3. **Nouvelles Interfaces** : Négocier la QoS avec le réseaux

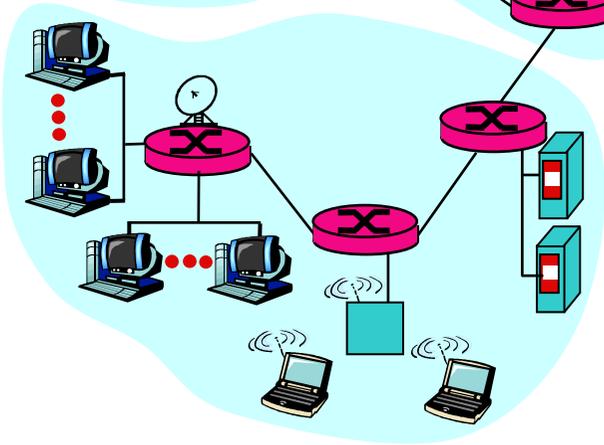
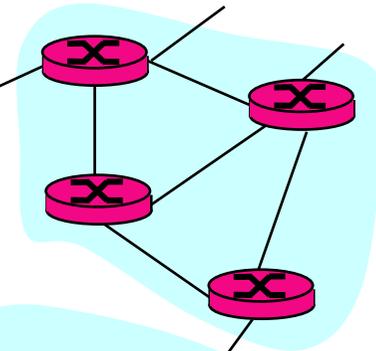
	Données	Voix	Vidéo
Pointes de largeur de bande	élevées	Basses	élevées
Durée de connexions	Courtes	Moyennes	Élevées
Régularité du trafic	Sporadiques	Continu	Continue
Contrôle d'erreur	Essentiel	Faible	Sommaire
Contraintes de synchronisation	Faibles	Importantes	Importantes++

Internet Aujourd'hui

ISP local



ISP regional

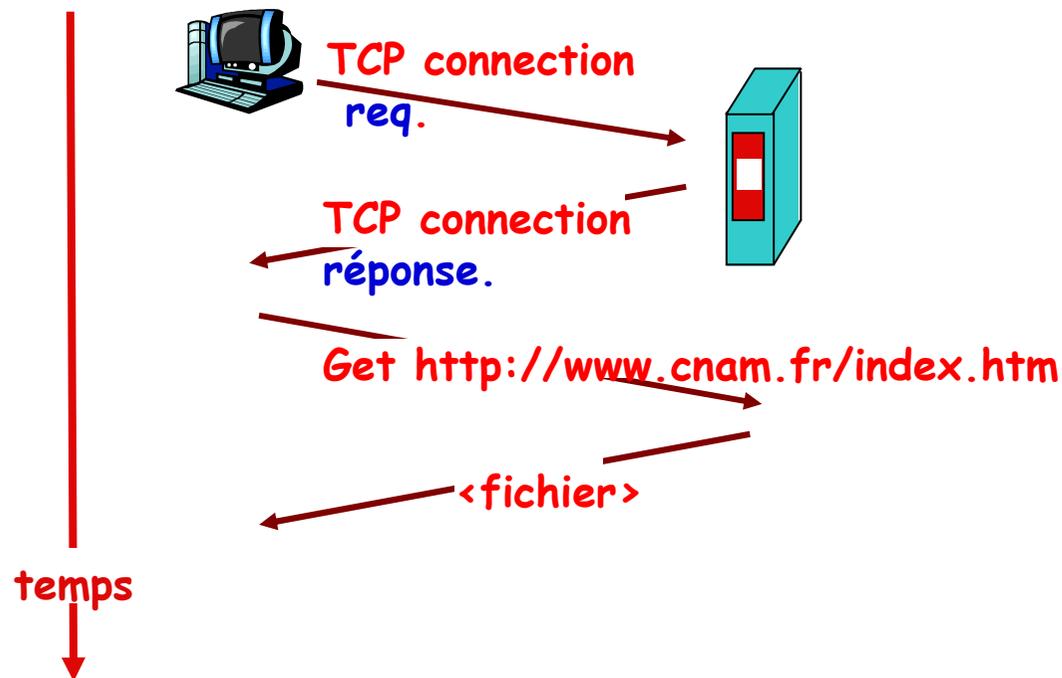


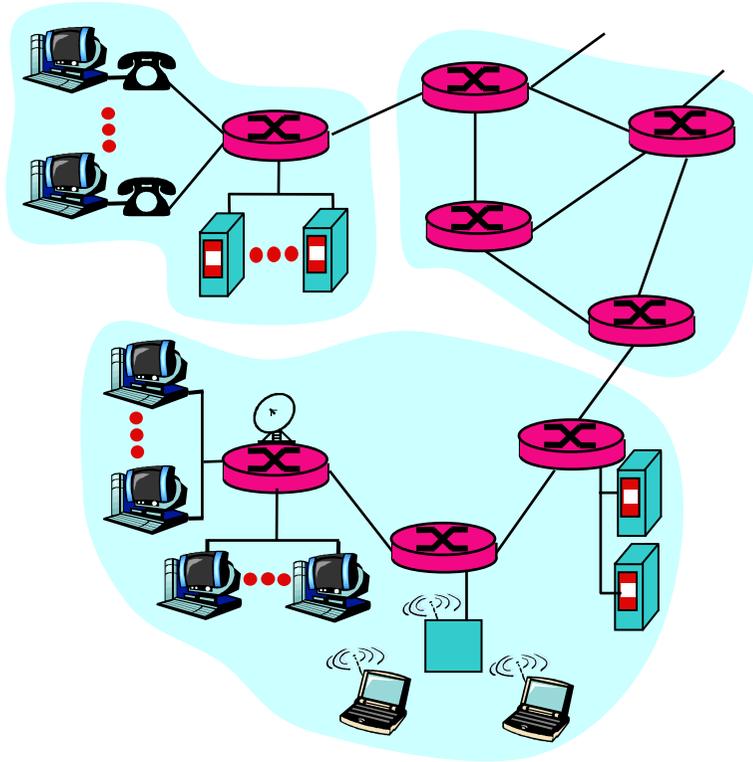
Réseau d'entreprise

- ◆ Des millions de **ressources** interconnectés ensembles : hôtes et des systèmes d'extrémité
- ◆ Des **hôtes** qui exécutent des applications distribuées
 - Clients et serveurs
 - Groupe d'hôtes "en mode peer"
- ◆ L'**infrastructure réseau** qui interconnecte les hôtes :
 - **Des nœuds réseaux** : routeurs, switches, ...etc.
 - **Liens de communications**: câbles, connecteurs, fibre, ...etc.

- Applications distribuées :
 - Exécutées sur les systèmes d'extrémités
 - Échanger des informations les unes avec les autres
 - Exemple : Remote login, transfert de fichiers, e-mail, P2P, ...etc.
 - Groupe d'hôte "en mode peer"
- Deux types de services fournis aux applications :
 - ☛ **Services fiables**, orientés connexion : garantie de délivrance, dans l'ordre et intégrité
 - ☛ **Services Non fiables**, sans connexions : pas de garanties
- Service de base : Best Effort
 - QoS pour fournir d'autres types de garanties (\$\$ chapitre QoS)

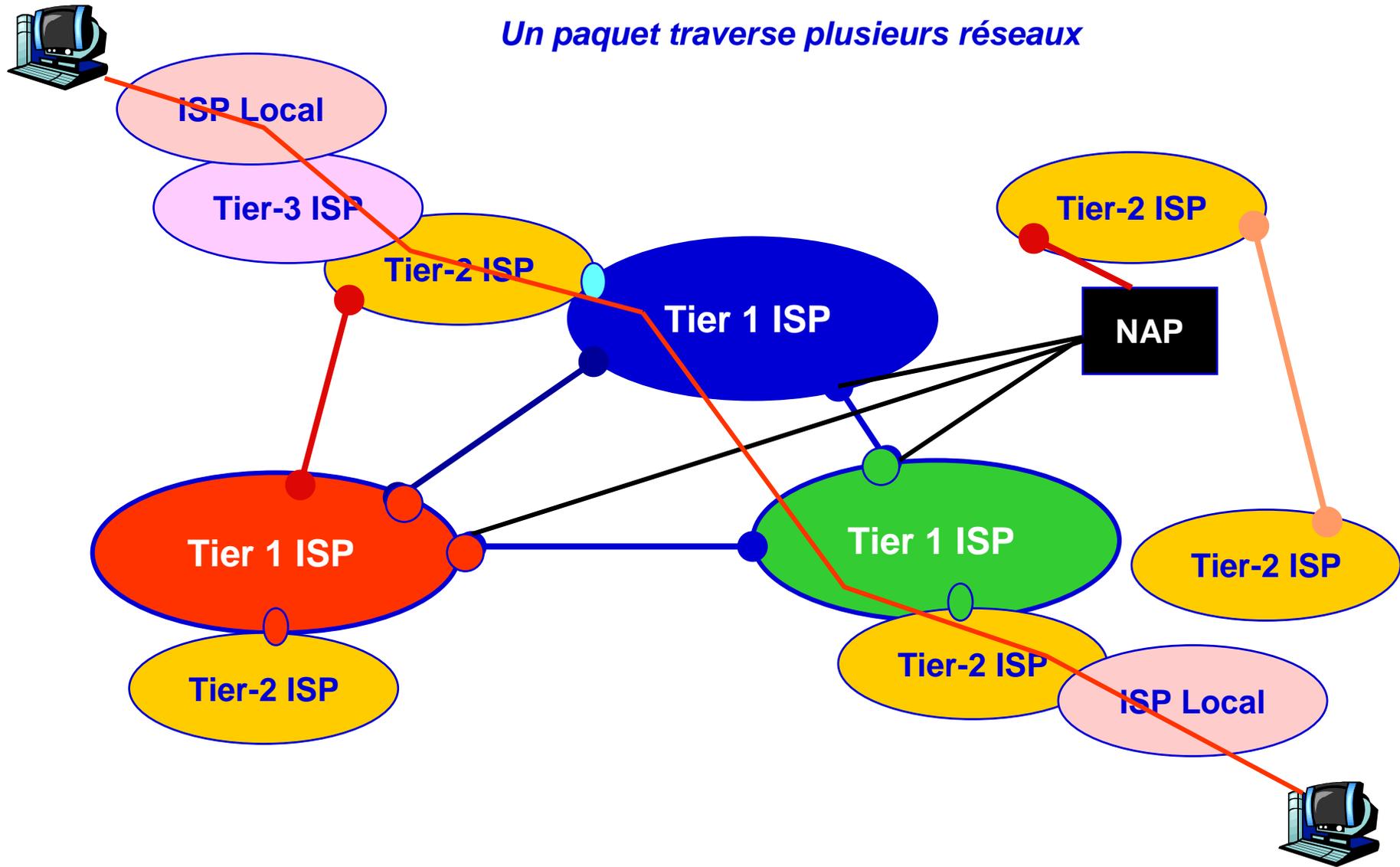
- Toutes les activités de communication dans l'Internet sont gouvernés par **des protocoles** :
- Un protocole définit :
 - Le format, l'ordre des messages échangés entre les entités du réseau
 - Les actions entreprises sur transmission et (ou) réception de message



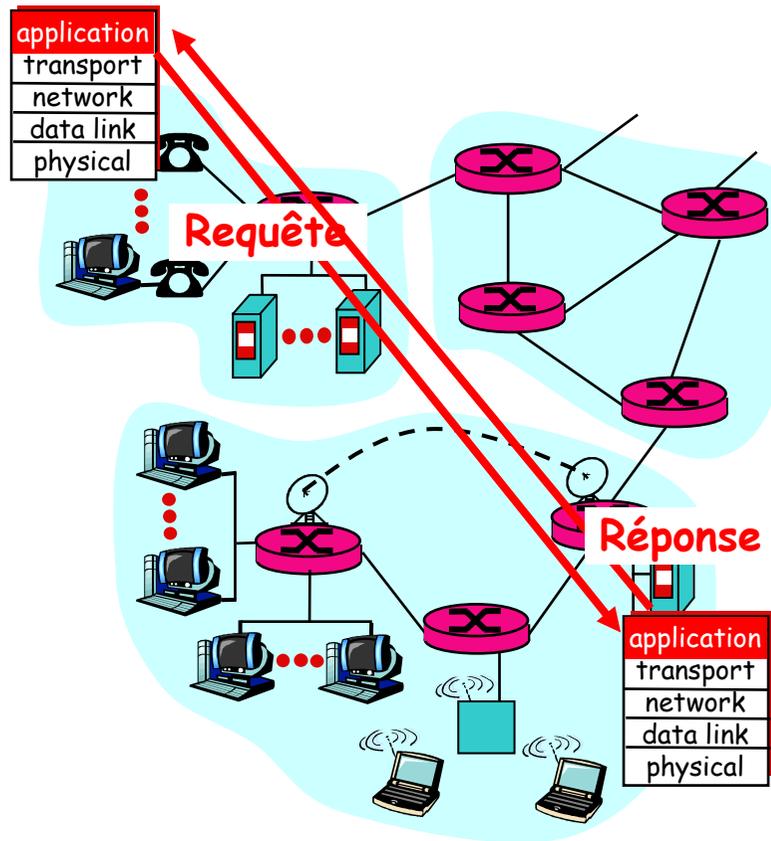


- **Équipements d'extrémités :**
 - Tous ces ordinateurs qui exécutent des applications distribuées
- **Le Coeur du réseaux**
 - Routeurs et des réseaux d'interconnexion
- **Réseaux d'accès**
 - Liens de communications (modems, ...etc.)

Un paquet traverse plusieurs réseaux



Internet Service Provider



- **Systèmes de bordure de réseau [Network Edge, End Systems] (hôtes) :**
 - Exécutent des processus applicatifs
 - Comme Web, E-mail
 - Aux frontières de réseaux
 - Échangent des messages pour implémenter ces applications
- **Deux modèles de communication**
 - Client Serveur
 - » Le client demande, reçoit un service à partir d'un serveur (supposé fonctionner en permanence)
 - Peer-To-Peer (P2P)
 - » Fonctionnement si disponibilité des deux entités impliqués dans l'échange (↔ mode AD-HOC)
 - » Exemple : Gnutella, KaZaA

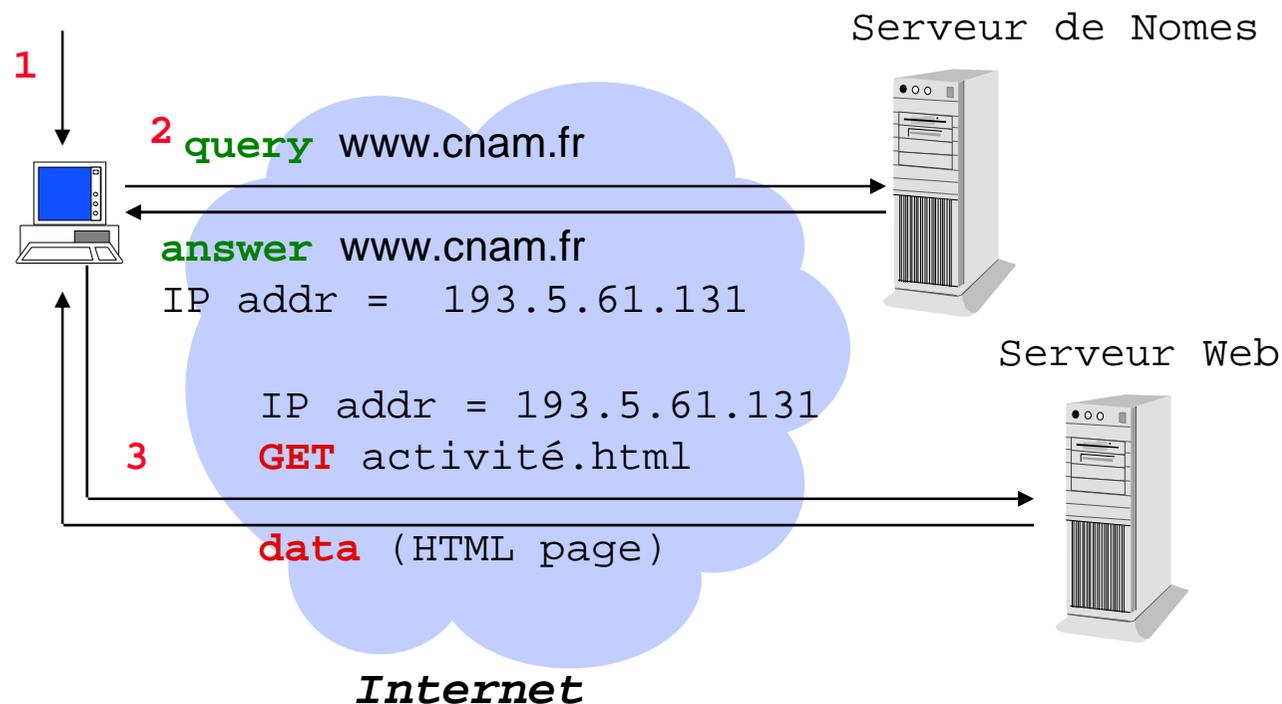
- **Une application Web =**
 - + Explorateur Web (Internet explorer, Opera, ...etc.)
 - + Serveur Web (Apache, Microsoft)
 - + Protocole applicatif (HTTP)
 - + Standard de format doc (HTML)

Exemple du modèle Client-Serveur

- Applications distribuées qui utilisent le modèle client-serveur
- **serveur** = un programme qui attend des données (requêtes) soient envoyées à lui
 - Interprète la requête et envoie une réponse
- **clients** : Envoie des données (Requêtes) aux serveurs
 - Attend pour la réponse

Utilisateur : exécute:

```
http://www.cnam.fr/activités.html
```



- **Objectif : (aussi bien pour les services orientés connexion que sans connexion)**
 - transférer les informations entre les hôtes
- 1. Services orienté connexion :
 - **Fonction de handshaking :**
 - » **Setup (avant le transfert de donnée) : établissement de connexion “contact” préalable entre les deux entités pour qu’elles se reconnaissent l’une avec l’autre et qu’elles se préparent à l’échange). Ce qu’on appelle l’information d’Etat qui est conservé dans les deux hôtes en communication**
 - **Optionnel :**
 - » **Fiable, en ordre :** transfert de données sous la forme d’un flux d’octets
 - Perte des paquets => Acquiescement et retransmission
 - » **Contrôle de flux :** l’émetteur évite d’écraser le receveur (asservissement du récepteur sur l’émetteur)
 - » **Contrôle de congestion :** l’émetteur diminue sa transmission lorsque le réseau est congestionné
- **TCP (Transmission Control Protocol, RFC 793) :**
 - » **Fournit des services Internet orientés connexion**

2. Les services Sans connexion :

- **Objectif :**
 - » **Transférer les informations entre les hôtes**
- **Pas de handshaking :**
 - » **Les donnée sont délivrées aussitôt**
- **Transfert Non fiable**
- **Pas de contrôle de flux**
- **Pas de contrôle de congestion**
- **UDP (User Data Protocol, RFC 768) :**
 - **Services internet sans connexion**

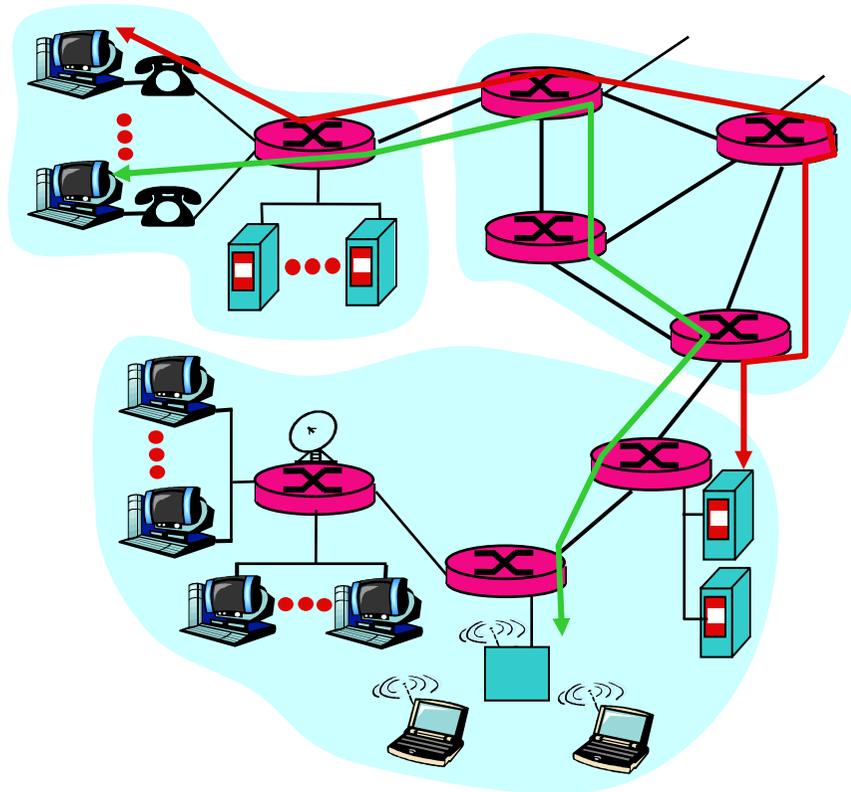
TCP	UDP
HTTP (Web: RFC 2616)	Streaming media
FTP (File Transfer : RFC2959)	Télé-conférences
Telnet (Remote login : RFC 854)	DNS Query
SMTP (e-mail : RFC2821)	Téléphonie sur Internet

Le cœur de L 'Internet

« Commutation Et Multiplexage »

Commutation

- *Commutation* : Processus d'acheminement des données à travers le réseau

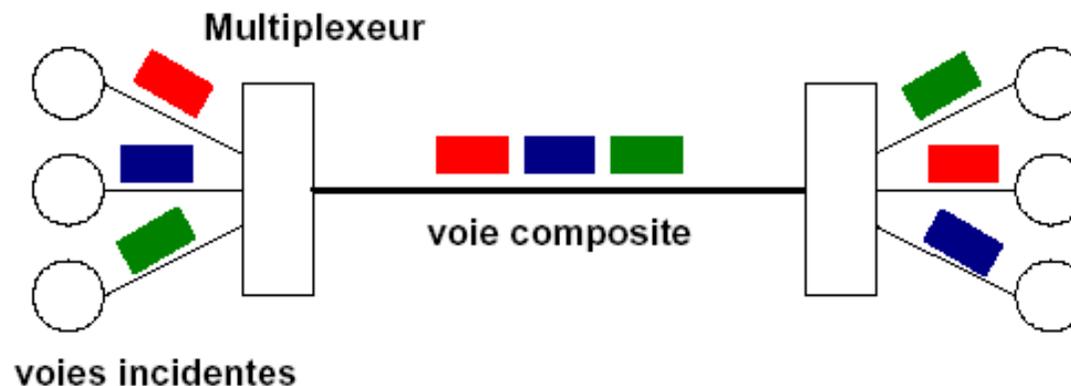


Pourquoi multiplexer et commuter ?

- **Quel est le problème ?**
 - Comment mettre **en relation** deux entités communicantes du réseau ?
 - interconnexion totale de N stations = $N(N-1)/2$ liens physiques
 - Comment répartir la charge du réseau et être résistant aux pannes ?
 - **résistance aux pannes** ⇔ plusieurs chemins pour aller de A à B
 - si **plusieurs chemins** ⇔ lequel choisir ?
- ☛ **Solution : Réseau maillé** ⇔ deux stations clientes du réseau peuvent être mises en relation par différents chemins
- **Minimiser le coût de l'infrastructure du réseau**
 - **Multiplexage** de plusieurs communications sur un même lien physique
 - **Minimisation** du nombre de liens au sein du réseau
- **Optimiser l'infrastructure du réseau**
 - ☛ Le nombre de communications potentielles peut être très **supérieur au nombre** de liens du réseau de commutation
 - ☛ Atteindre le destinataire de la façon la plus efficace
 - ☛ Optimiser l'utilisation des ressources du réseau : si chaque abonné désire joindre un autre abonné, il faut optimiser **le partage des ressources**

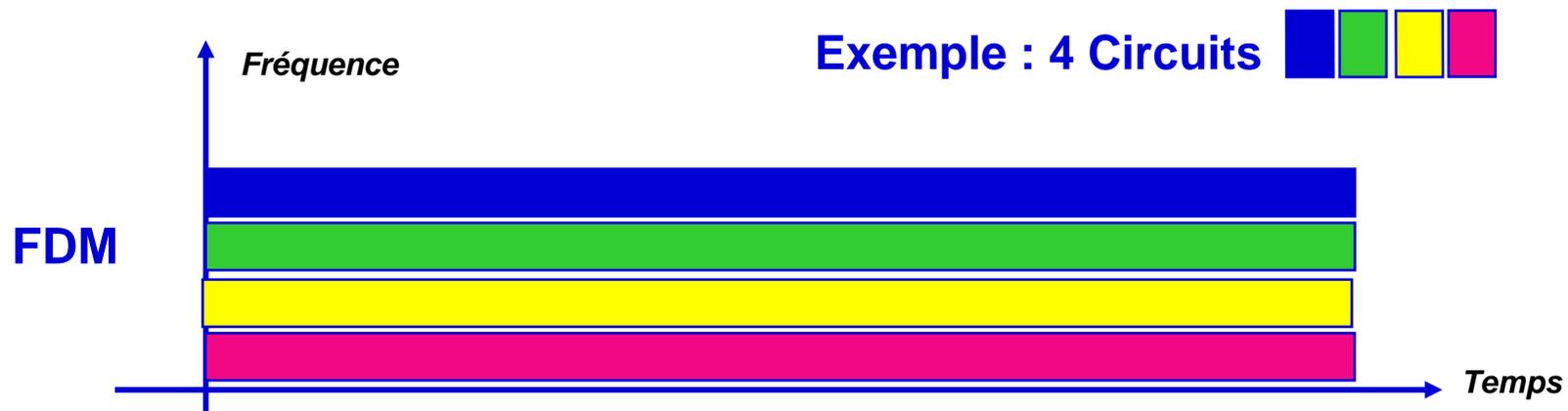
Pourquoi multiplexer et commuter ?

- **Acheminement simultané de plusieurs Communications sur un même lien physique**
 - Multiplexage : regroupement de plusieurs voies incidentes sur un même support
- **Démultiplexage** : restitution à chaque destinataire des données des différentes voies



▪ Multiplexage spatial ou fréquentiel

- Partage de la bande disponible
- Chaque voie dispose **en permanence** d'une **fraction de la bande disponible**
- Le spectre de fréquences est divisé en canaux logiques (sous-bandes), indépendants les uns des autres

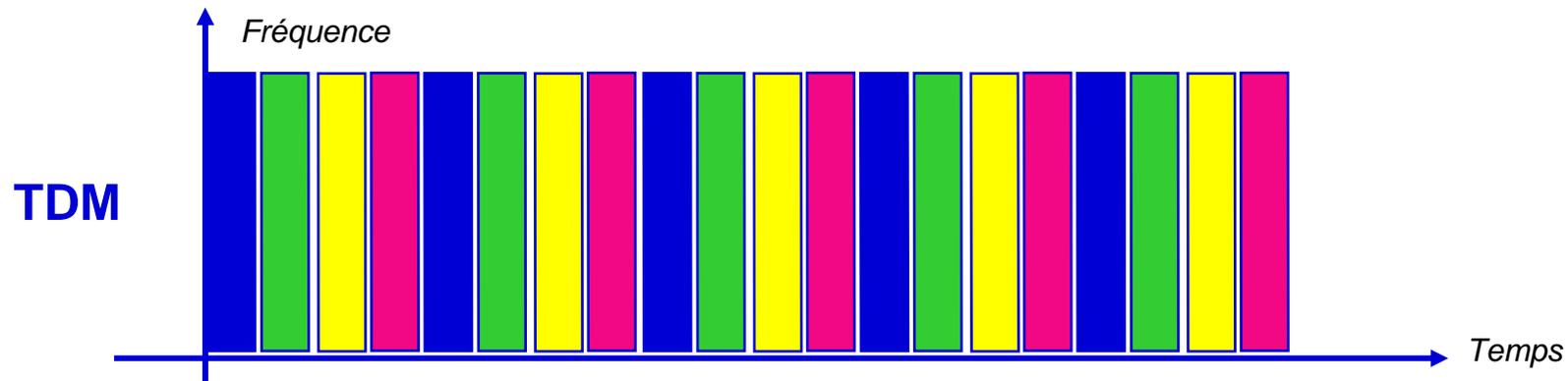


FDM : Frequency Division Multiplexing

■ Multiplexage temporel

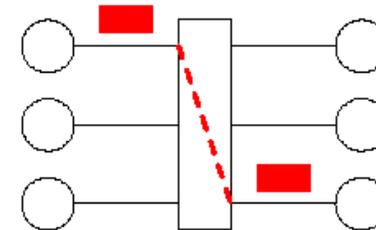
- Partage du temps d'utilisation de la voie
- Chaque voie utilise durant un temps prédéterminé **toute la bande disponible**
- Un usager prend toutes les ressources disponibles pendant un intervalle de temps

Exemple : 4 Circuits



TDM : Time Division Multiplexing

- **Objectif : Aiguillage de la communication d'un canal en entrée vers un canal de sortie**
- **Diverses techniques :**
 - Commutation de circuits
 - Commutation de messages
 - Commutation de paquets
 - Commutation de trames
 - Commutation de cellules
- **Un réseau à commutation assure une connectivité totale du point de vue utilisateurs (différence entre topologie physique et topologie logique)**

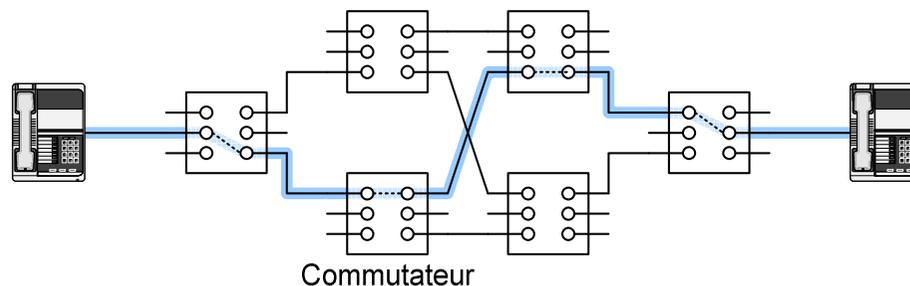


Commutateur

Commutation de circuits

▪ Principe

- Établissement d'un **circuit physique** avant la communication par les commutateurs intermédiaires
- Le circuit est **réservé à une seule communication**
- Transmission d'un **signal continu**



▪ Utilisée dans les réseaux téléphoniques

- Durée de l'établissement est petite par rapport à la conversation
- Qualité et délai de transmission constants

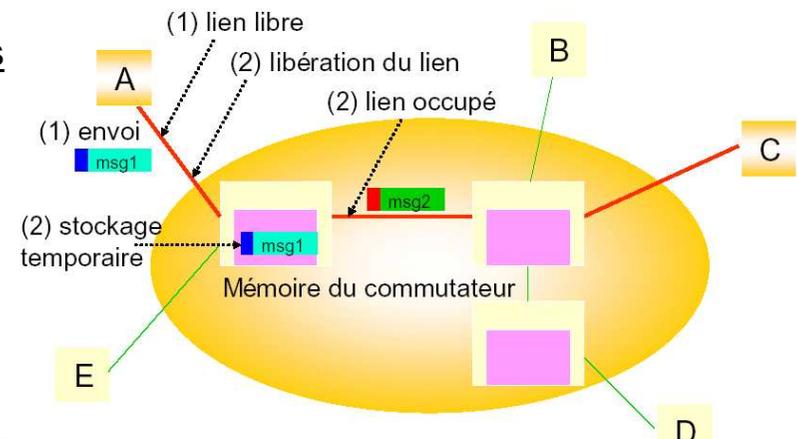
▪ Mal adaptée au trafic téléinformatique

- Trafic téléinformatique ⇔ Courts échanges de données avec de **longues pauses**

Commutation de messages

- **Principe** : pas d'établissement préalable de la communication (**aucun lien physique entre la source et le destinataire**)
- **Message** : un message constitue une unité de transfert qui est acheminée individuellement dans le réseau (p.ex. un E-mail)
- **Sur chaque nœud du réseau, un message est** (mode transmission "*Store and Forward*")
 - reçu en entier
 - stocké si nécessaire (occupation des lignes)
 - analysé (contrôle des erreurs)
 - transmis au nœud suivant, etc...
 - transmis au nœud suivant, etc...
- **Facturation en fonction de la quantité de données**
- **Temps de transit dépend de la taille du message, de la taille du réseau et du trafic**
- **Avantages** :
 - Utilisation économique des lignes de transmission (par rapport à la commutation de circuits)
 - Transfert même si le destinataire est déconnecté
 - Diffusion économique d'un messages à plusieurs destinataires
 - Conversion de codes et formats possible
- **Inconvénient**: **délais trop longs pour un dialogue**

Commutation de messages



▪ Principe :

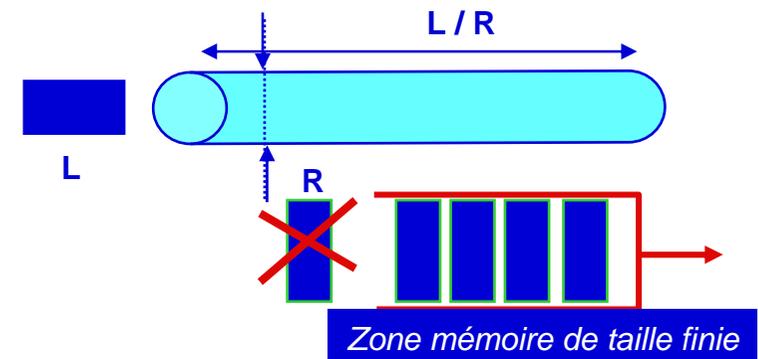
- **Paquet** : petit bloc de données (p.ex. 512 octets d'un eMail)
- La source **segmente le message** à transmettre en paquets et les transmet l'un après l'autre au premier commutateur
- Chaque **paquet** utilise la bande passante totale du lien
- Un commutateur transmet un paquet reçu dès que possible sans attendre les prochains paquets
- **Store and Forward** : on attend l'arrivée complet d'un paquet dans un nœud intermédiaire avant de le retransmettre
- Le destinataire **re-combine les paquets** reçus pour obtenir le message

▪ Avantages

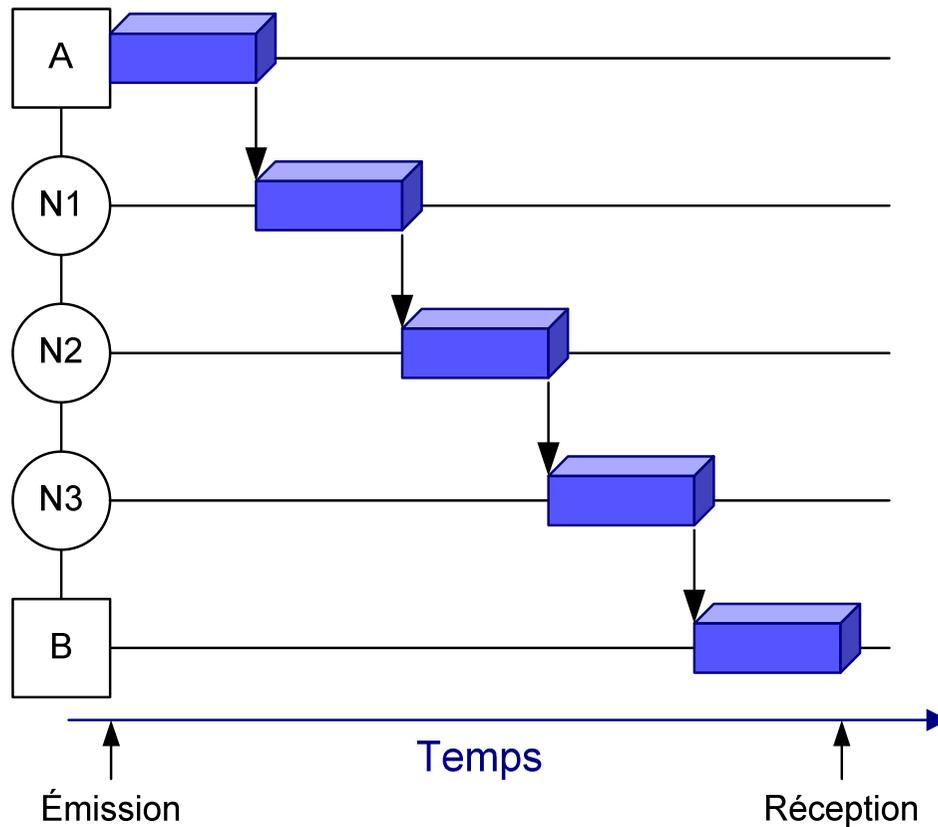
- Utilisation économique de la ligne
- Délai de transfert court
- La petite taille des paquets évite de monopoliser la ligne
- Conversion de codes et de formats possible

▪ Problèmes à prendre en compte

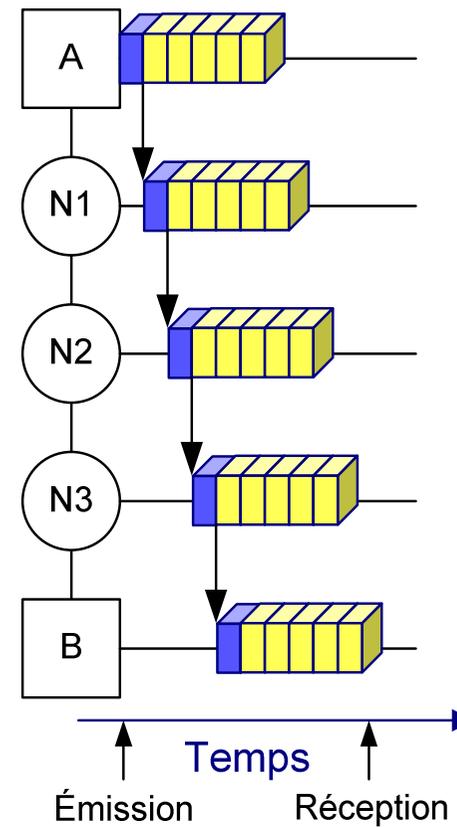
- Délai de transfert variable (Délai = L/R (dépend de la taille du paquet et la capacité de transmission du lien)
- Pertes de paquets possibles (\Leftrightarrow Congestion)
 - » Les paquets sont mis dans une file d'attente,
 - » Attendre la disponibilité du lien
 - » Ceci génère des délais d'attente (délais de files d'attente \Leftrightarrow queuing delays) et même perte de paquets



Commutation de messages vs. Commutation de paquets



a) Commutation de messages



b) Commutation de paquets

Deux modes de commutation de paquets

1. Sans connexion

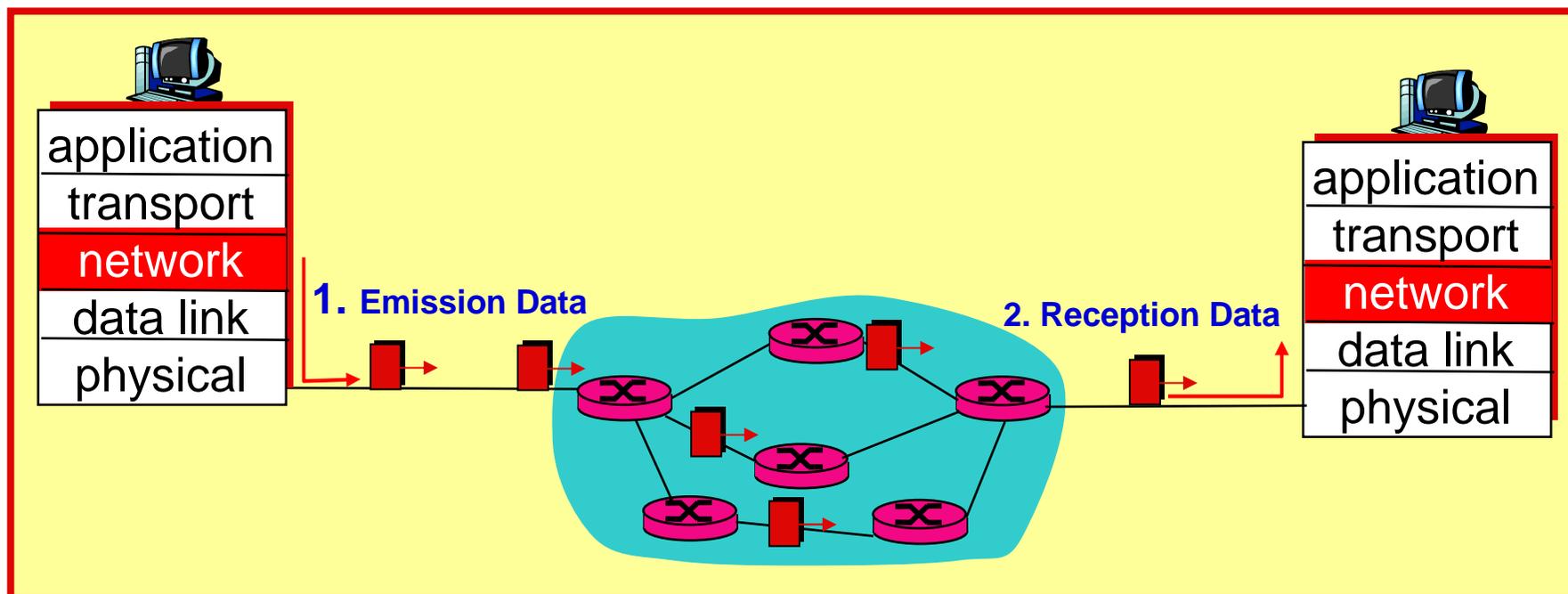
- Transmission de « datagrammes » sans établissement préalable d'une connexion
- Tous les datagrammes contiennent l'adresse du destinataire et sont acheminés indépendamment l'un de l'autre (l'adresse de destination détermine le prochain saut (Next Hop))
- Exemples : brefs échanges de messages, transmissions multicast

2. Orienté connexion

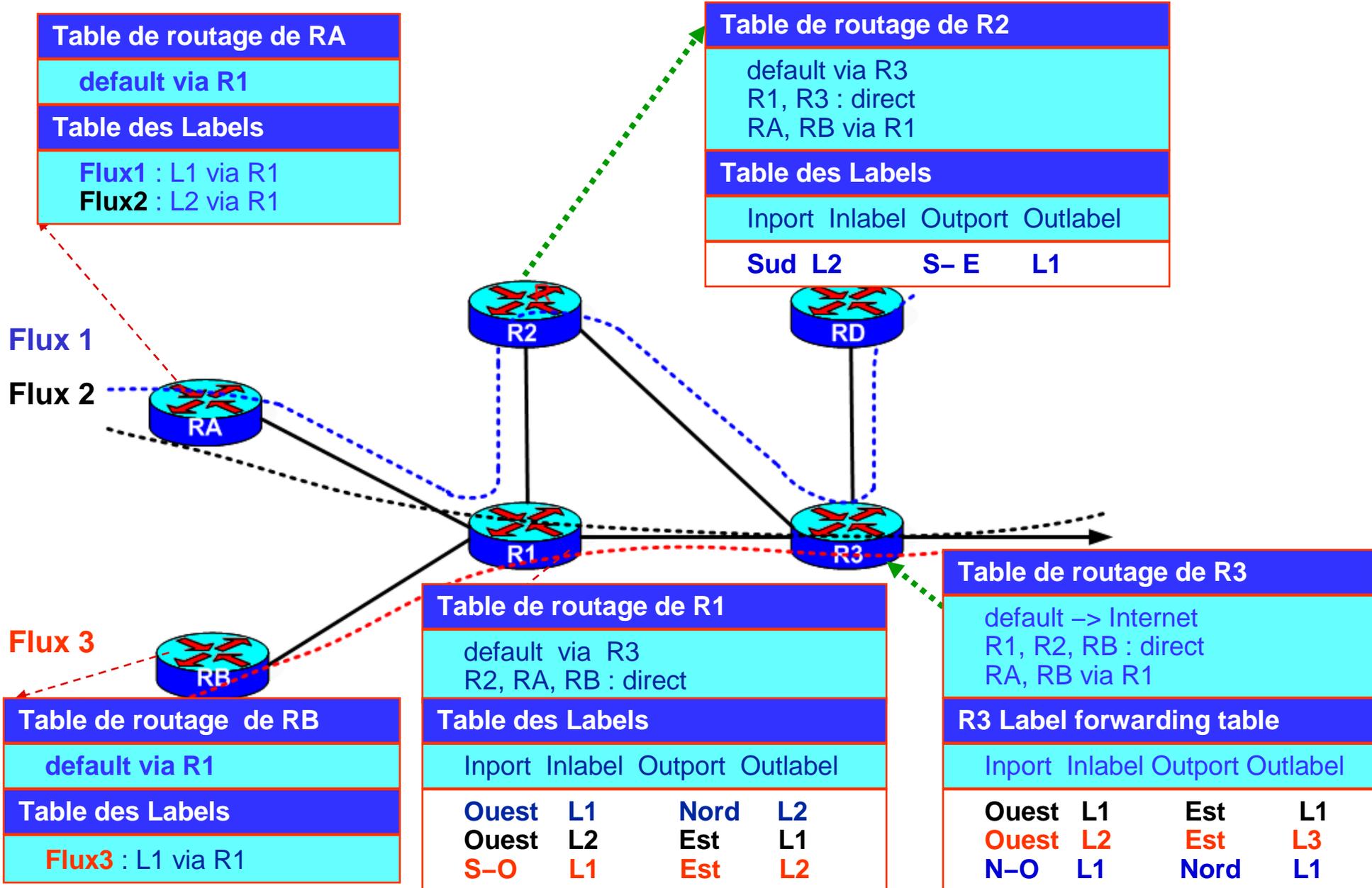
- Une liaison logique (connexion) est établie avant la transmission
- Négociation des paramètres de la transmission
- Les deux partenaires mémorisent des informations sur la connexion
- Permet de vérifier l'ordre correct des paquets, ...
- Exemple : transfert d'un fichier long
- Extension: Circuits virtuels
 - Choix d'un chemin fixe lors de l'établissement du circuit virtuel
 - Un *identificateur* du circuit virtuel est ajouté aux paquets

Réseaux Datagramme : Le modèle de l'Internet

- Pas de phase d'établissement d'appel a niveau de la couche réseau
- Au niveau des routeurs : pas d'information d'état sur les connexions : bout à bout (E2E)
 - Pas de concept de connexion au niveau de la couche réseau
- Les paquets sont routés sur la base de l'adresse de destination
 - Paquets entre une paire source-destination peuvent emprunter des chemins différents

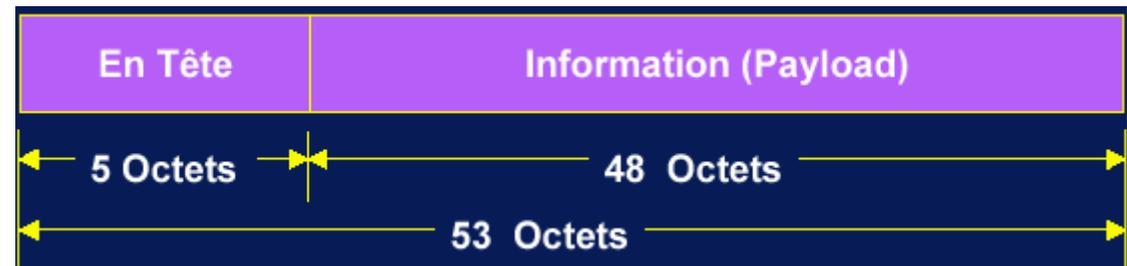


Réseaux à commutation de circuit virtuel « Mode MPLS dans un réseau IP »



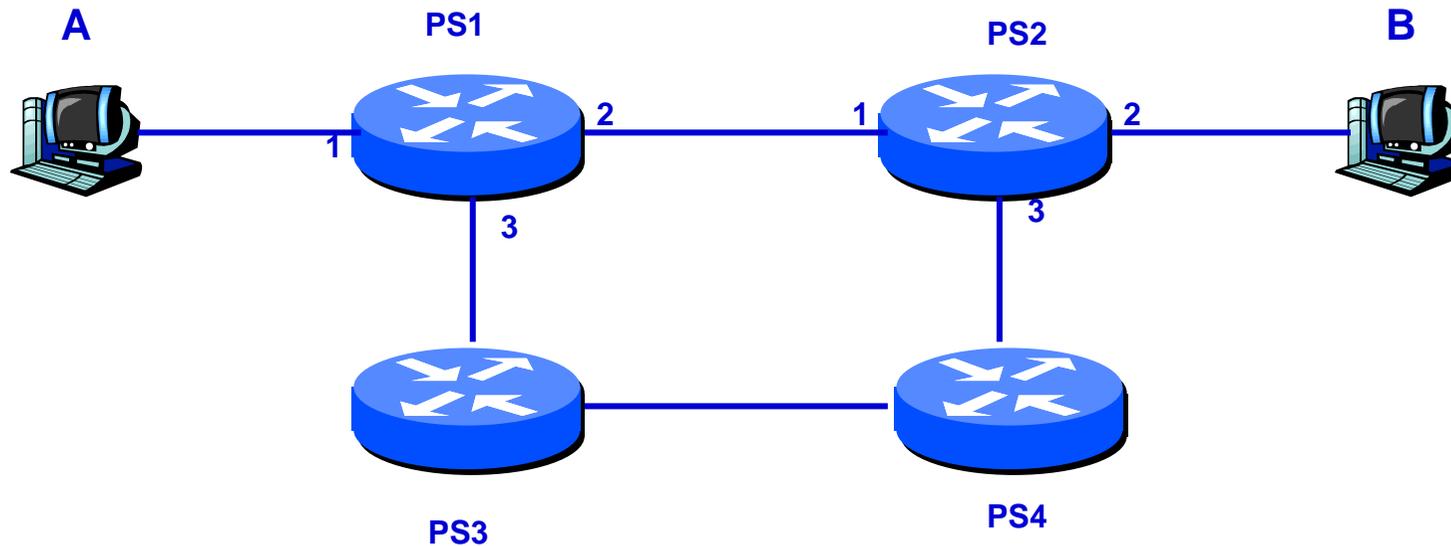
La commutation de cellules

- Commutation de paquets mais taille des paquets fixe : 53 octets - on parle de cellule
 - 5 octets d'en-tête + 48 octets de données
 - Commutation plus simple et plus rapide
 - Meilleure gestion de la mémoire et des files d'attente



- Utilisée dans les réseaux ATM : transmission de cellules sur un circuit virtuel (VCI+VPI)
 - VCI (Virtual Channel Identifier) : numéro de voie virtuelle
 - VPI (Virtual Path Identifier) : numéro de chemin virtuel
 - un VP peut contenir plusieurs VC (multiplexage)
 - un VP est en fait une connexion multipoint
- Les VCI et VPI sont contenus dans l'en-tête ATM de chaque cellule (données d'acheminement)

Réseaux à commutation de circuit virtuel « Mode ATM »

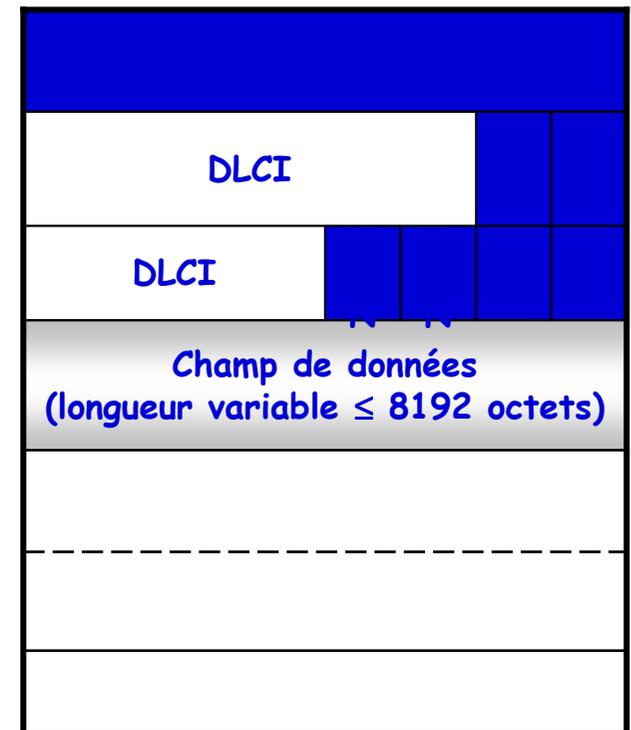


PS1			
In		OUT	
Port	VC	Port	VC
1	12	2	22
1	25	3	31
1	21	2	24

PS2			
In		OUT	
Port	VC	Port	VC
1	22	2	32
2	33	1	21
3	15	1	33

Chemin A -> PS1 -> PS2 -> B = {VC12, VC22, VC32}

- Améliorer la performance de la commutation de paquets en reportant la commutation au niveau de la liaison (niveau trame)
- Commutation en fonction **de l'adresse de la trame** (l'adresse de niveau 3 n'est plus examinée)
- Le commutateur assure les fonctionnalités suivantes
 - délimitation et transparence des trames
 - inspection de la longueur de la trame
 - détection des erreurs et demande de retransmission à l'émetteur
 - contrôle de flux entre commutateurs



Structure d'une trame FR

Réseaux à commutation de circuit virtuel « Mode Frame Relay »

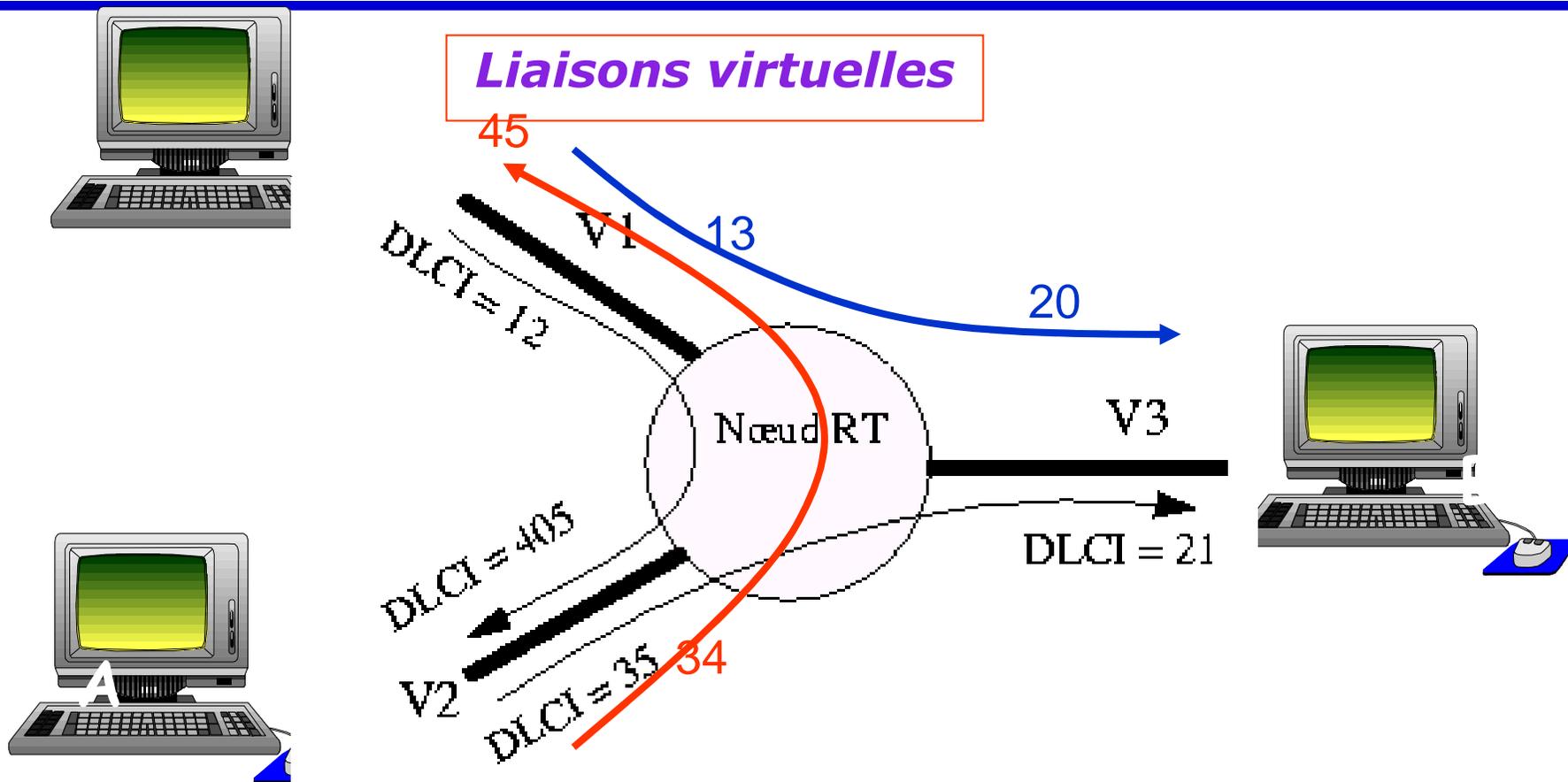


Table de commutation V1

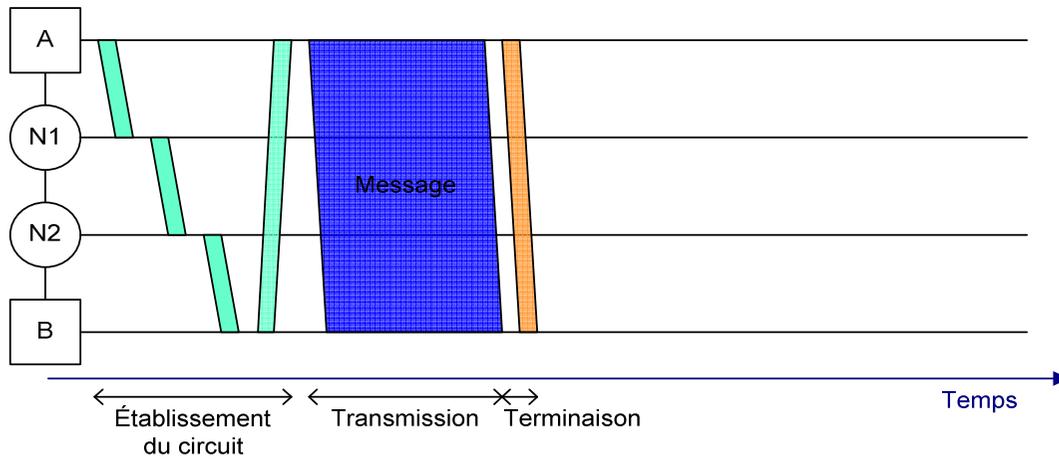
DLCI entrant	Voie Sortie	DLCI sortant
12	V2	405
13	V3	20

Table de commutation V2

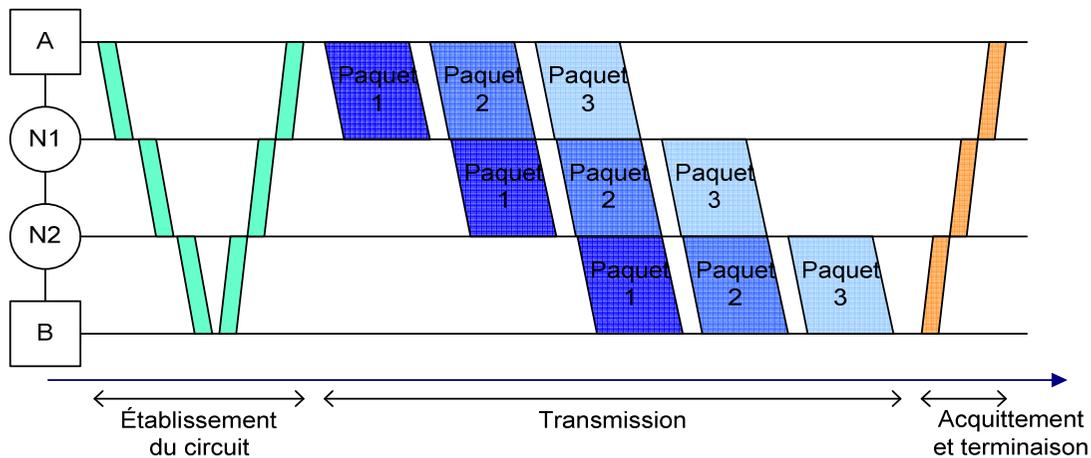
DLCI entrant	Voie Sortie	DLCI sortant
34	V1	45
35	V3	21

Commutation : Bilan

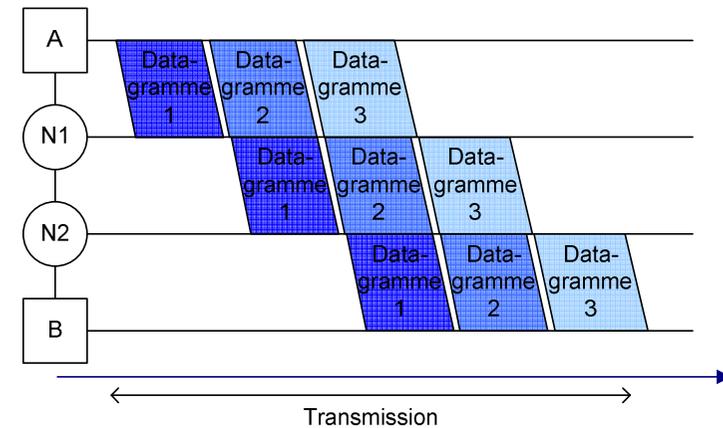
Comparaison des délais



a) Commutation de circuits



b) Commutation par paquets, orientée connexion



c) Commutation par paquets, sans connexion

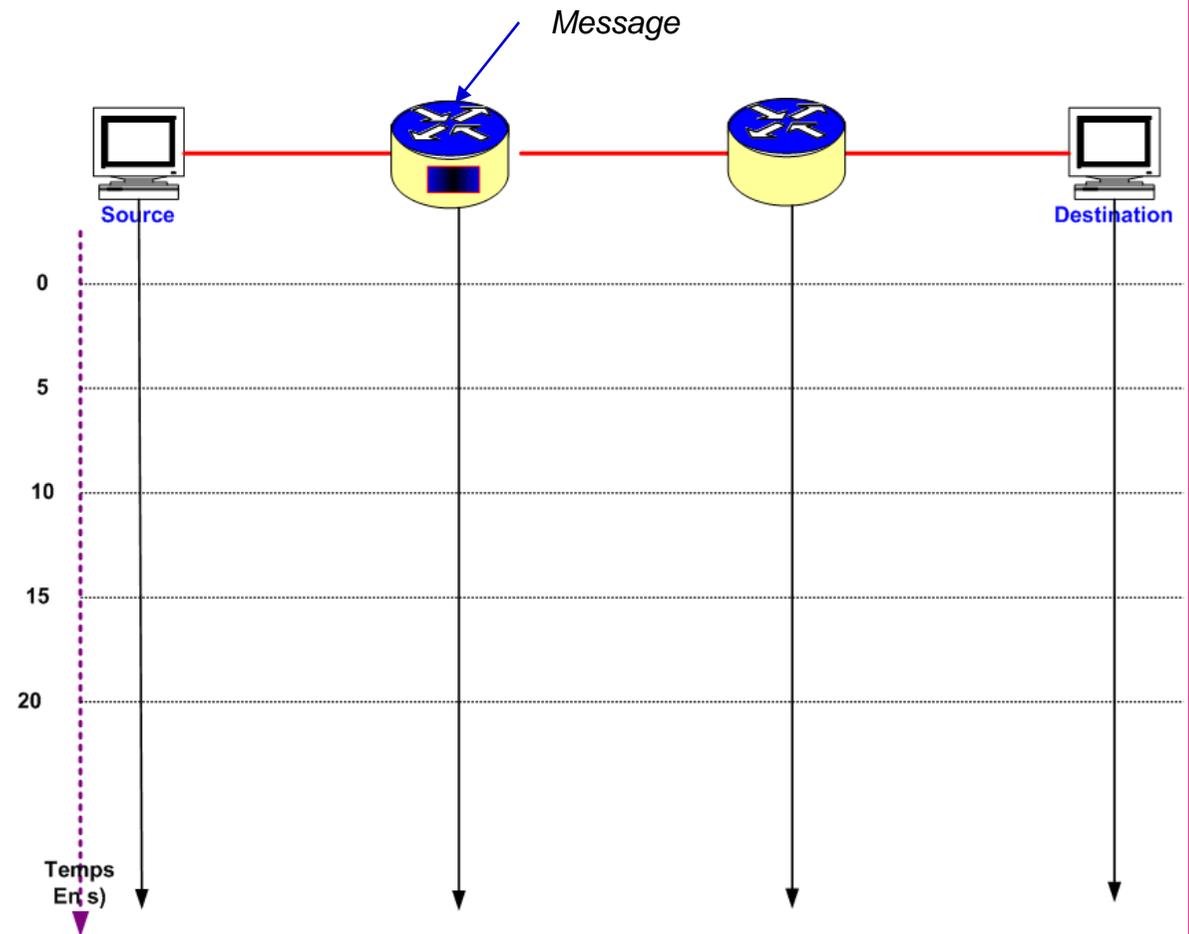
Comparaison des méthodes de commutation

Commutation de circuits	Commutation par paquets		
	Circuit virtuel	Orientée connexion	Sans connexion
Transmission de bits	Transmission de paquets	Transmission de paquets	Transmission de paquets
Établissement d'un circuit (physique) avant la transmission	Établissement d'un circuit virtuel avant la transmission	Établissement d'une connexion avant la transmission	Transmission à tout moment possible
Chemin de transmission fixe	Chemin de transmission fixe	Chemin de transmission variable	Routage indépendant pour chaque datagramme
Données sans en-têtes	Identificateur du circuit virtuel dans chaque paquet	Adresse destinataire et identificateur de connexion dans chaque paquet	Adresse destinataire dans chaque datagramme
Réseaux téléphoniques	Réseaux d'ordinateurs à haut débit	Transfert de fichiers sur Internet	Courtes interactions et trafic multimédia

Tester votre compréhension

Commutation de messages

- Message de 7.5×10^6 bits
- Deux Commutateurs
- Trois liens de 1.5 Mbits/s
- Transmission en Store and Forward



[solution](#)

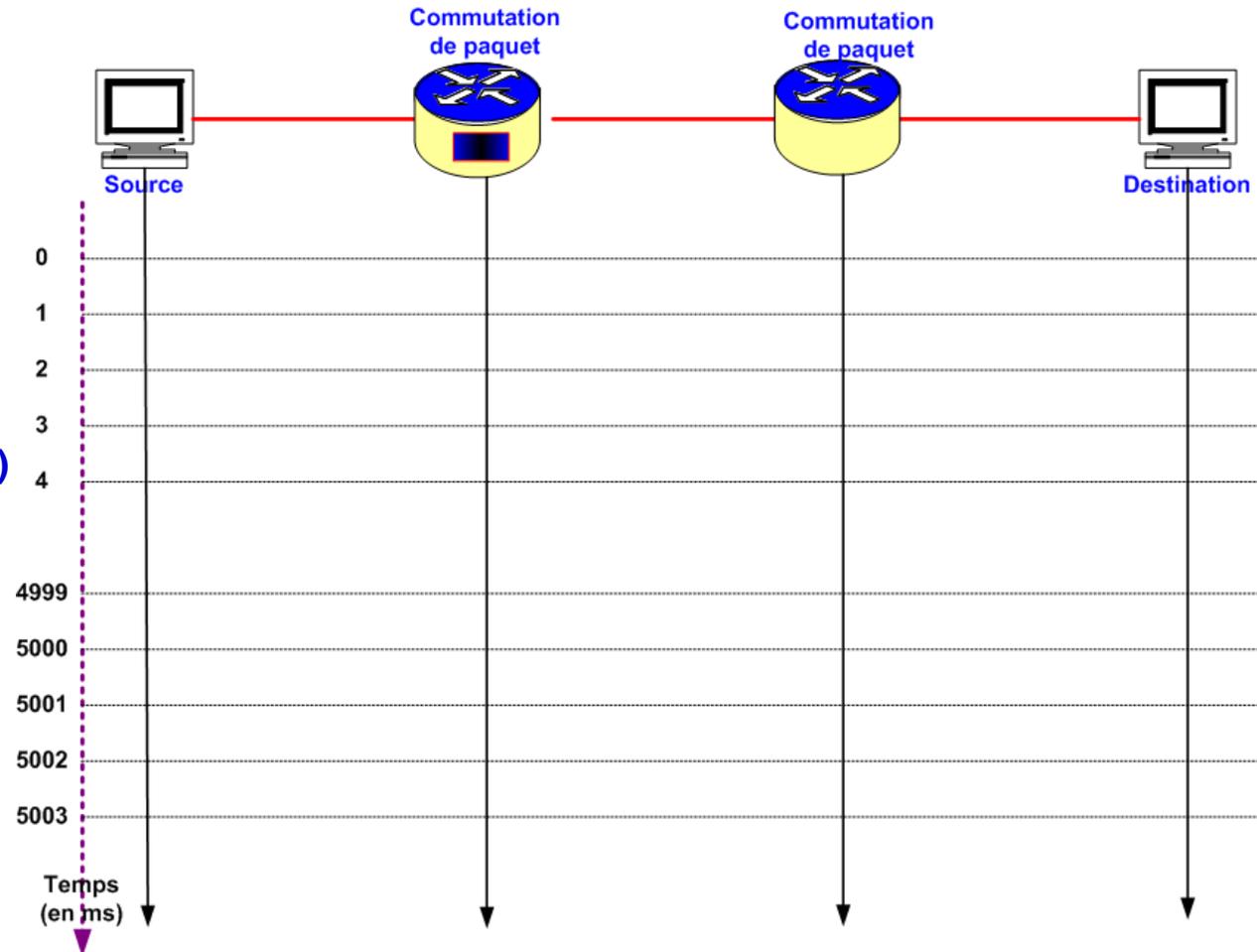
Voir aussi :

http://media.pearsoncmg.com/aw/aw_kurose_network_2/applets/message/messagesegmentation.html

Commutation de paquets

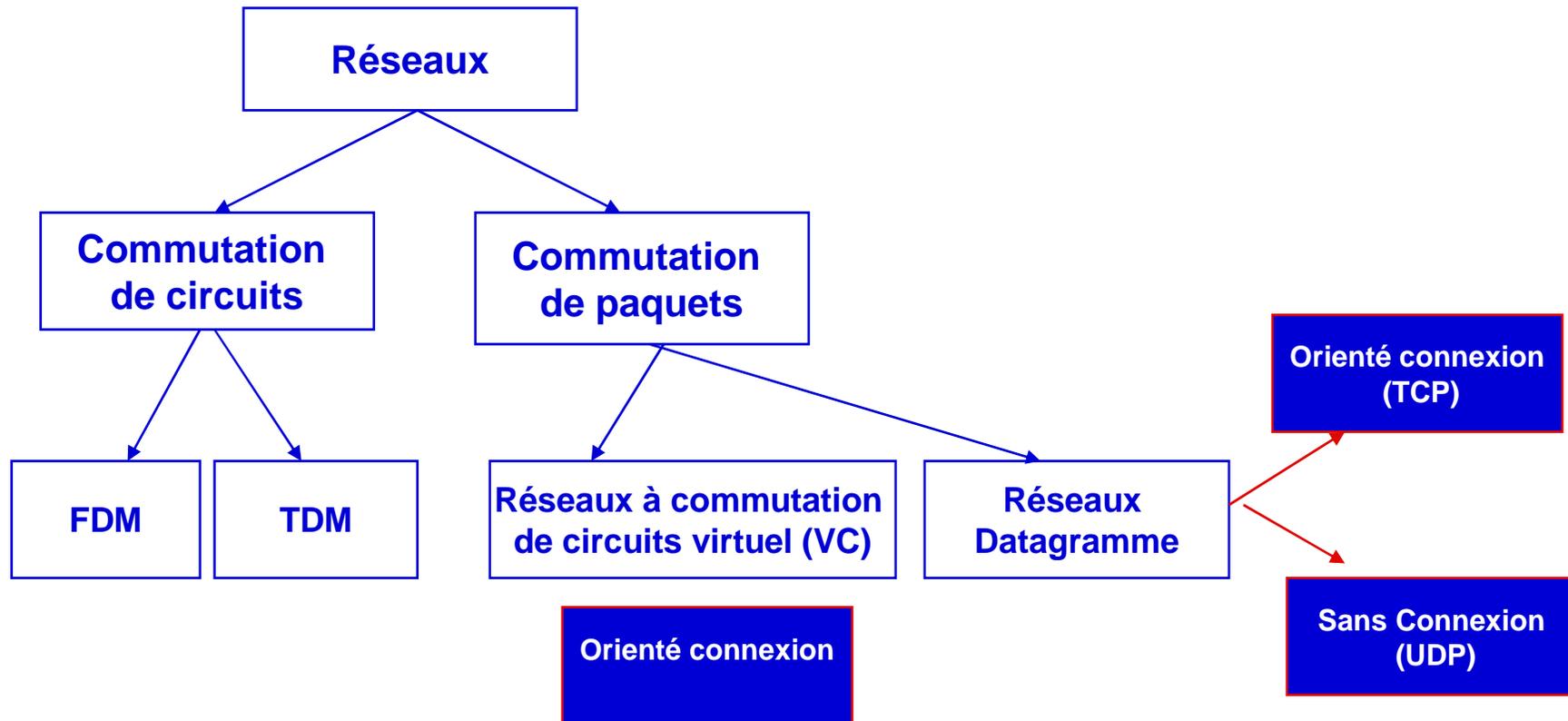
- Message de 7.5×10^6 bits
- Segmenté en 5000 paquets de 1500 bits
- Deux Commutateurs
- Trois liens de 1.5 Mbits/s

(pénalité Store and Forward = 1ms)



solution

- **Trois classes de concepts :**
 - **Commutation de circuits vs Commutation de paquet**
 - **Datagramme vs Circuit Virtuel**
 - **Orienté connexion vs Sans Connexion**



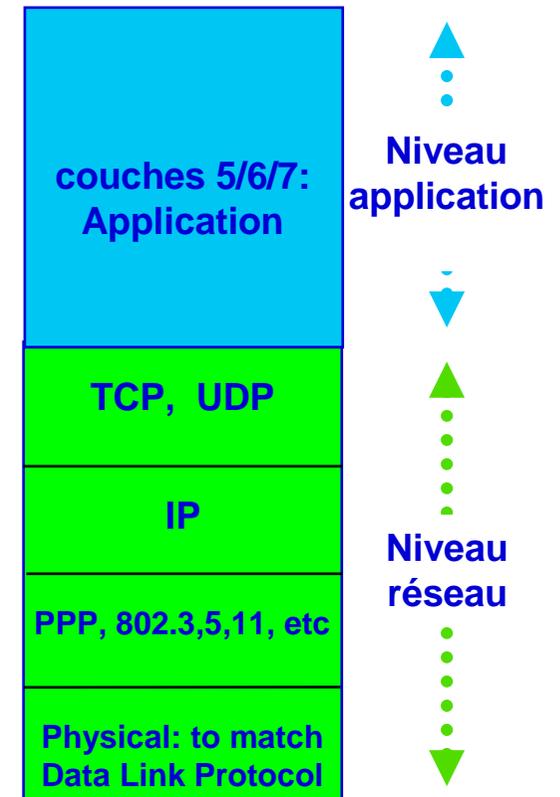
Structuration des protocoles en couches : modèles et services

- L'objectif principal est de réaliser le **transfert de données** entre deux processus communicants
- Messages “flots d'information” échangés :
 - ☞ **Entre les nœuds de réseau** .
 - Par exemple, entre commutateurs reliés au même LAN
 - Des routeurs reliés au même LAN ou WAN
 - ☞ **Entre les nœuds réseaux et des stations de managements**
 - ☞ **Messages de gestion de réseau** .
- **Toutes ces communications sont supportées par un ou plusieurs protocoles de transmissions**
 - On parle des fois de “ **pile des protocoles : stack**”
- **Un protocole définit :**
 - Le format et l'ordre des messages échangés entres les entités du réseau
 - Les actions entreprises suite à leur transmission / Réception

- Modèle fondé sur un principe énoncé par Jules César
 - **Diviser pour mieux régner**
- Le principe de base est la description des réseaux sous forme d'un ensemble de couches superposées les unes aux autres
- L'étude de tout est réduit à celle de ses parties, l'ensemble devient plus simple
- Le découpage du réseau en couches présente les avantages suivants :
 - 1) Il permet de diviser les communications sur le réseau en **éléments plus petits et plus simples**. ce qui permet de les comprendre plus facilement.
 - 2) Il permet à différents types de matériel et de logiciel réseau de communiquer entre eux (**interopérabilité**).
 - 3) Il empêche les changements **apportés à une couche d'affecter** les autres couches, ce qui assure un développement plus rapide.

Modèle en couches

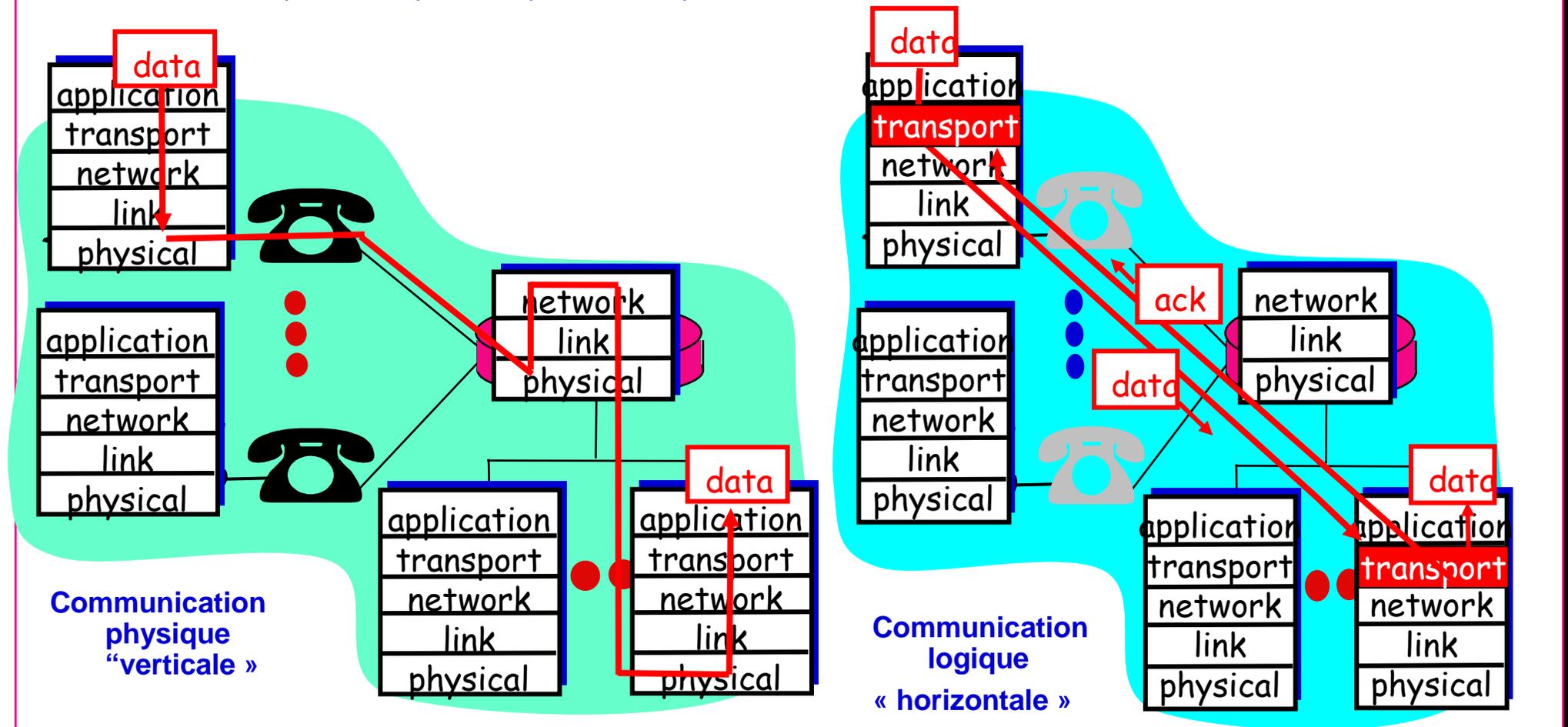
- Chaque couche fournit **un ou plusieurs services** à la couche immédiatement supérieure
 - Chaque service d'une couche est supportée par **un protocole de communication**
- Le protocole agit au niveau d'une couche qui communique avec une autre couche du même niveau dans un autre système d'extrémité
- **Les hôtes** ou **les systèmes d'extrémité** exécutent les protocoles de toutes les couches
- Exemple : les nœuds réseaux exécutent un ou plusieurs des protocoles des couches 1-3
 - **Routeurs, switches** et **hubs**



Modèle en couches

Au niveau des Entités :

- Récupérer les données d'une application
- Ajouter Adresse, des infos de contrôle "
- Envoyer le datagramme au correspondant
- Attendre que le récepteur acquitte la réception



Protocole, services et autres concepts de base

- **Entités Paires : Peer Entities**

- Deux (ou plus) instances du même niveau, dans deux systèmes distants

- **Protocole et PDU:**

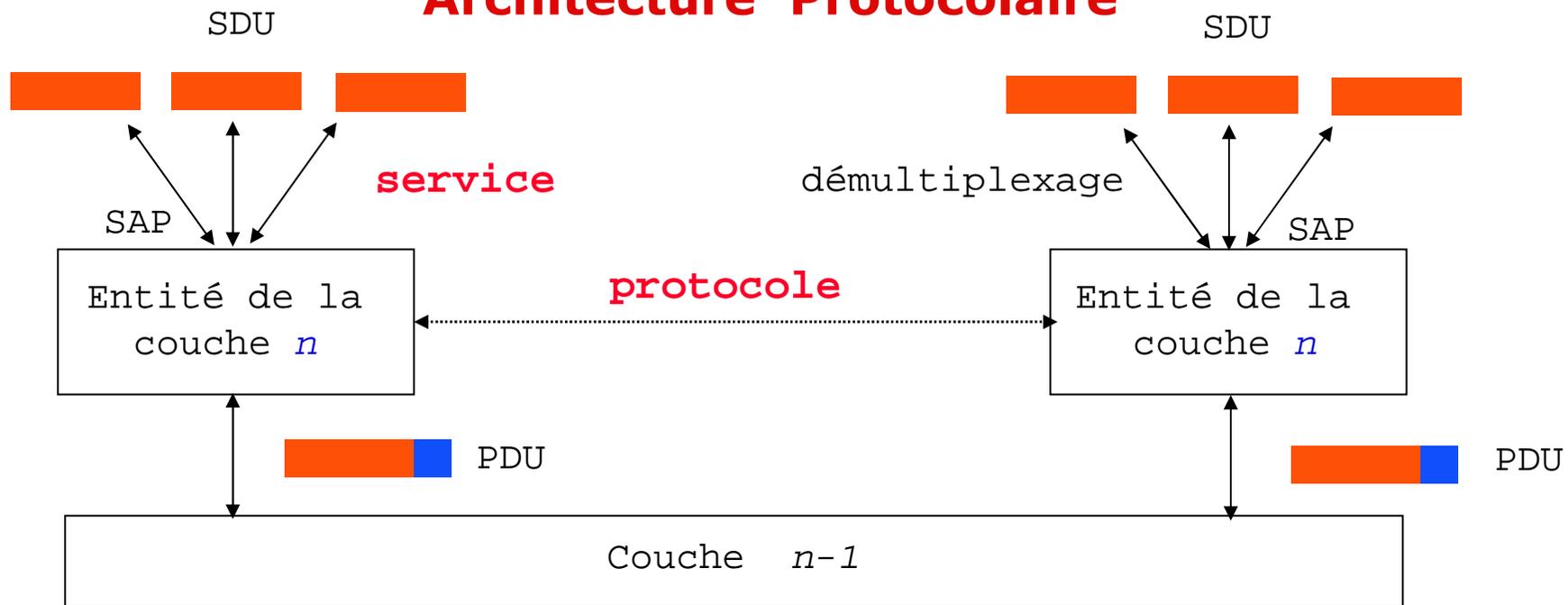
- Les règles qui régissent les opérations effectuées par les entités paires
- Les données échangées sont appelées **PDU** (Protocol Data Unit)
- Il y a un protocole (ou plus) dans chaque couche

Exemples de protocoles : TCP; UDP; IP; Ethernet

- **Service et SDU :**

- L'interface entre une couche et la couche immédiatement supérieure est appelée :
SAP (Service Access Point)
- **L'entité de données échangée** à travers cette interface est le **SDU** (Service Data Unit)

Architecture Protocolaire

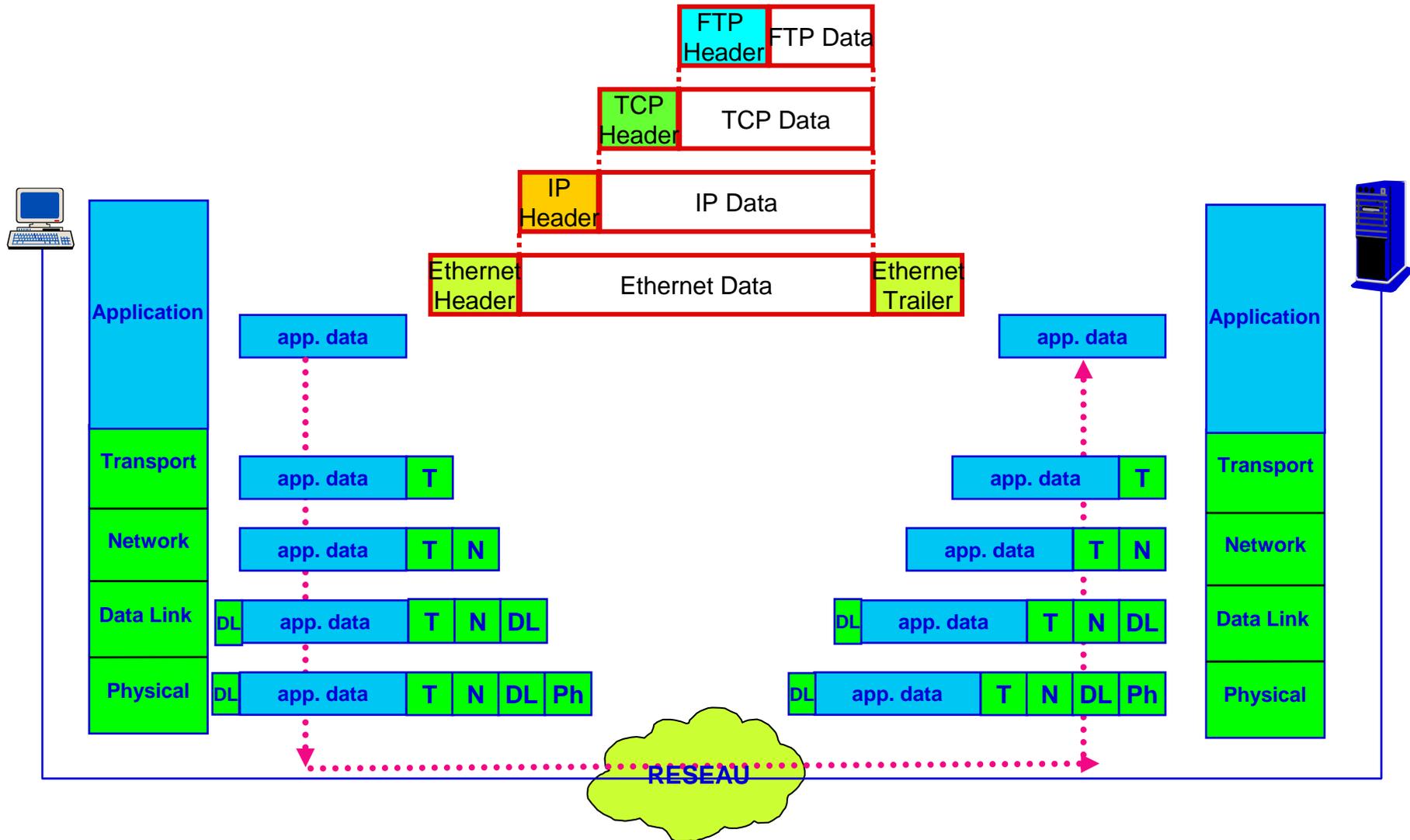


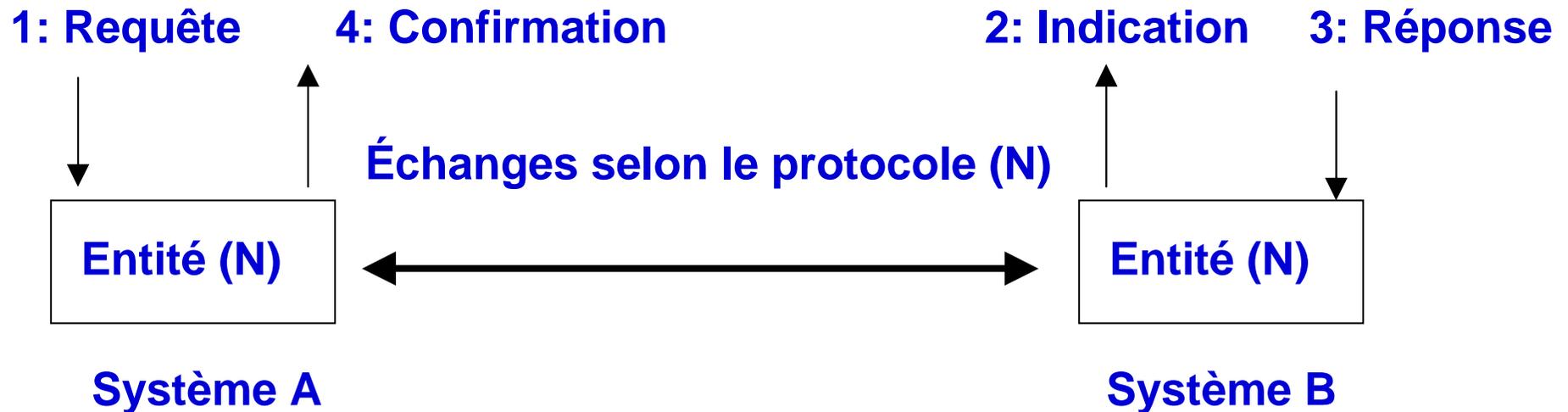
- ♦ **Les fonctions de réseau sont structurées sous la forme d'un modèle en couches :**
 - La couche n communique avec une autre entité de la couche n entités en utilisant les PDUs de la couche n
 - La couche n utilise les services de la couche $n-1$ et offre des services à la couche $n+1$.
 - Les entités distantes du même niveau sont appelés : **Entités paires** \Leftrightarrow peer entities
 - Les Règles d'opérations entre entités paires sont appelées : **protocole** \Leftrightarrow protocol

- ♦ **La mise en couche des protocoles des différents niveaux est appelée : Pile de protocoles " protocol stack".**

Encapsulation

- Le PDU de la couche N est transféré comme SDU à la couche N-1



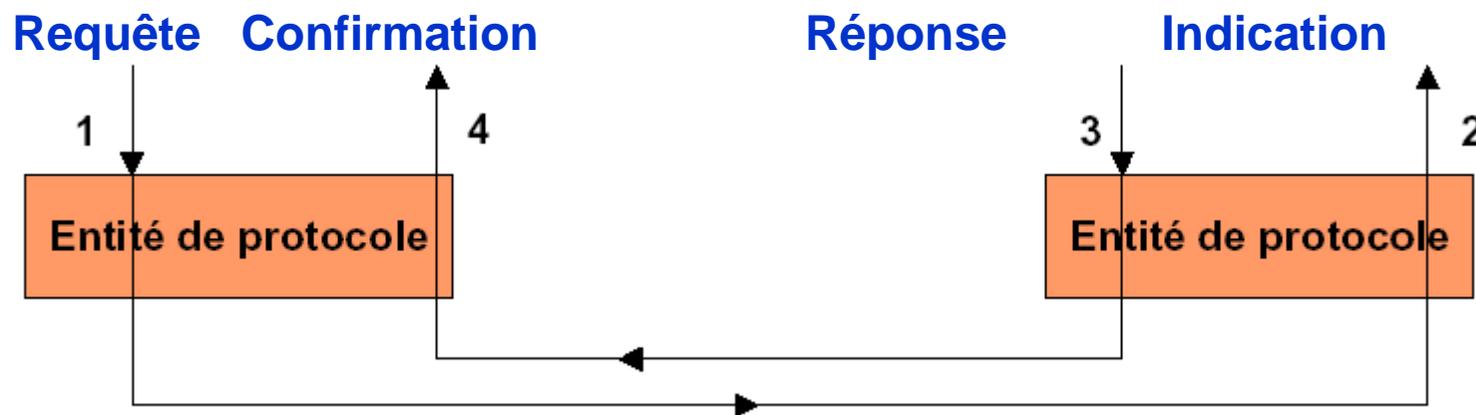


- **La requête** : l'entité (N) demande à l'entité (N-1) d'activer un service;
- **L'indication** : l'entité (N-1) avise l'entité (N) de l'activation d'un service;
- **La réponse** : c'est la réponse de l'entité (N) à une indication reçue;
- **La confirmation** : l'entité (N-1) avise l'entité (N) que le service demandé est actif.

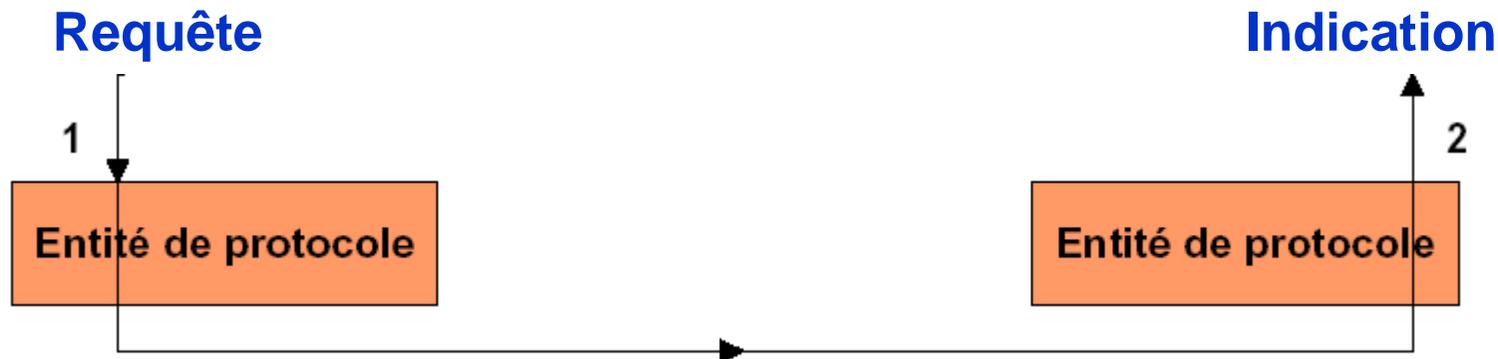
Synchronisation de l'échange par des PDU de contrôle –1-

- **Connexion (N)** : Association logique entre deux entités (N) de systèmes Différents.
- **Mode Connecté** : comprend 3 phases
 1. Établissement de la connexion
 2. Transfert des données
 3. Libération de la connexion

Il est adapté aux applications de **longues interactions** (Transfert de gros fichiers, Chat, courriel,...)



- **Mode non connecté** : envoi d'une lettre postale
 - Permet l'envoi de données **sans l'établissement** d'une connexion au préalable
 - Chaque paquet est indépendant des autres et peut suivre plusieurs chemins différents
 - On peut avoir un mode non-connecté avec accusé de réception

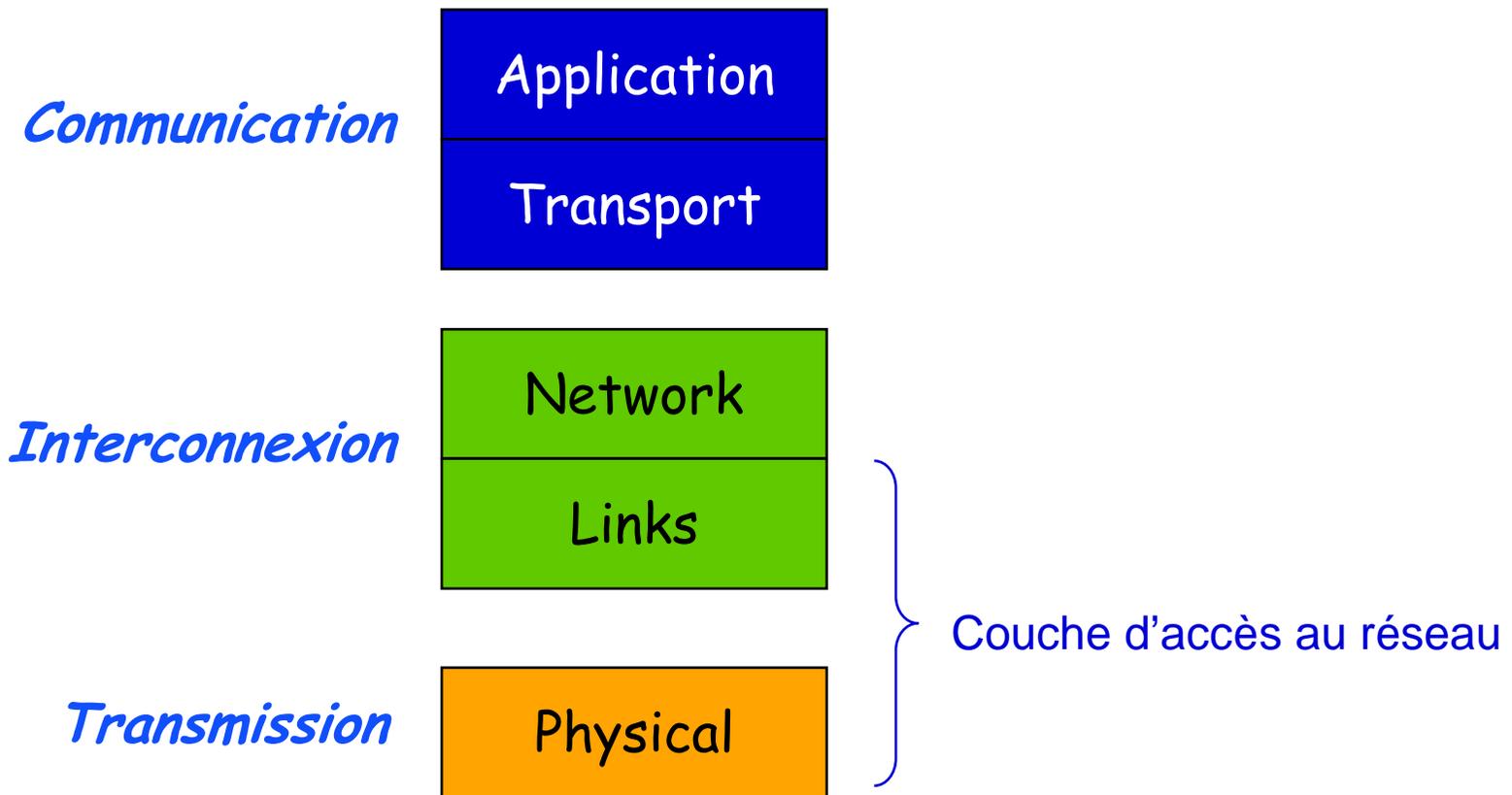


- **Contrôle de flux : Rôles :** Les fonctions de contrôle de flux agissent :
 1. **Localement sur les PDU (entre 2 nœuds ou entre une station et un nœud).**
 2. **De bout en bout entre des entités appartenant à des couches hautes homologues**
- **Localisation de la fonction de Contrôle de flux :**
 - A. Le contrôle de flux entre **entités « distantes »** homologues détermine la cadence d'envoi des PDU(N) sur une connexion (N);
 - il est défini par le protocole (N) et est basé sur la taille des PDU(N)
 - B. Le contrôle **de flux à l'interface (N)** détermine la cadence de transfert des données d'une entité (N+1) à une entité (N) à l'interface (N);
 - il est basé sur la taille des SDU(N).
- **Contrôle de congestion :**
 - Ensemble des activités de contrôle de flux de toutes les couches et de tous les **éléments actifs du réseau**
 - L'objectif est de prévenir la congestion globale du réseau. Activité en général **supervisée par un système de gestion**

Le modèle TCP/IP

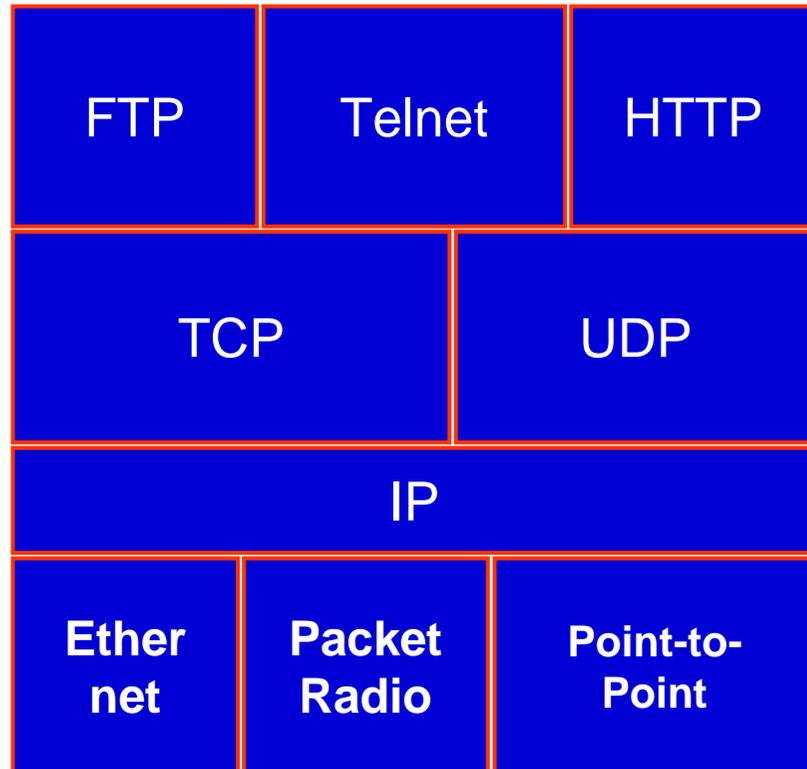
- ◆ Développé dans le contexte d'ARPANET et financé par le Département de Défense américain
- ◆ Les objectifs de l'ARPANET étaient :
 - L'interconnexion des réseaux (incompatibles ! des universités et instituts de recherche) de manière transparente (internet)
 - Grande tolérance aux pannes
 - Architecture souple, appropriée pour des applications très différentes
- ◆ L'architecture et les protocoles sont maintenant utilisés dans Internet

- TCP/IP est modèle **d'architecture en couches**

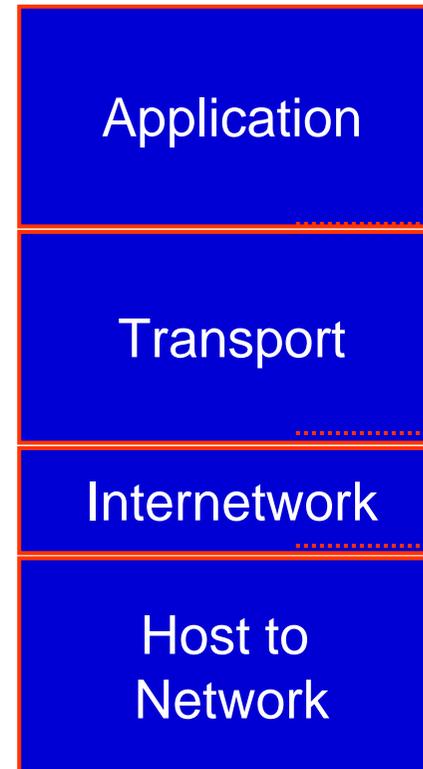


Modélisation TCP/IP – OSI Comparaison

Protocole TCP/IP



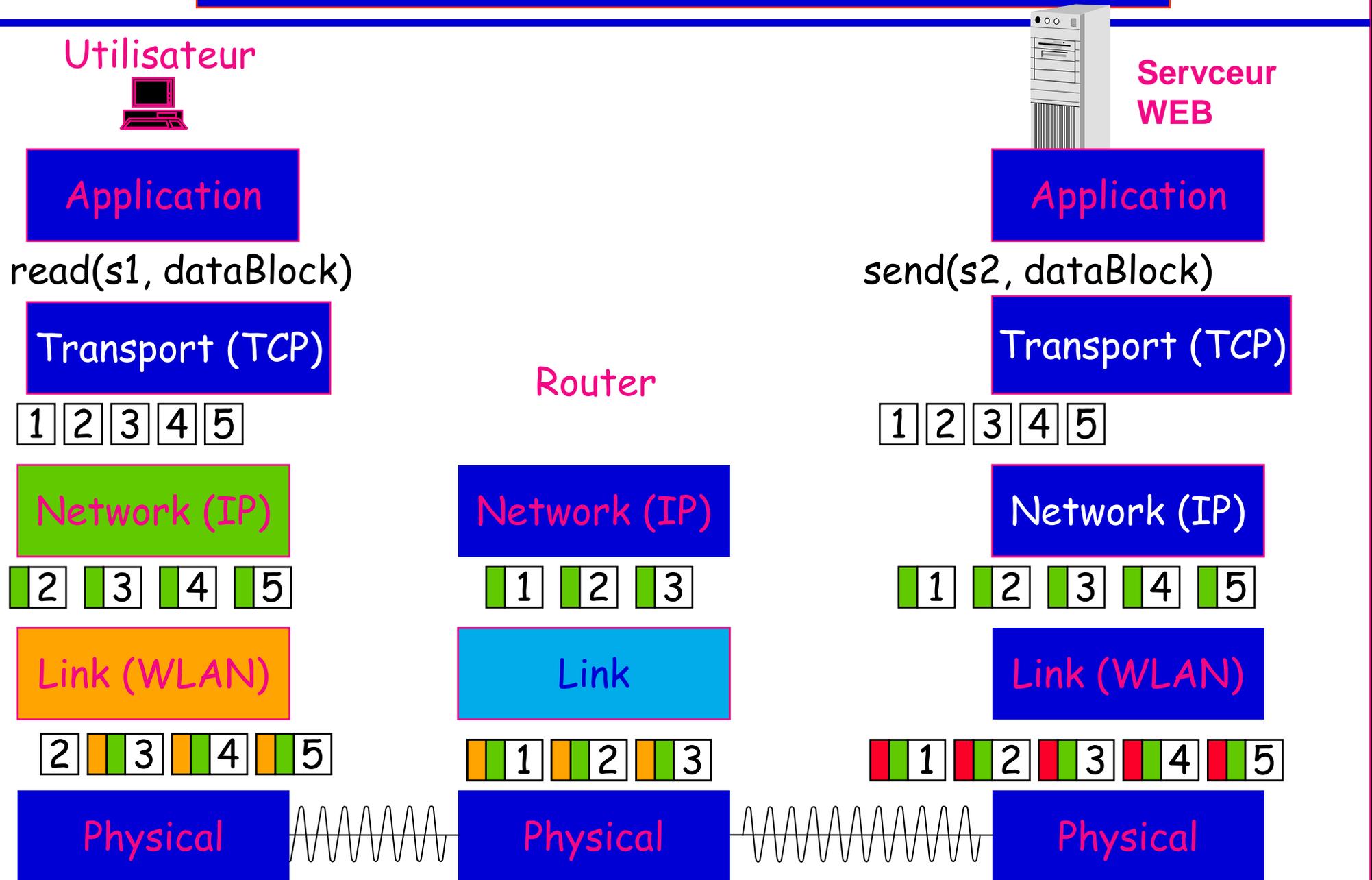
Modèle TCP/IP



Modèle OSI

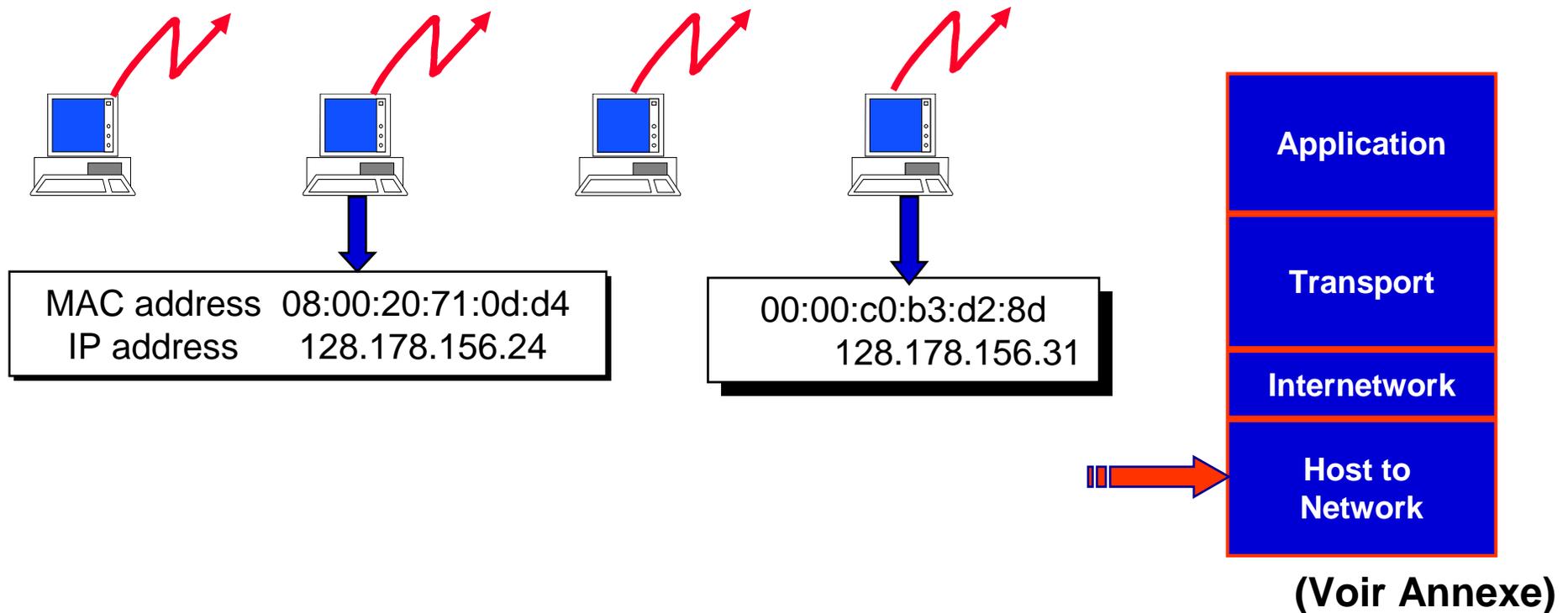


TCP/IP : Encapsulation des données – principe -



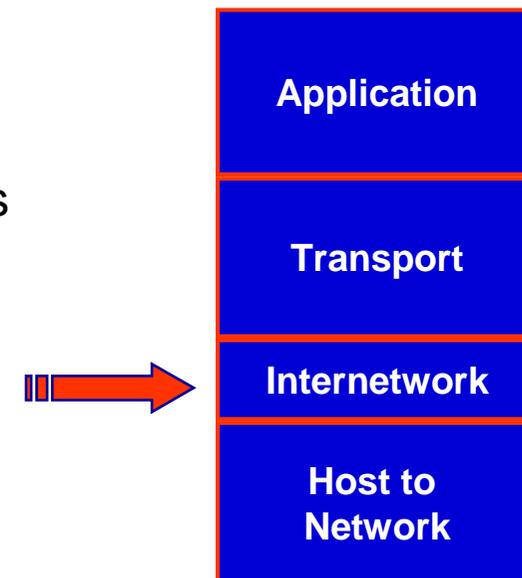
Couche accès au réseau

- ◆ Cette couche est responsable de la **transmission fiable** des trames de données à travers les supports de communication physiques.
- ◆ Un hôte peut supporter **plusieurs protocoles de réseau**.
- ◆ La transformation des adresses IP en **adresses physiques** se fait à ce niveau.

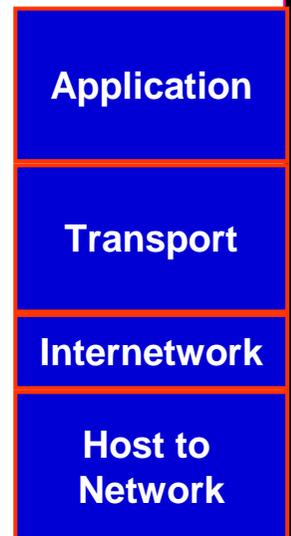


Couche inter-réseau (Internet)

- ❑ Le protocole IP (*Internet Protocol*) est basé sur un modèle de **commutation de paquets**.
- ❑ Il s'agit d'un protocole en mode **sans connexion**.
- ❑ Les fonctions de cette couche sont:
 1. Définition d'une méthode d'adressage pour le **réseau entier**
 2. Acheminement des données entre la couche transport et les couches réseaux
 3. Routage des datagrammes vers les destinataires
 4. Fragmentation et réassemblage des datagrammes

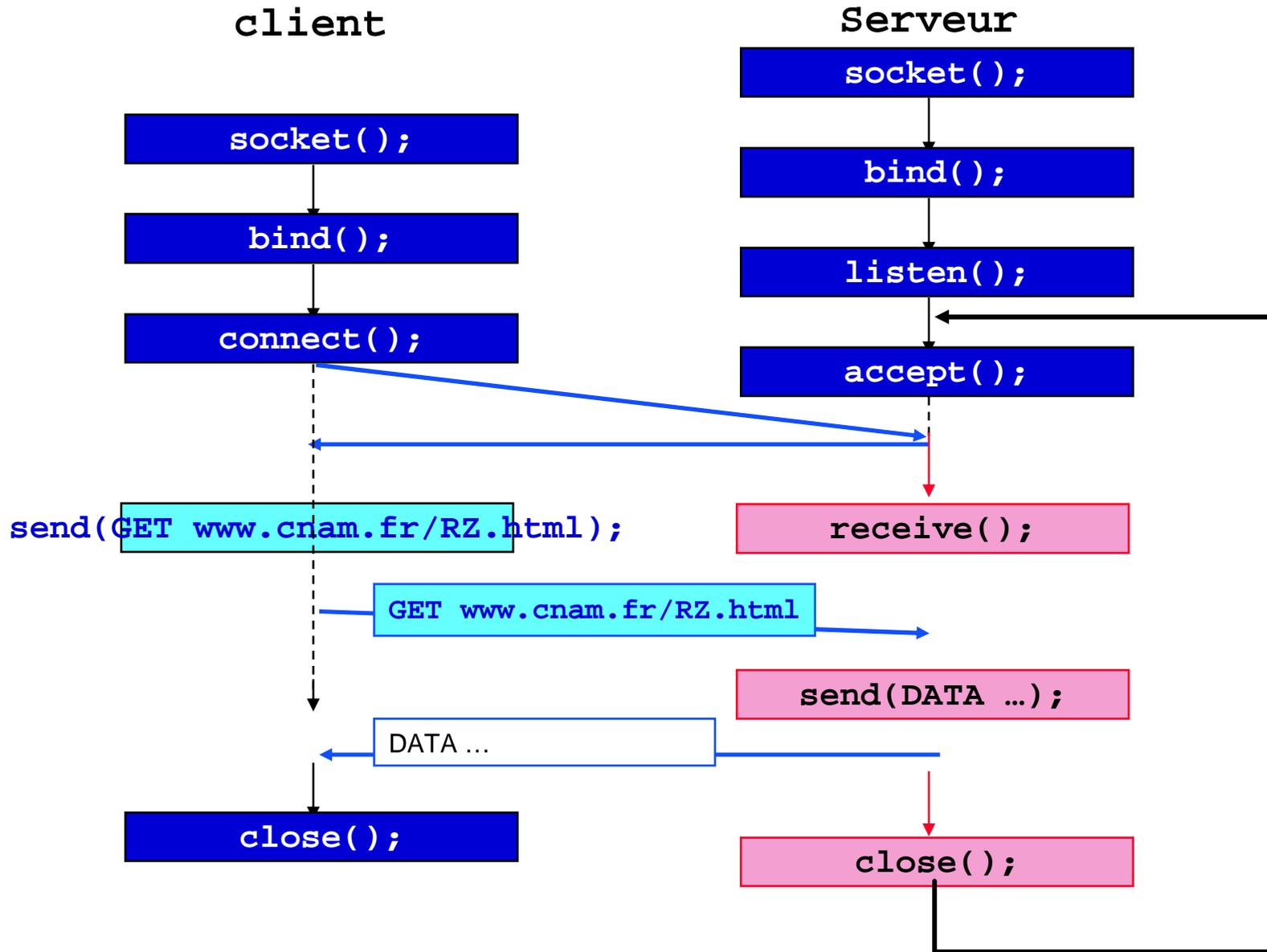


- ❑ Le protocole *TCP* assure un **circuit virtuel** entre les applications utilisateurs. Ses caractéristiques sont les suivantes :
 - Il est orienté connexion.
 - Il est fiable.
 - Il divise les messages sortants en segments.
 - Il assemble des messages au niveau de la station de destination.
- ❑ *UDP* transporte les données de manière non fiable entre les hôtes. Les caractéristiques du *protocole UDP* sont les suivantes :
 - Il n'est pas orienté connexion.
 - Il est peu fiable.
 - Il transmet des messages (appelés datagrammes utilisateurs).
 - Pas de vérification logicielle pour la livraison des messages (non fiable).
 - Il n'assemble pas les messages entrants.
 - Il n'utilise pas d'accusés de réception.
 - Il n'assure aucun contrôle de flux.



La couche transport aide la couche Application

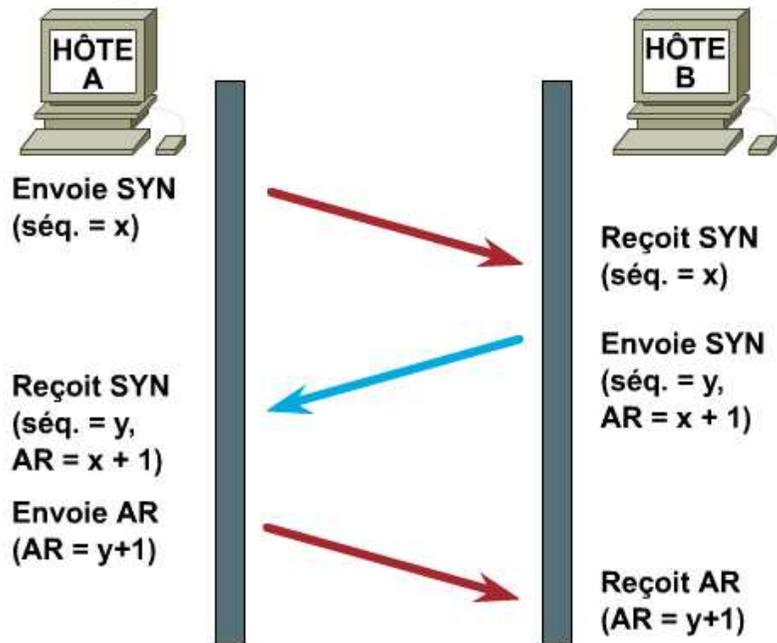
- La couche transport fournit une interface de programmation pour la couche application
- Les applications peuvent utiliser soit UDP, soit TCP en fonction des besoins
- Les interfaces de programmation s'appellent : **L' API Sockets**



Internet : Pile de protocoles

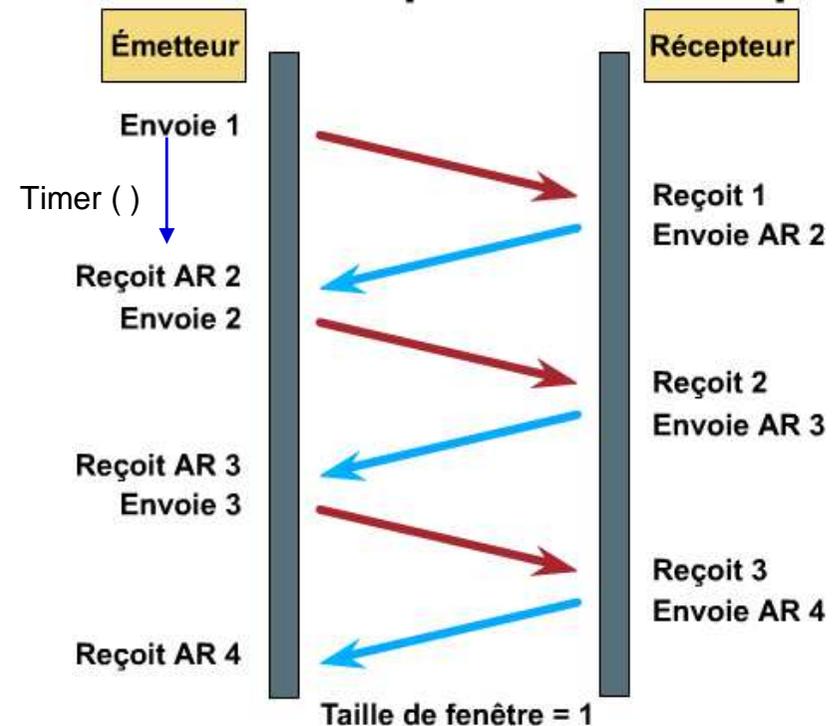
Exemple : connexion TCP avec échange en trois phase

Établissement d'une connexion TCP avec Trois (three way handshake) échanges



synchronisation d'une connexion

Accusé de réception TCP simple



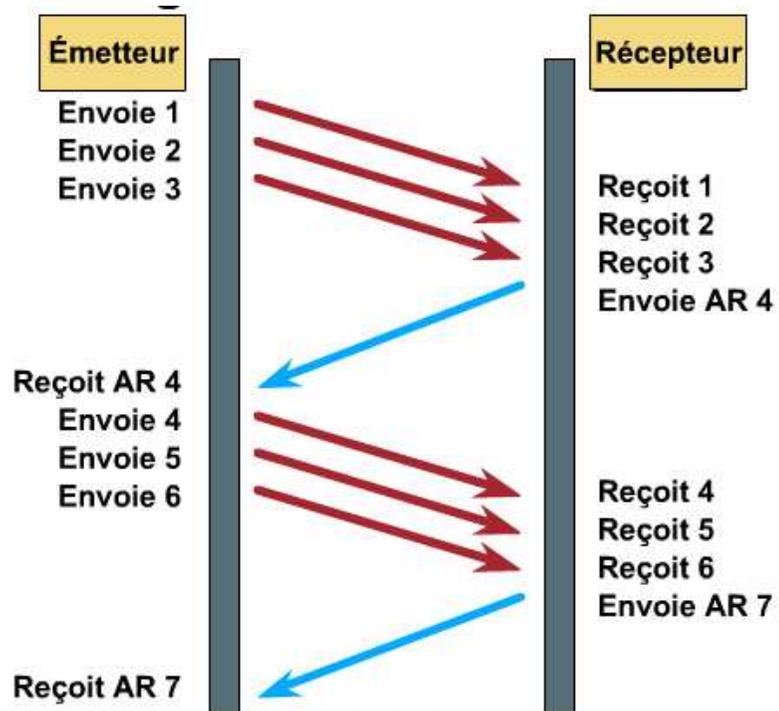
◆ Principe de la technique PAR (Positive Acknowledgement and Retransmission) :

- la source envoie un paquet, démarre un compteur et attend un accusé de réception avant d'envoyer le paquet suivant
- Si le compteur arrive à expiration avant que la source n'ait reçu un accusé de réception, elle retransmet le paquet et redémarre le compteur.

Internet : Pile de protocoles

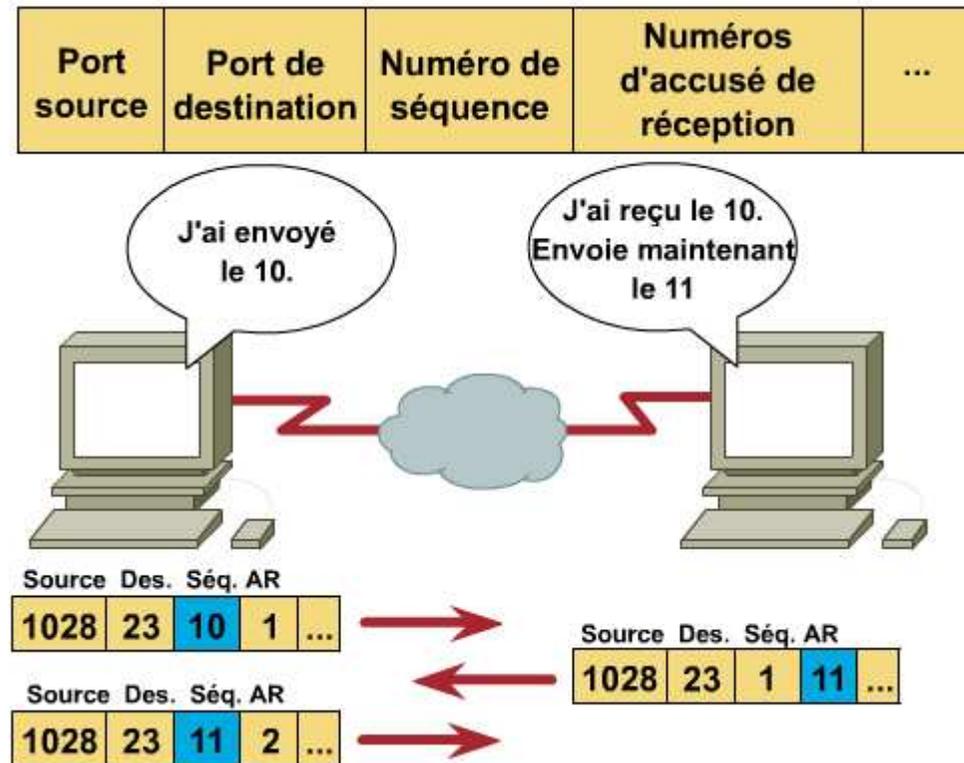
Exemple : connexion TCP avec échange en trois phase

Fenêtre glissante TCP



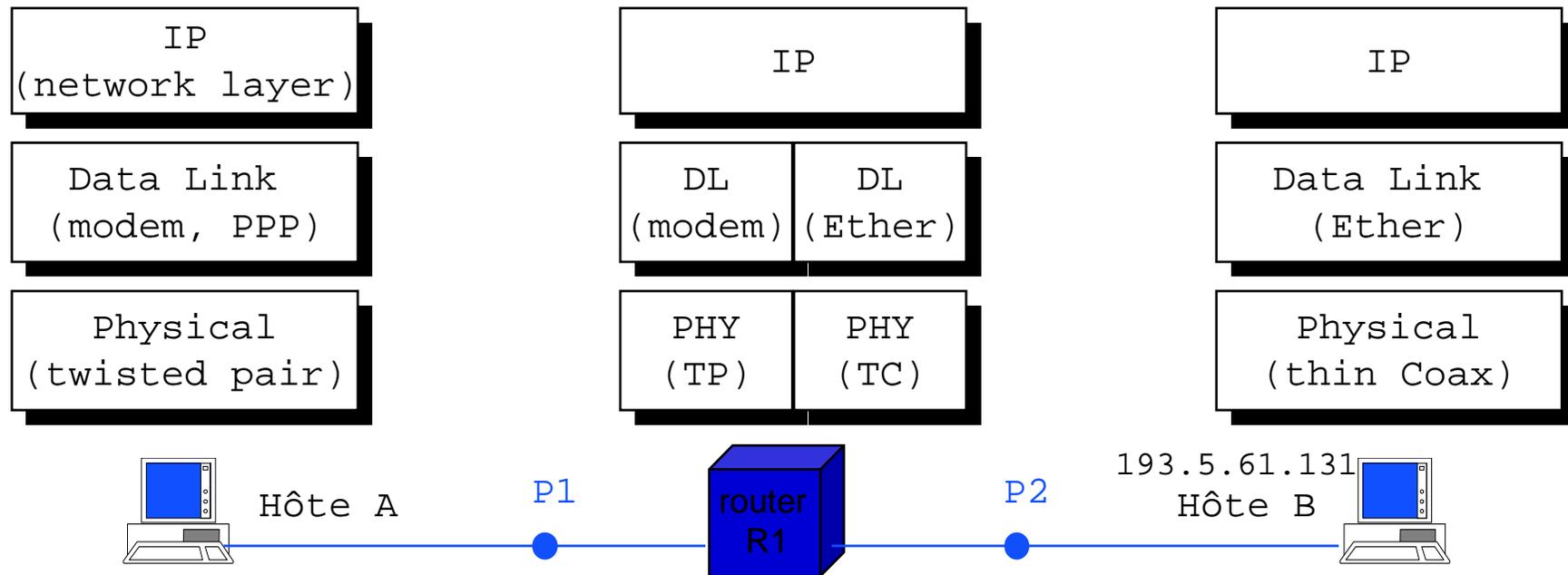
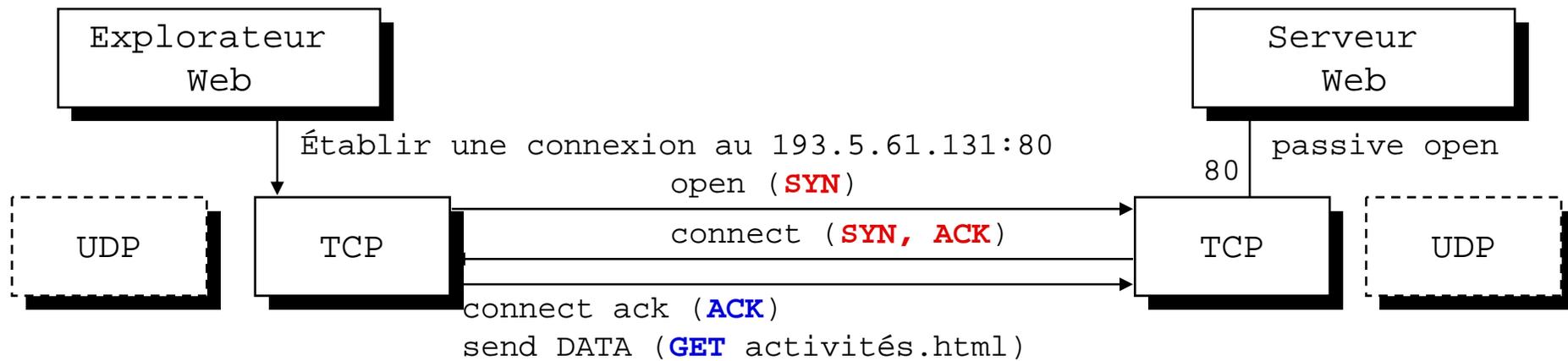
taille de fenêtre est de trois

Numéro de séquence et accusé de réception TCP

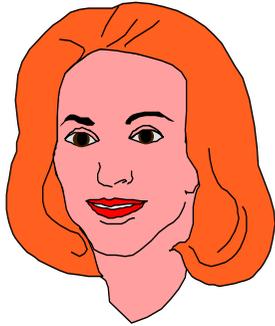


Le protocole TCP assure le séquençage des segments

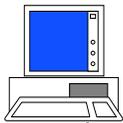
Exemple : avec TCP



Couche application



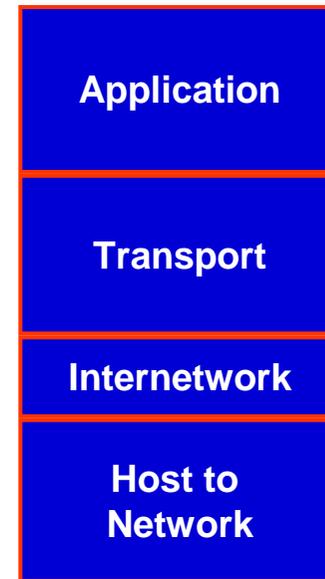
Utilisateur :
`http://www.cnam.fr/RZ.html`



IP addr = 193.5.61.131
`GET www.cnam.fr/RZ.html`

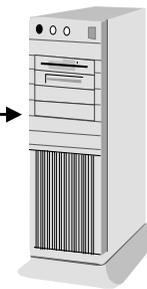
data (HTML page)

- Utilisent des protocoles “well defined protocols”
- ex: HTTP



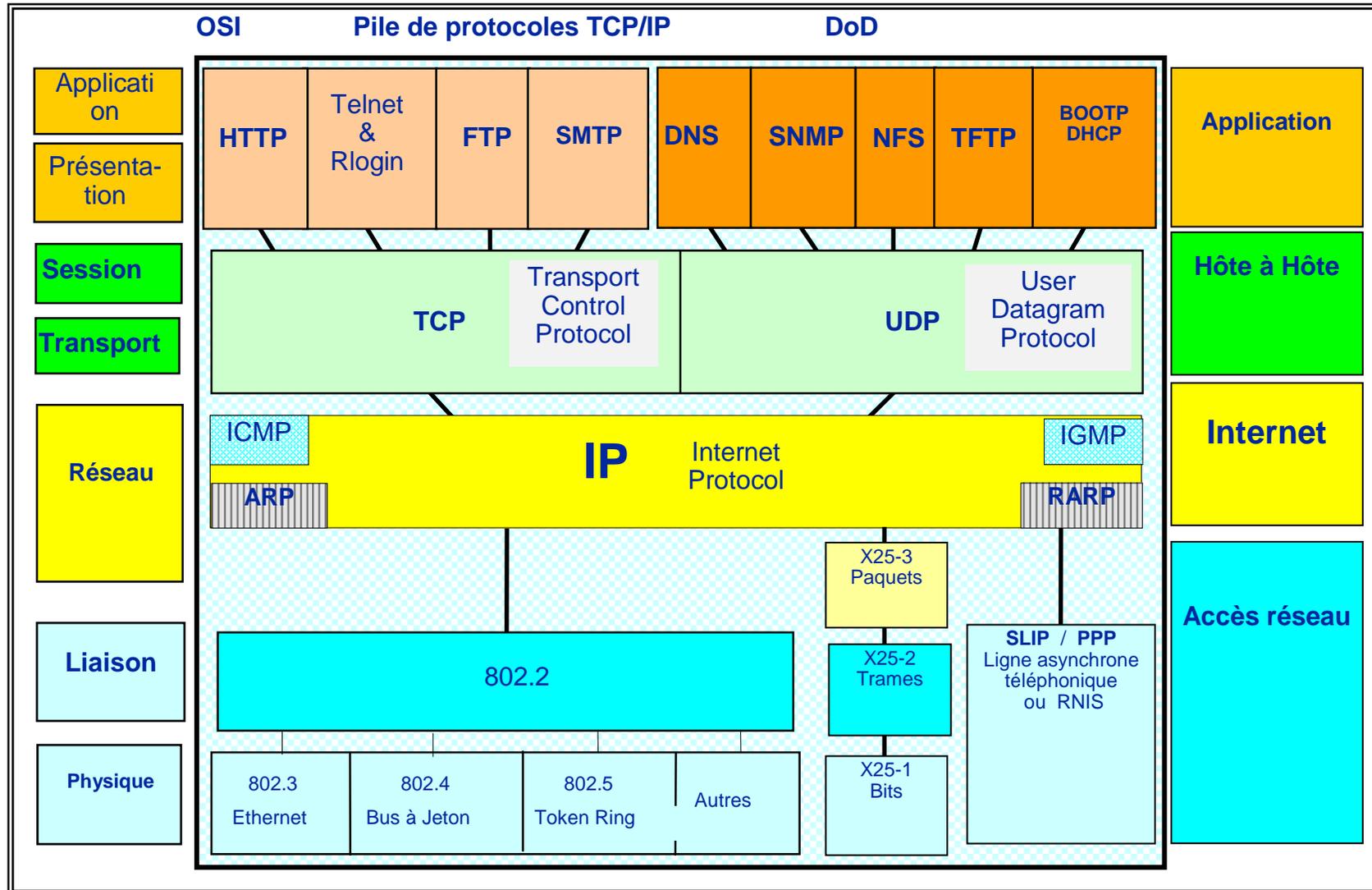
- FTP
- Telnet
- SMTP
- HTTP

Serveur Web



- ❑ Les protocoles TCP et UDP utilisent des **numéros de port** (ou de socket) pour transmettre des informations aux couches supérieures.
- ❑ Les **numéros de port** servent à distinguer les différentes conversations qui circulent simultanément sur le réseau.
 - Numéros de ports bien connus (RFC1700)
 - *Exemple : FTP utilise le numéro de port standard 21.*
- ❑ Les conversations qui ne visent pas des applications ayant des numéros de port reconnus se voient attribuer des **numéros aléatoires sélectionnés** à l'intérieur d'une plage donnée.
- ❑ Certains ports sont réservés au sein des protocoles TCP et UDP , bien que les applications ne soient pas nécessairement écrites pour les supporter. Les plages attribuées aux numéros de port sont les suivantes :
 - **Numéros inférieurs à 255** - réservés aux applications publiques.
 - **Numéros entre 255 et 1023** - attribués aux entreprises pour les applications commercialisables.
 - **Numéros supérieurs à 1023** - ne sont pas attribués.

Internet : Pile de protocoles



<http://www.iana.org/assignments/port-numbers>

Exemple d'un protocole applicatif : Serveur de noms de domaine (DNS)

Exemple : Serveur de noms de domaine (DNS)

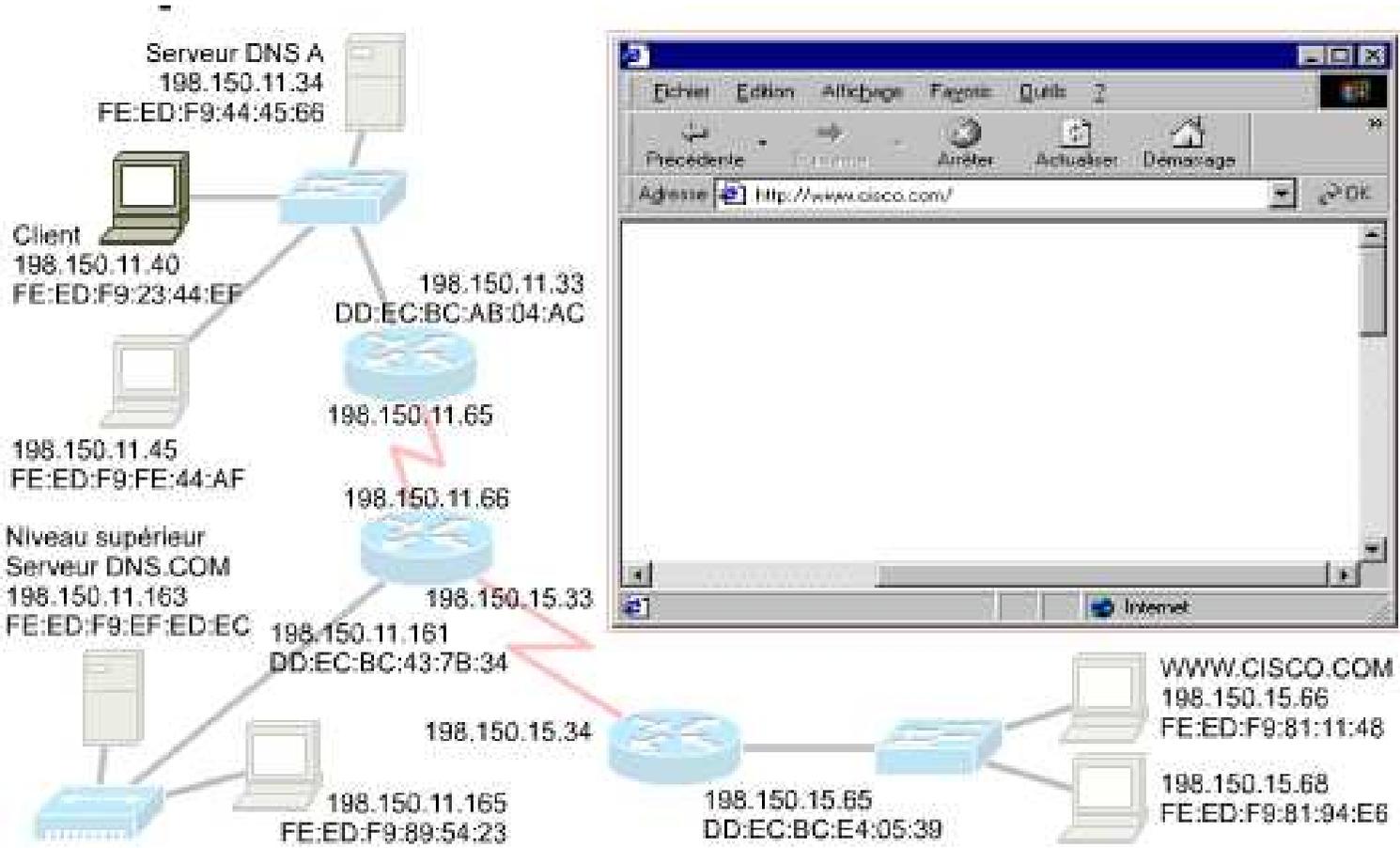
Rappel :

- Le serveur de noms de domaine (DNS) est un élément du réseau. Il répond aux demandes des clients portant sur la traduction des noms de domaine en adresses IP.
 - Le système DNS est une **structure hiérarchique** qui crée des serveurs DNS de différents niveaux.
 - Si un serveur **DNS local** est capable de traduire un nom de domaine en adresse IP, il le fait et retourne le résultat au client.
 - Dans le cas contraire, il achemine la demande au serveur **DNS de niveau immédiatement supérieur** qui tente, à son tour, de traduire l'adresse.
 - Si ce nouveau serveur DNS est capable de traduire le nom de domaine en adresse IP, il le fait et retourne le résultat au client. Sinon, il achemine la demande **au serveur de niveau immédiatement supérieur**.
 - Ce processus se répète jusqu'à ce que le nom de domaine soit traduit ou jusqu'à ce que la demande atteigne le serveur DNS du plus haut niveau.
 - Un nom de domaine introuvable au plus haut niveau est considéré comme erroné et le serveur renvoie un message d'erreur. **Toutes les applications qui utilisent des noms de domaine pour représenter des adresses IP font appel au système DNS pour traduire ces noms en adresses IP correspondantes**

Exemple : Serveur de noms de domaine (DNS)

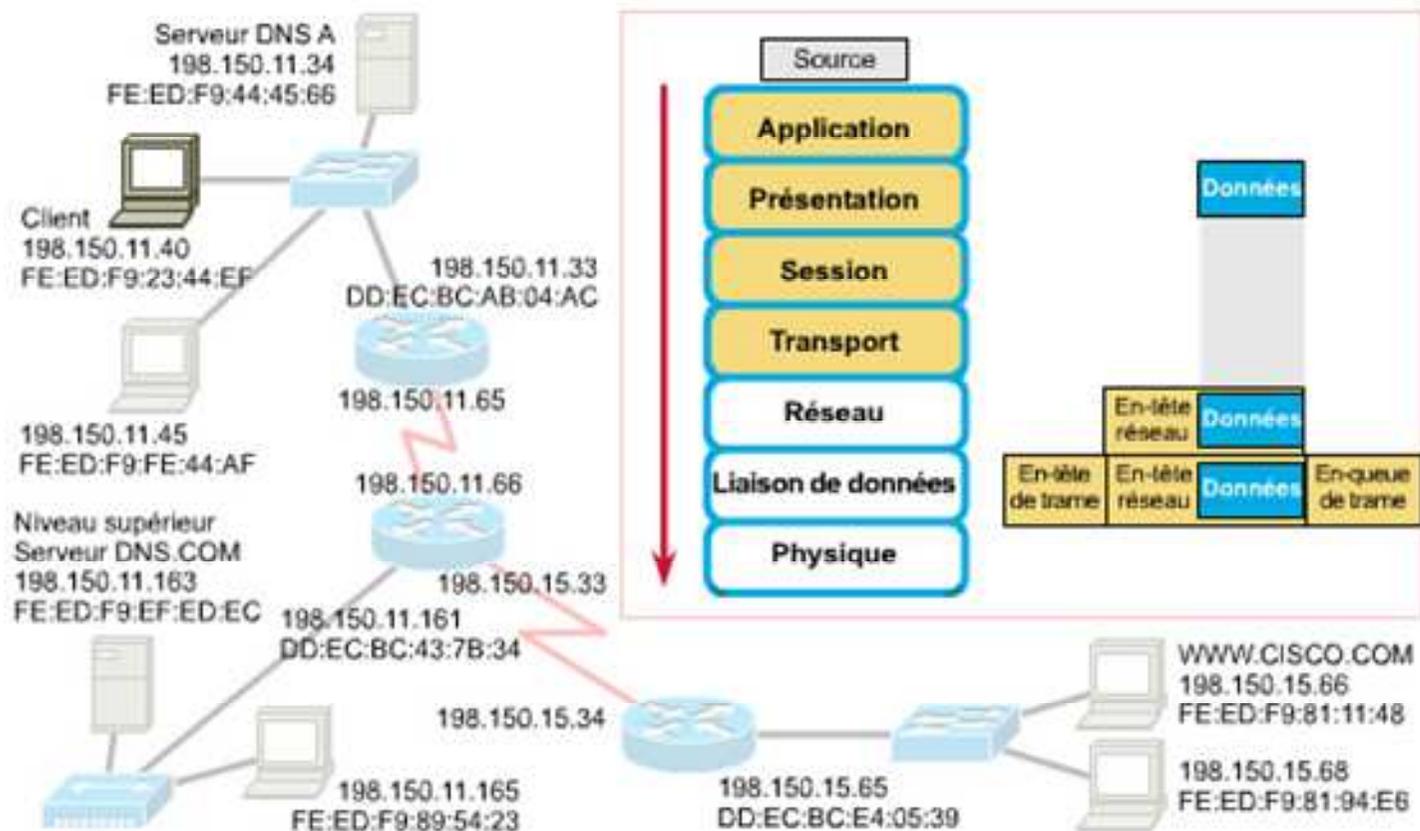
Séquence de recherche DNS

Le client doit accéder au nom de domaine : www.cisco.com/fr/



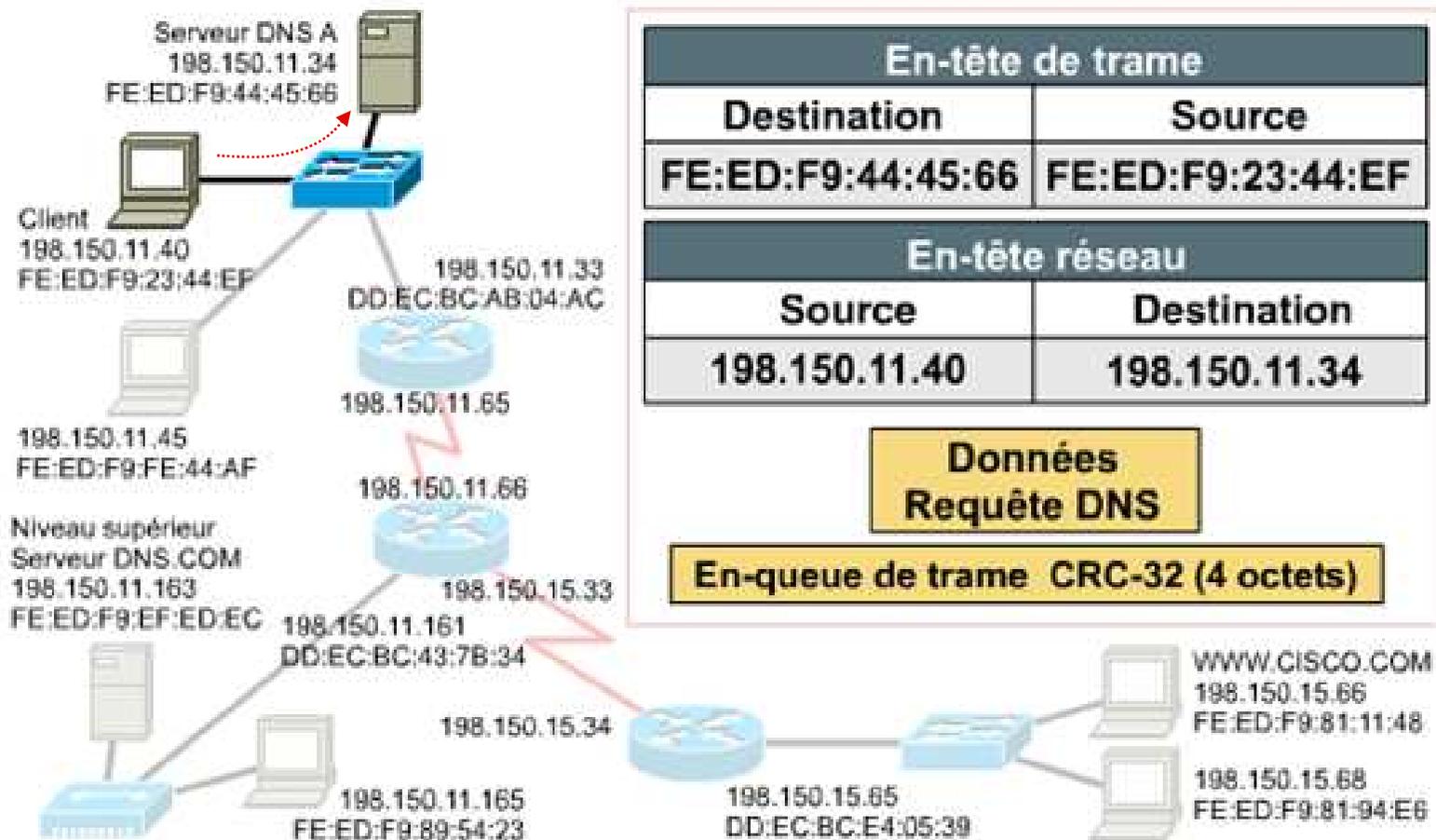
Exemple : DNS – Etape 1

- Le navigateur, à l'aide de la pile de communication **crée ensuite les paquets** à envoyer au serveur DNS A.
- L'adresse IP du serveur est stockée dans le fichier de **configuration TCP/IP**, alors que l'adresse MAC est déjà dans la mémoire **cache ARP** suite à une transaction antérieure



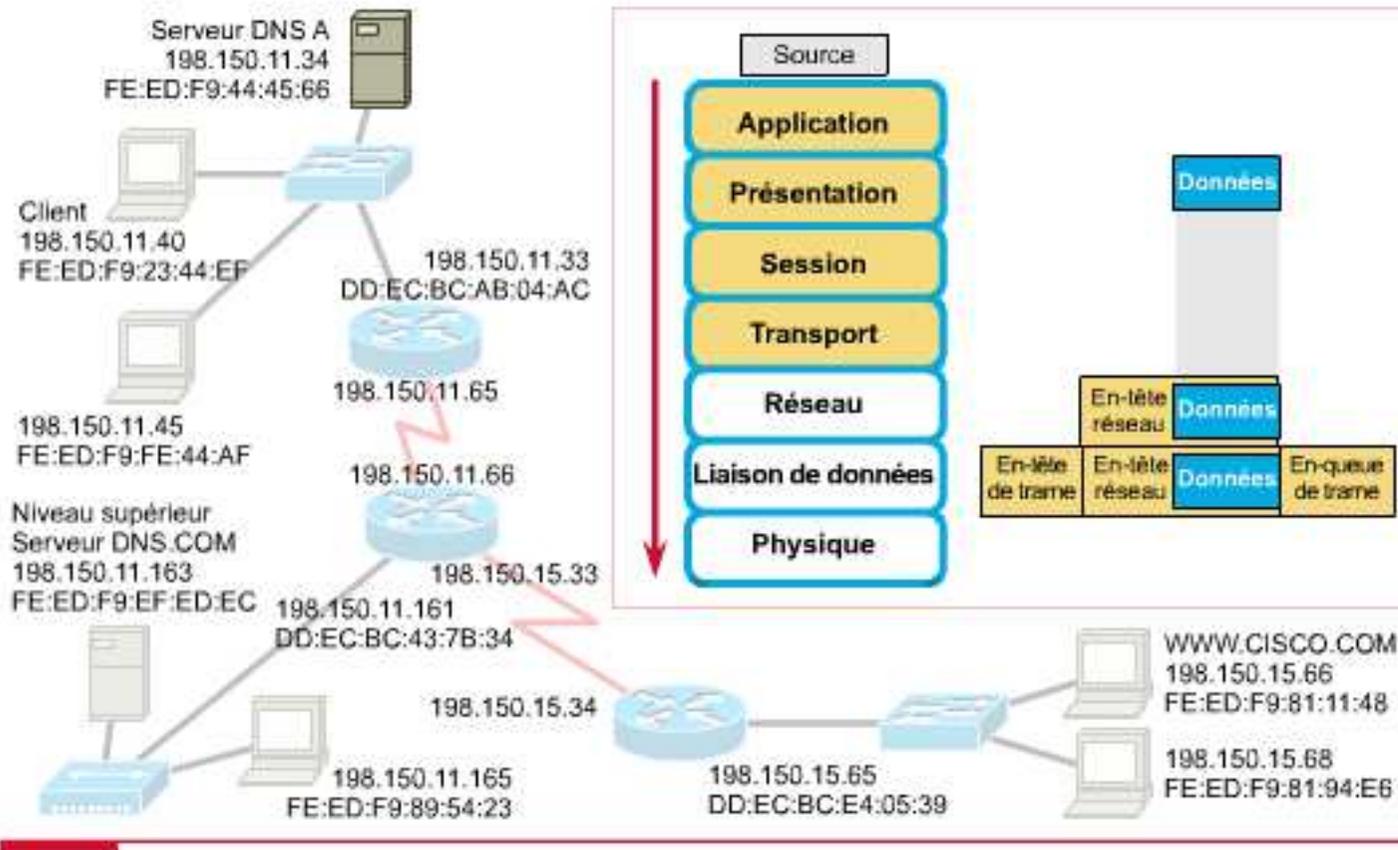
Exemple : DNS – Etape 2

Les trames de données sont alors transmises au **serveur DNS A**, sur le segment Ethernet, par le biais du commutateur



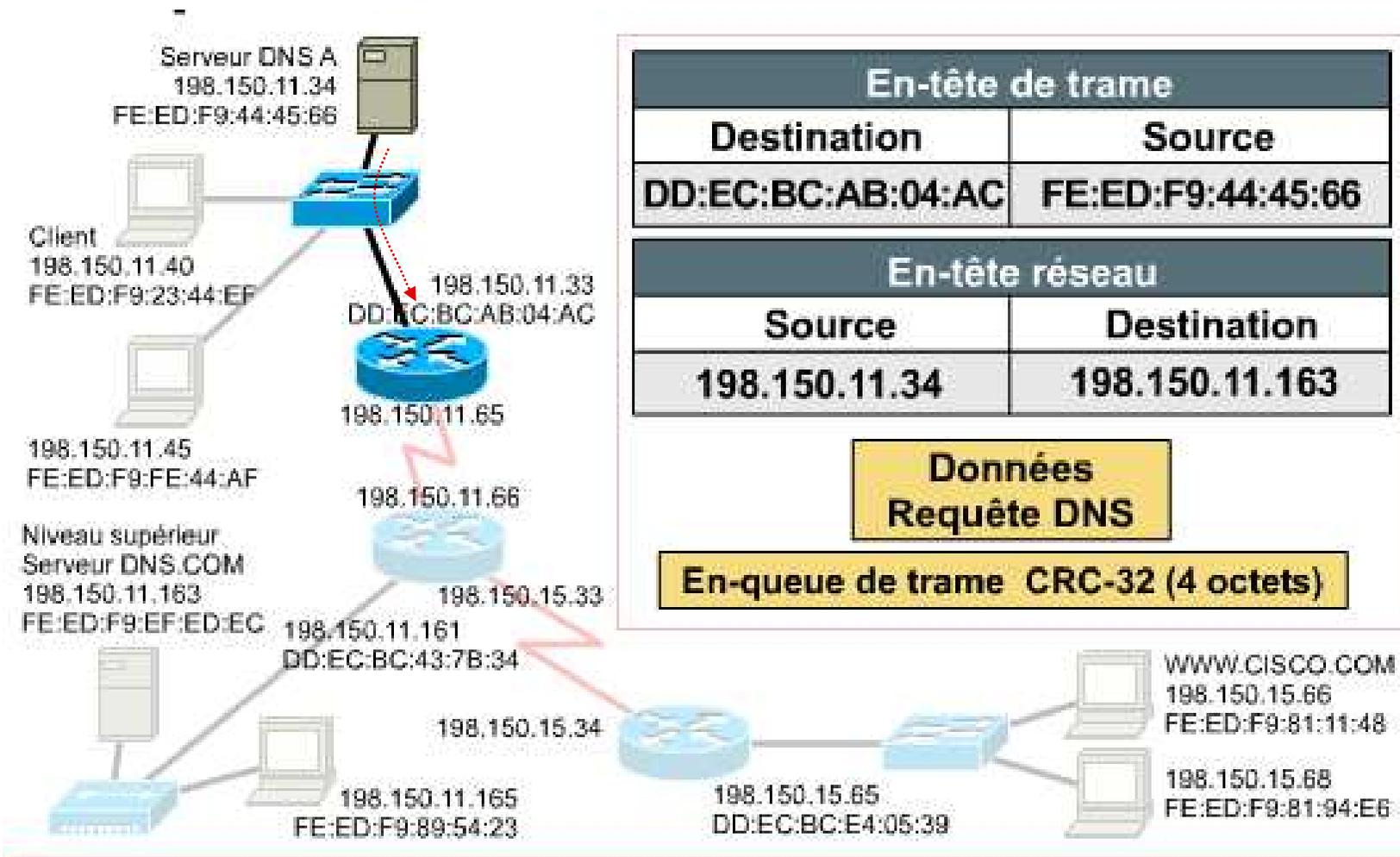
Exemple : DNS – Etape 4

- Le serveur DNS A, à l'aide de la pile de communication, **crée ensuite les paquets** à envoyer au serveur DNS.COM.
- L'adresse est déjà dans la mémoire **cache ARP** suite à une transaction antérieure.
- Comme le serveur DNS.COM se trouve sur un sous-réseau différent, l'adresse MAC dans la mémoire cache ARP est celle du **routeur associé à l'adresse IP** du serveur DNS.COM



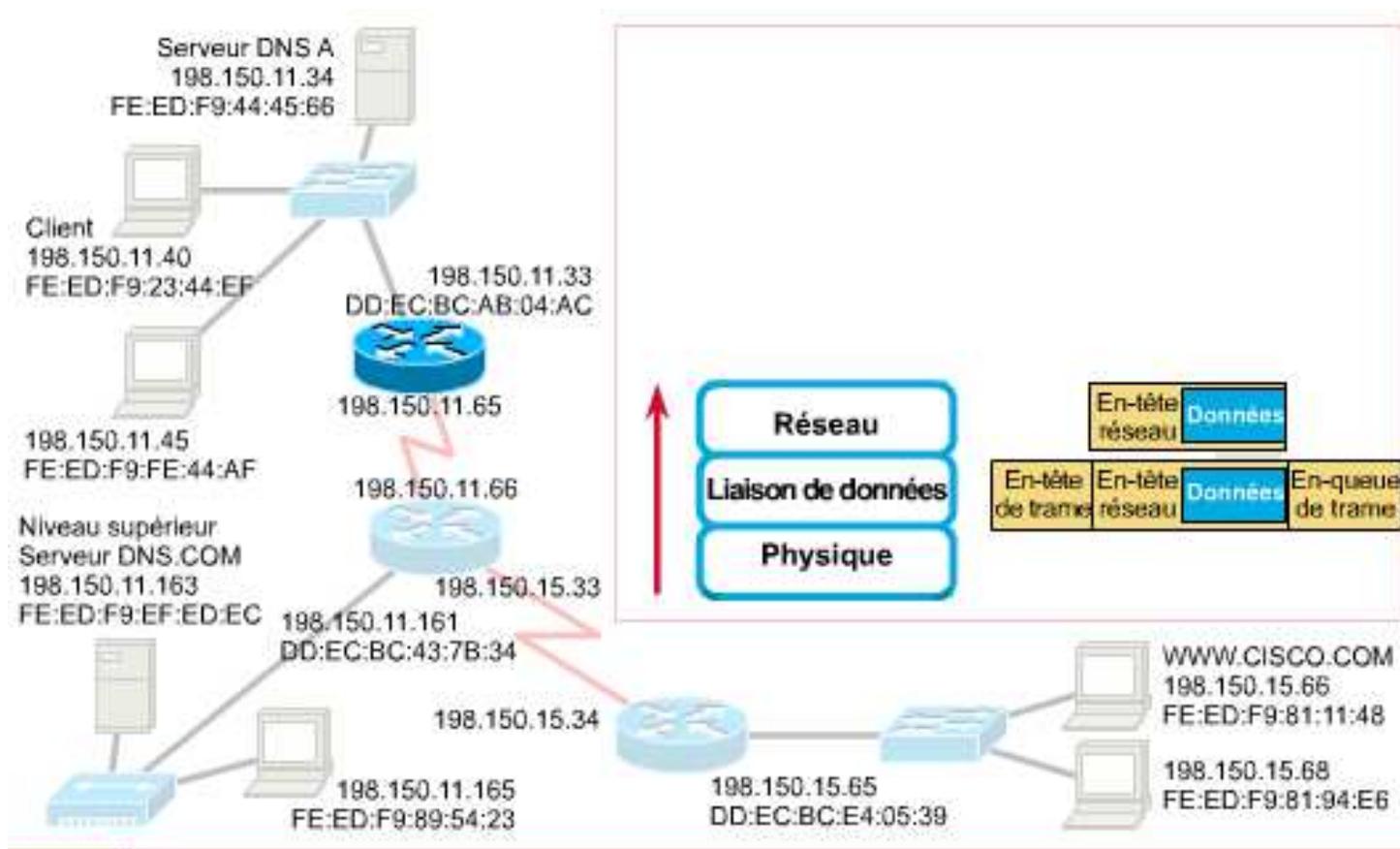
Exemple : DNS – Etape 5

- Les trames de données sont alors transmises au routeur, sur le segment Ethernet, par le biais du commutateur



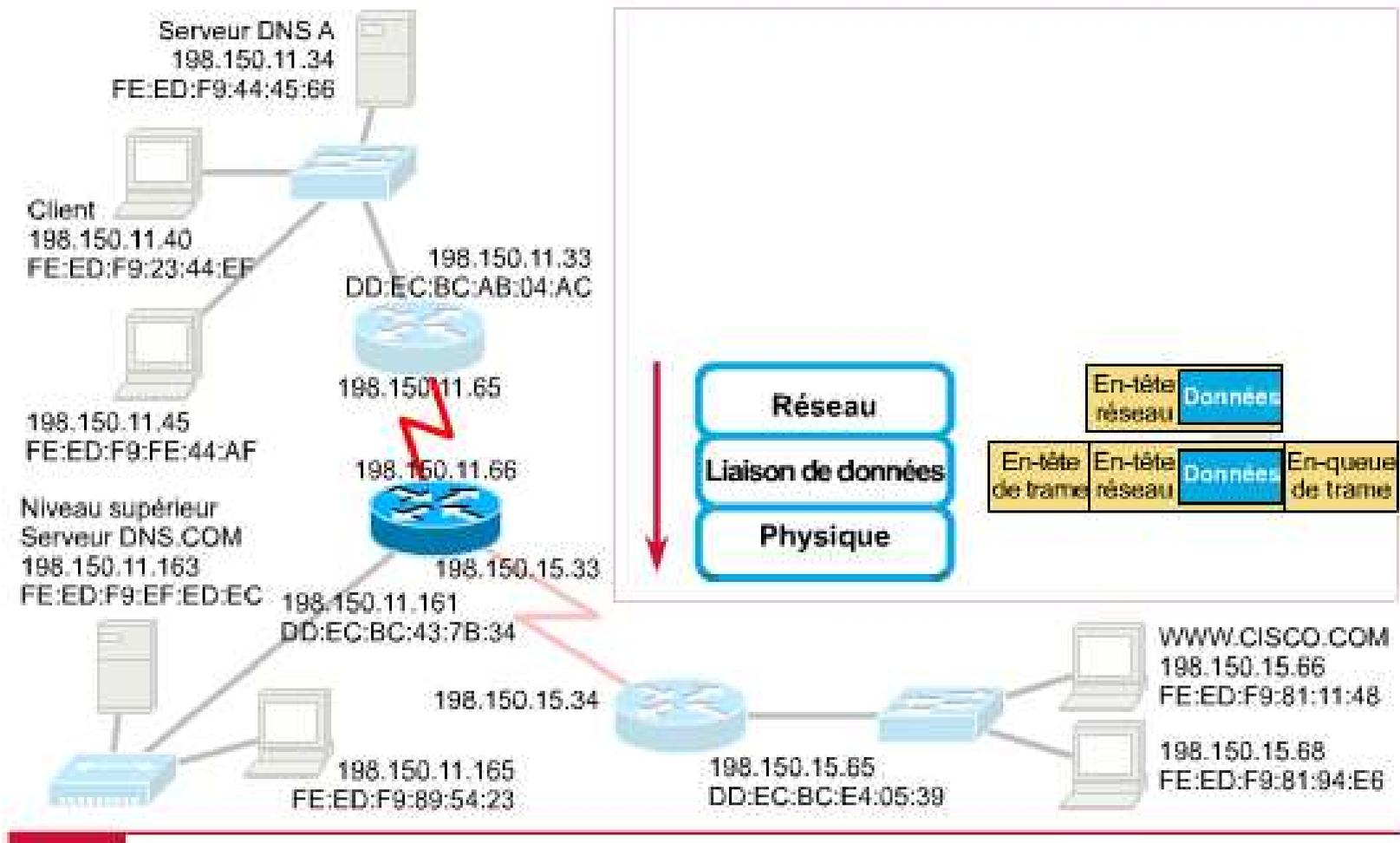
Exemple : DNS – Etape 6

- Le routeur retire la trame Ethernet et recherche l'adresse de sous-réseau dans sa table de routage.
- Pour accéder au sous-réseau **198.150.11.160**, il doit utiliser l'interface **198.150.11.65**



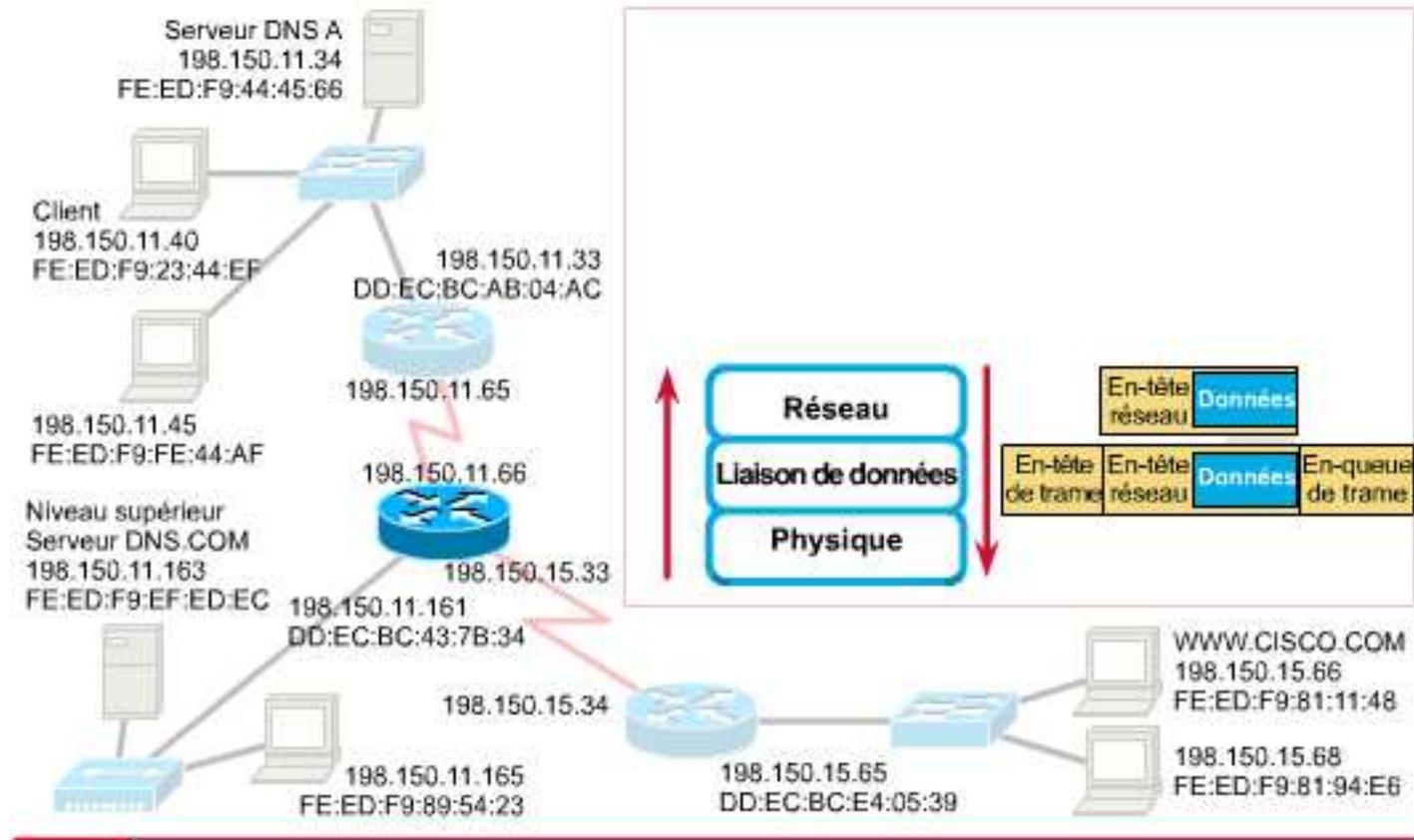
Exemple : DNS – Etape 7

La requête est encapsulée puis envoyée au routeur suivant par liaison série



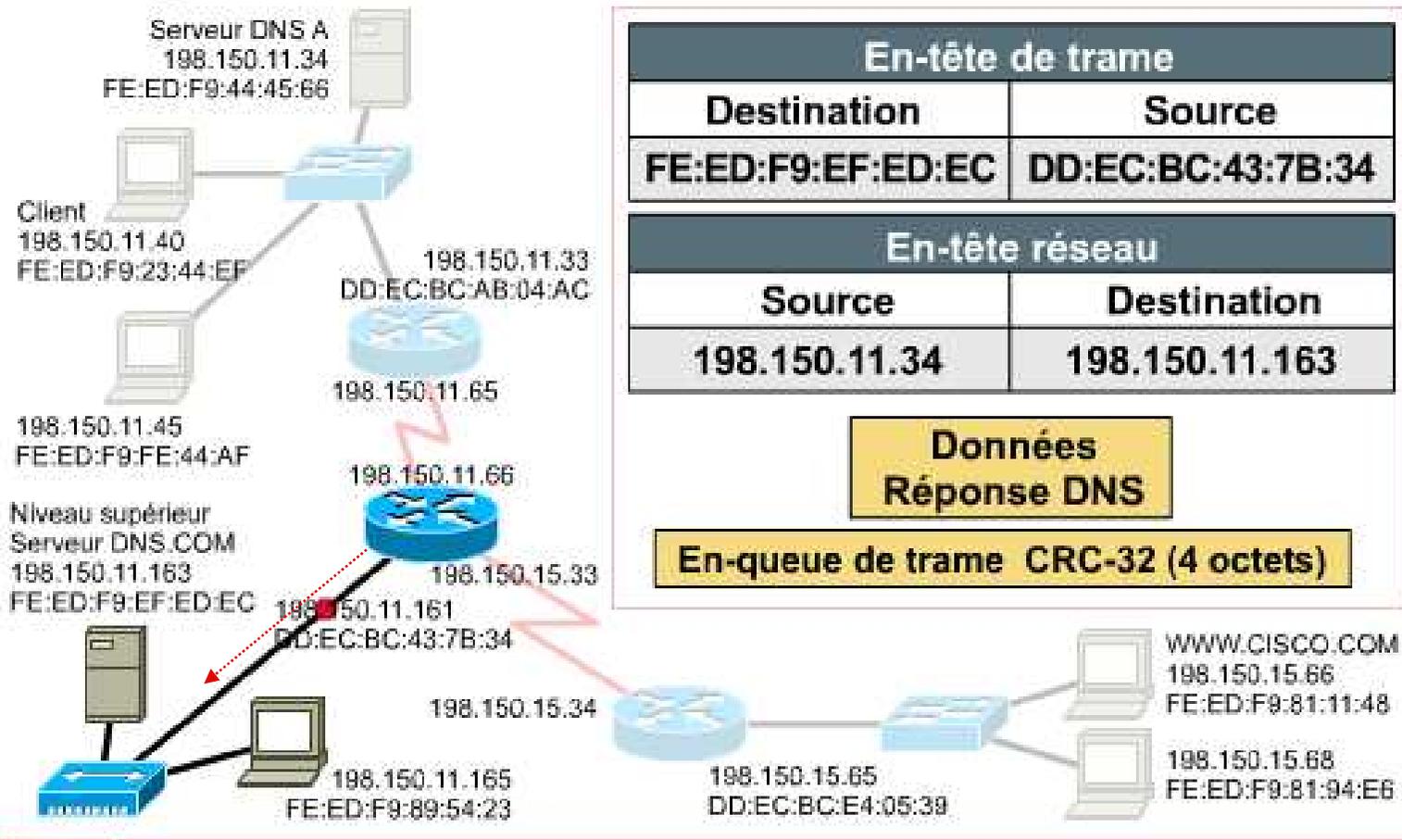
Exemple : DNS – Etape 8

- Le routeur suivant retire l'encapsulation série et recherche le sous-réseau IP **198.150.11.160** dans sa table de routage.
- Il détermine que le sous-réseau est directement connecté à l'interface **198.150.11.161**.
- La requête DNS est encapsulée dans une trame Ethernet et envoyée au sous-réseau **198.150.11.160**



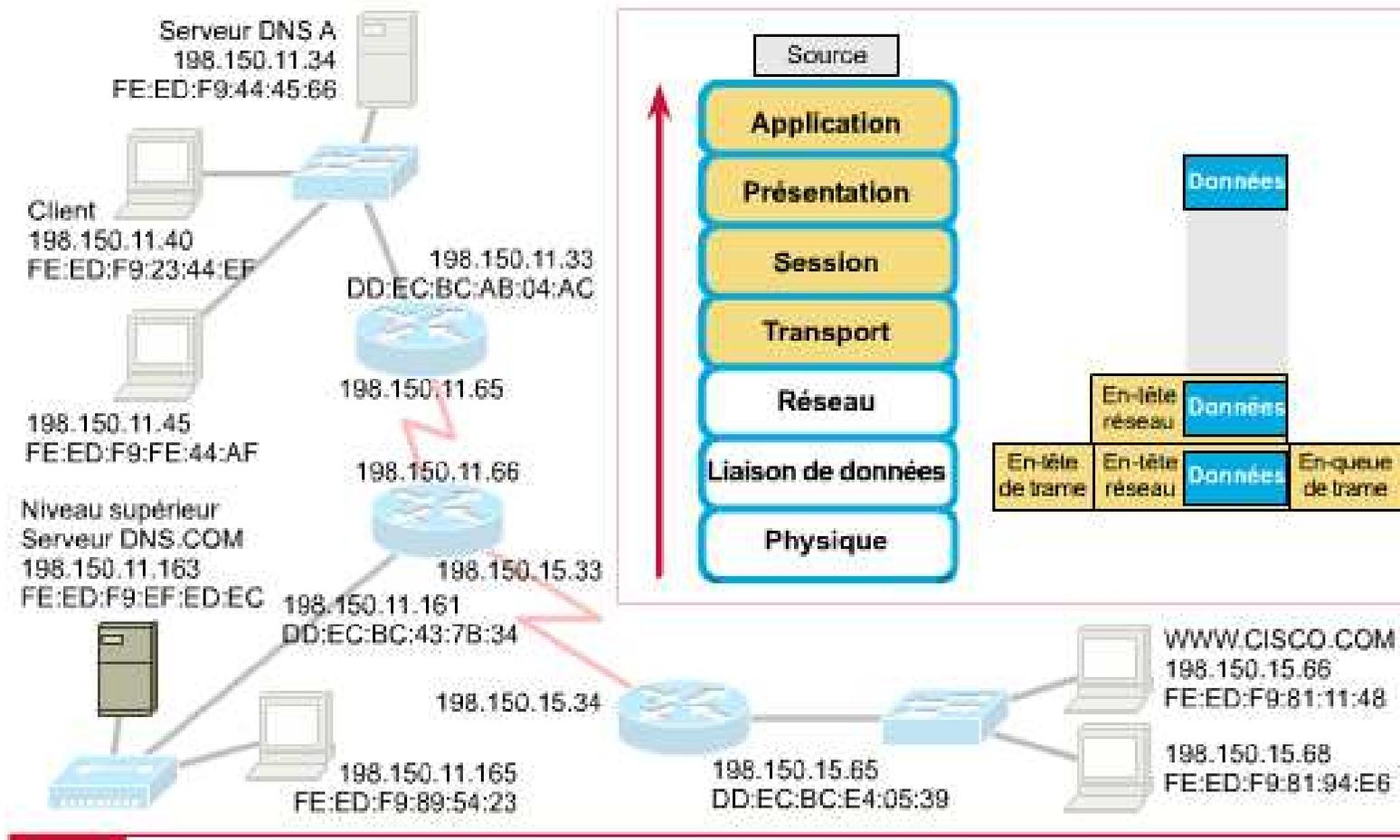
Exemple : DNS – Etape 9

- Les trames de données sont alors transmises sur le segment Ethernet.
- Toutes les stations prennent le paquet et vérifient si ce dernier leur est destiné.
- A l'exception du serveur DNS, toutes les unités, y compris le routeur, rejettent les paquets



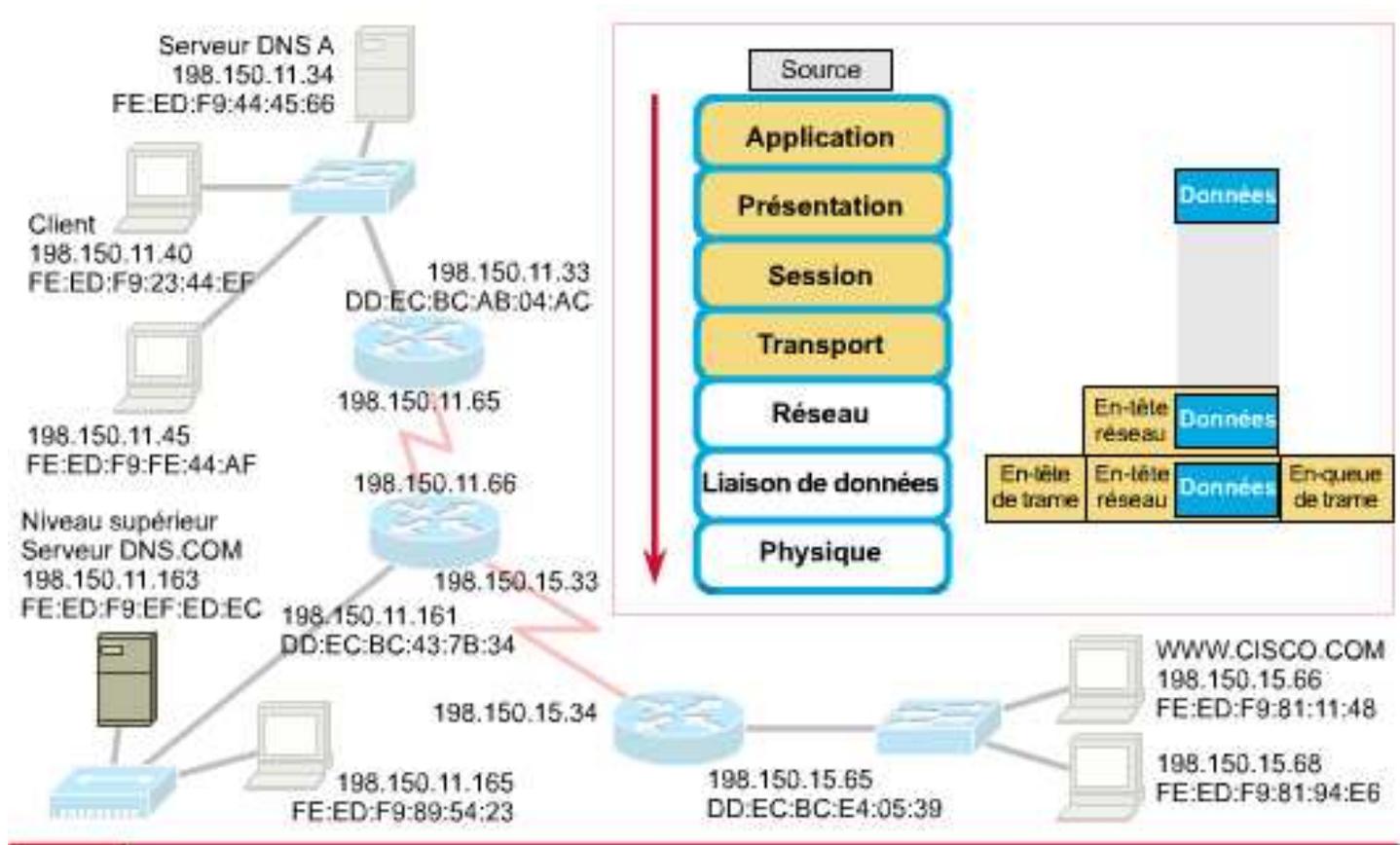
Exemple : DNS – Etape 10

- Le serveur DNS retire la trame de données et vérifie si l'adresse IP est la sienne.
- Il retire ensuite l'en-tête IP et traite les données



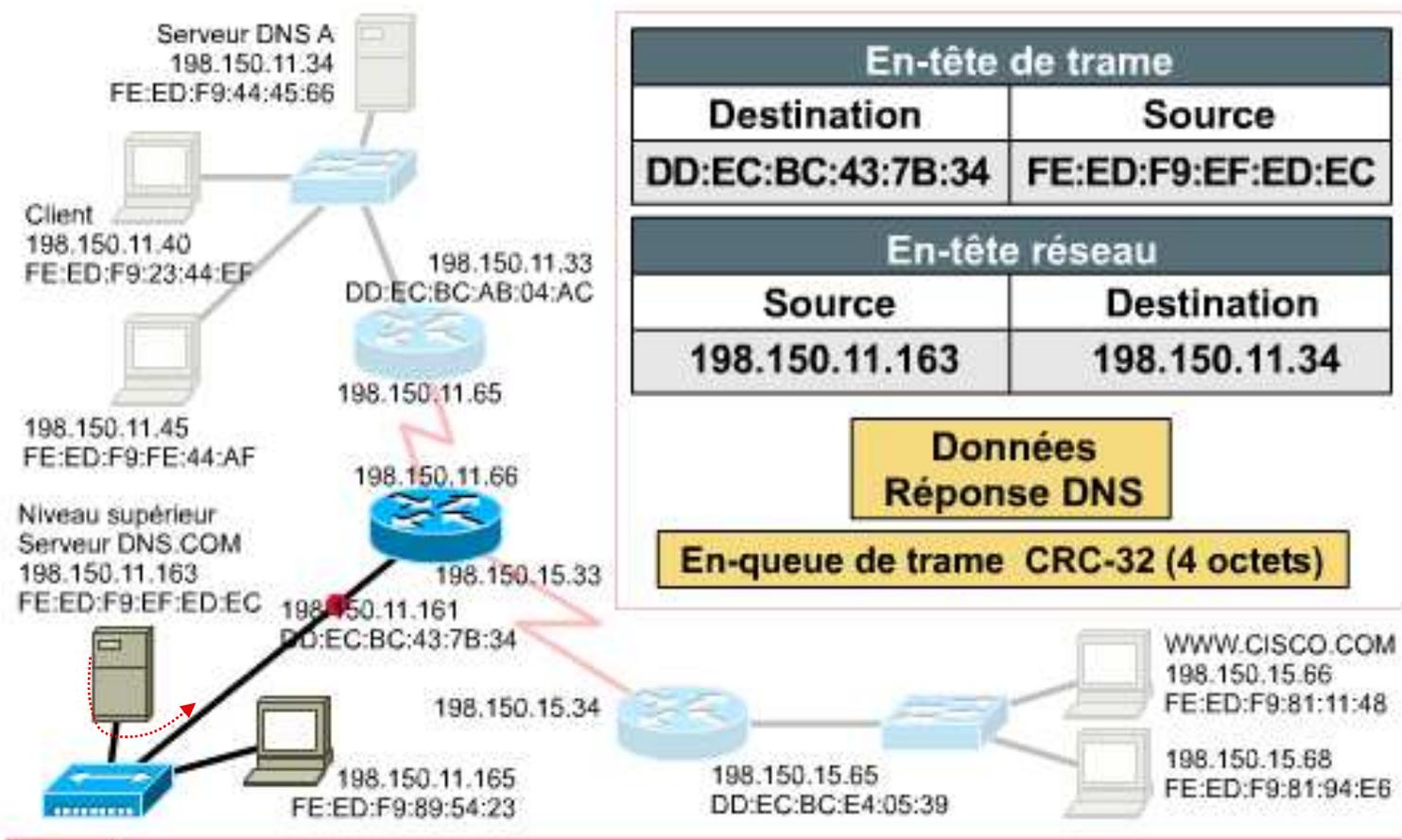
Exemple : DNS – Etape 11

- Le serveur DNS recherche le nom DNS dans la table DNS et trouve l'entrée.
 - Elle correspond à l'adresse IP **198.150.11.66**
- Le serveur **DNS.COM** prépare la réponse DNS à l'aide de la pile de protocole de communication



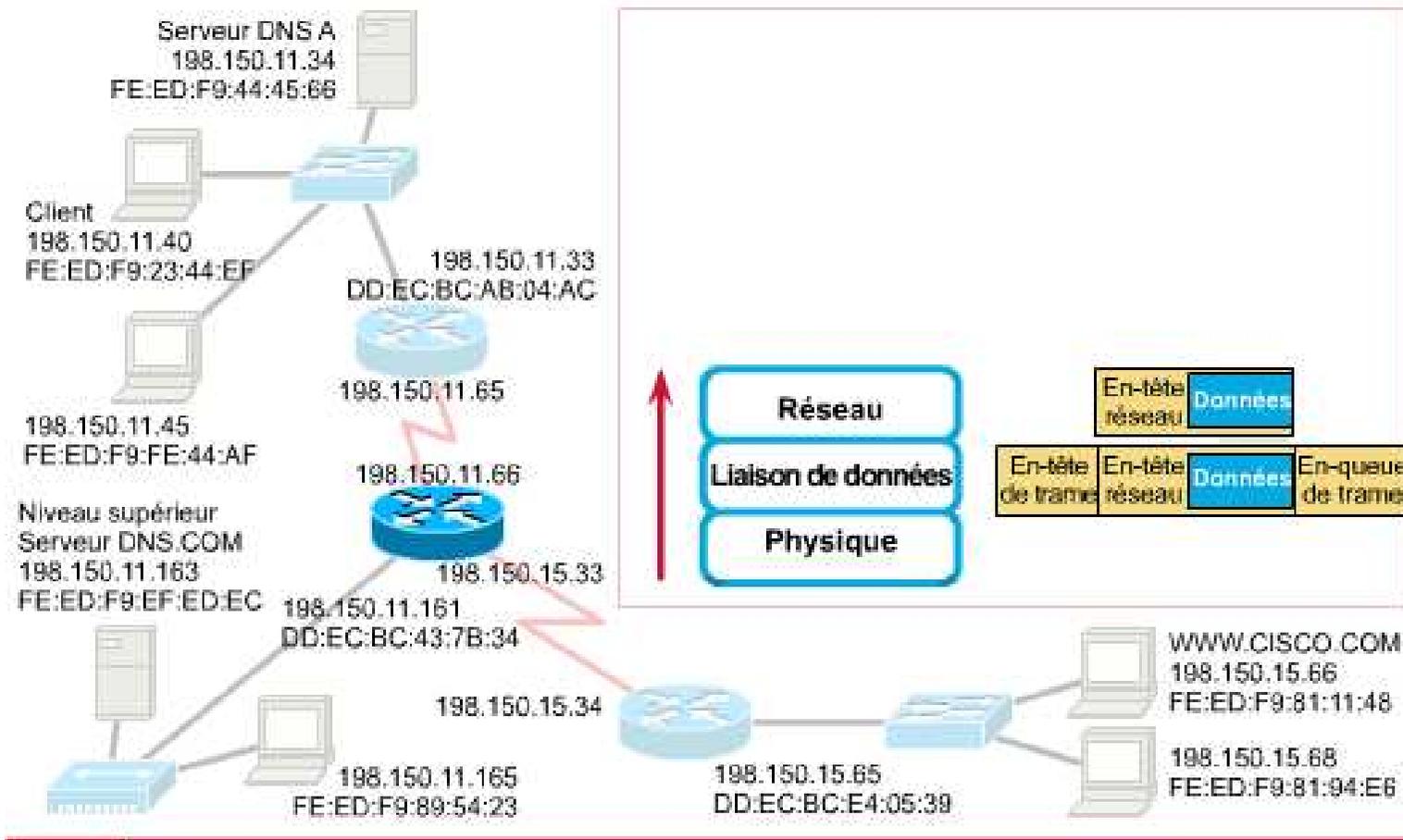
Exemple : DNS – Etape 12

- Les trames de données sont alors transmises sur le segment Ethernet
 - Toutes les stations prennent le paquet et vérifient si ce dernier leur est destiné
- A l'exception du routeur, toutes les stations rejettent les paquets



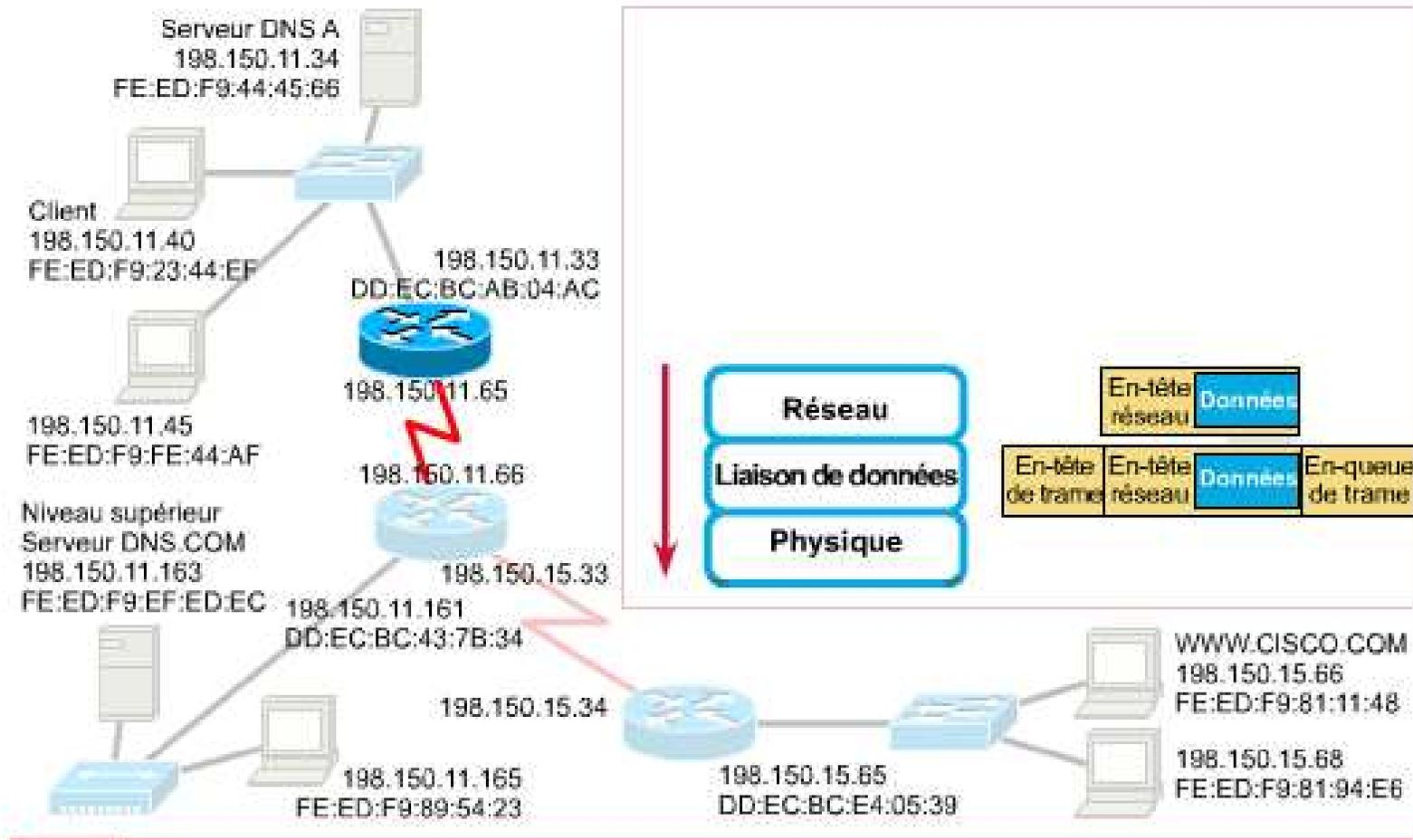
Exemple : DNS – Etape 13

- Le routeur retire la trame Ethernet et recherche l'adresse de sous-réseau dans sa table de routage.
- Pour accéder au sous-réseau **198.150.11.32**, il doit utiliser l'interface **198.150.11.66**



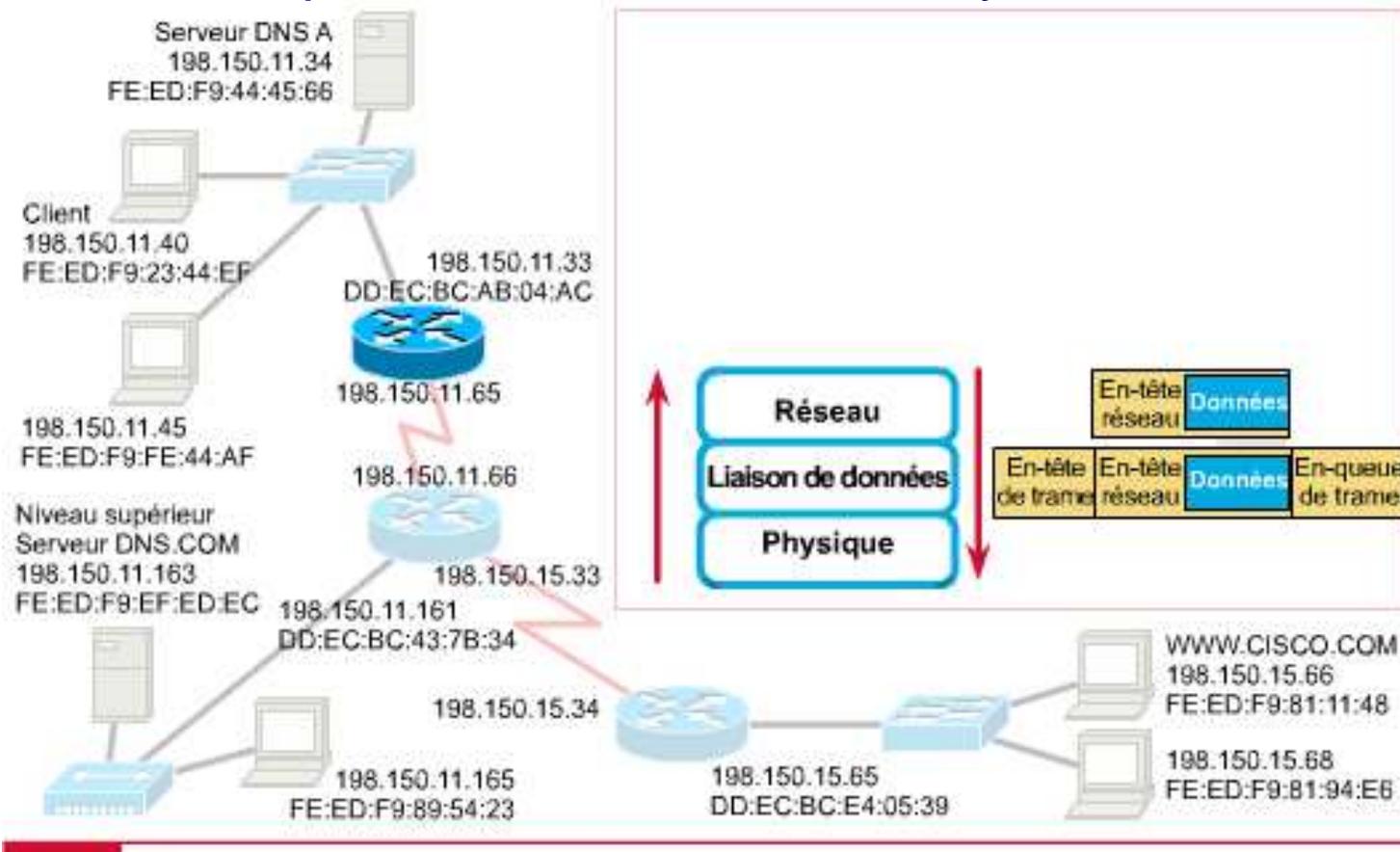
Exemple : DNS – Etape 14

La demande est encapsulée puis envoyée au routeur suivant par liaison série



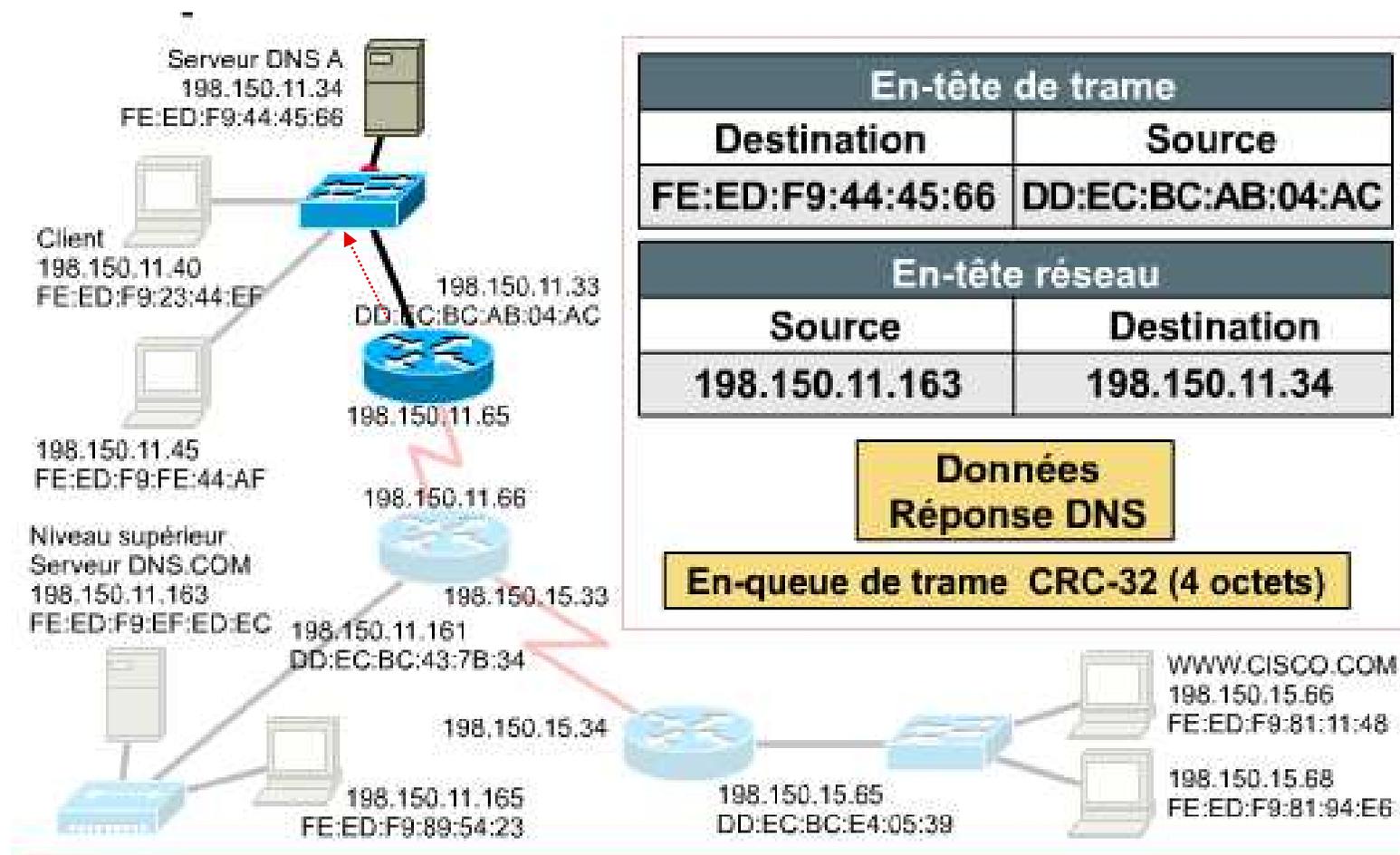
Exemple : DNS – Etape 15

- Le routeur suivant retire l'encapsulation série et recherche le sous-réseau IP **198.150.11.32** dans sa table de routage.
- Il détermine que le sous-réseau est directement connecté à l'interface **198.150.11.33**.
- La réponse DNS est encapsulée dans une trame Ethernet et envoyée au sous-réseau **198.150.11.32**



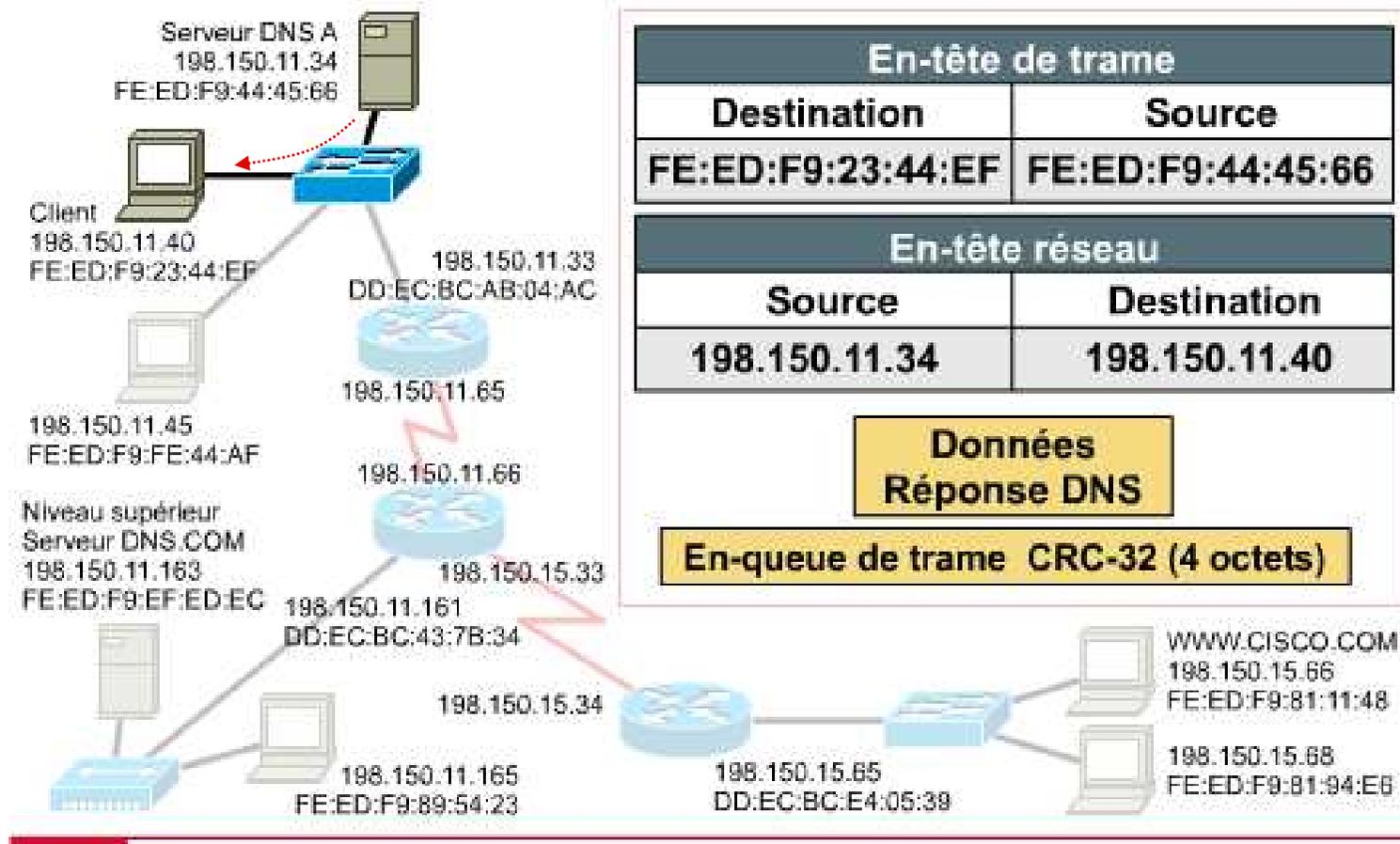
Exemple : DNS – Etape 16

- La réponse est envoyée au serveur **DNS A** afin que ce dernier puisse mettre à jour ses tables pour une utilisation ultérieure.



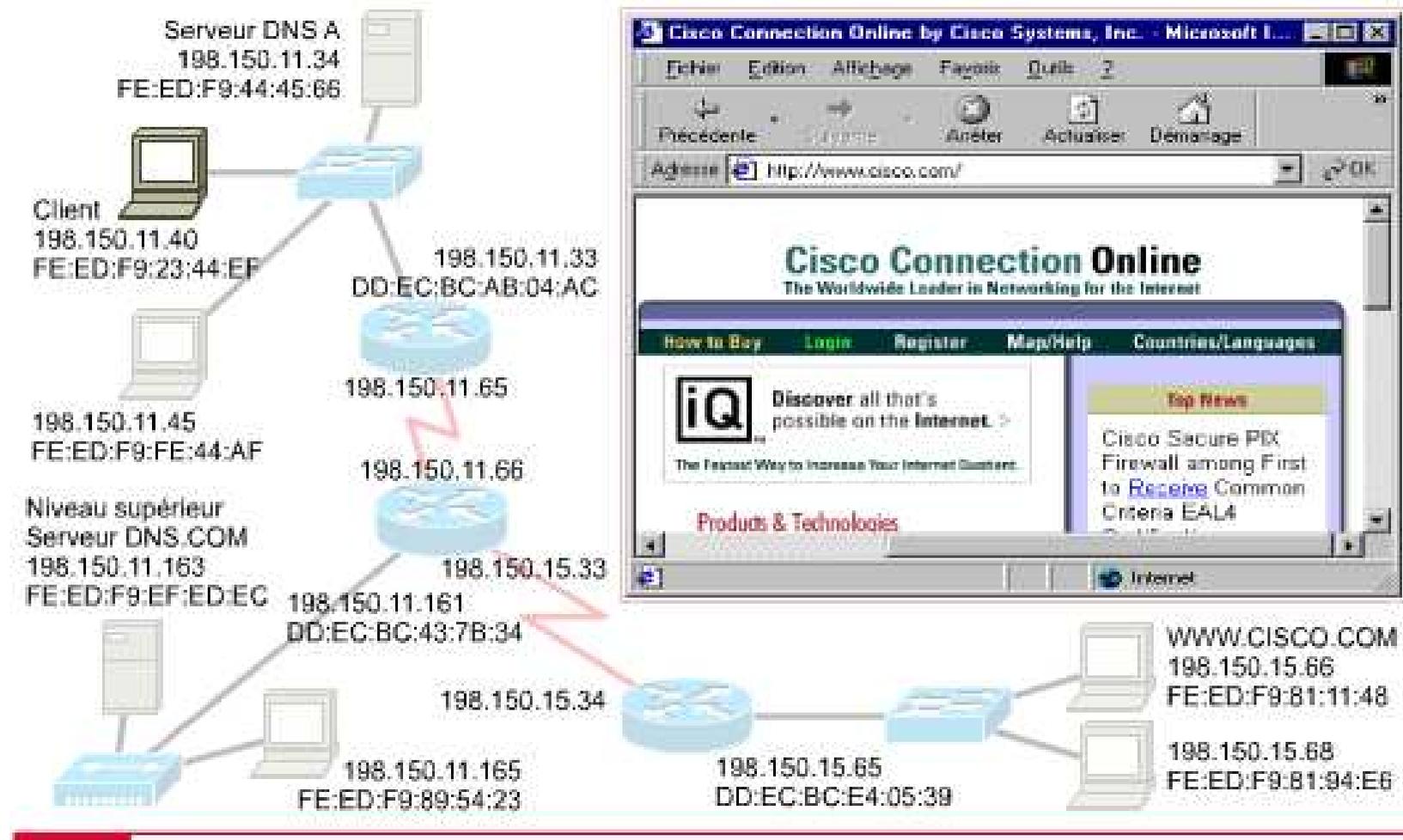
Exemple : DNS – Etape 17

- Le serveur DNS A est maintenant en mesure d'envoyer la réponse DNS au client demandeur initial.



Exemple : DNS – Etape 18

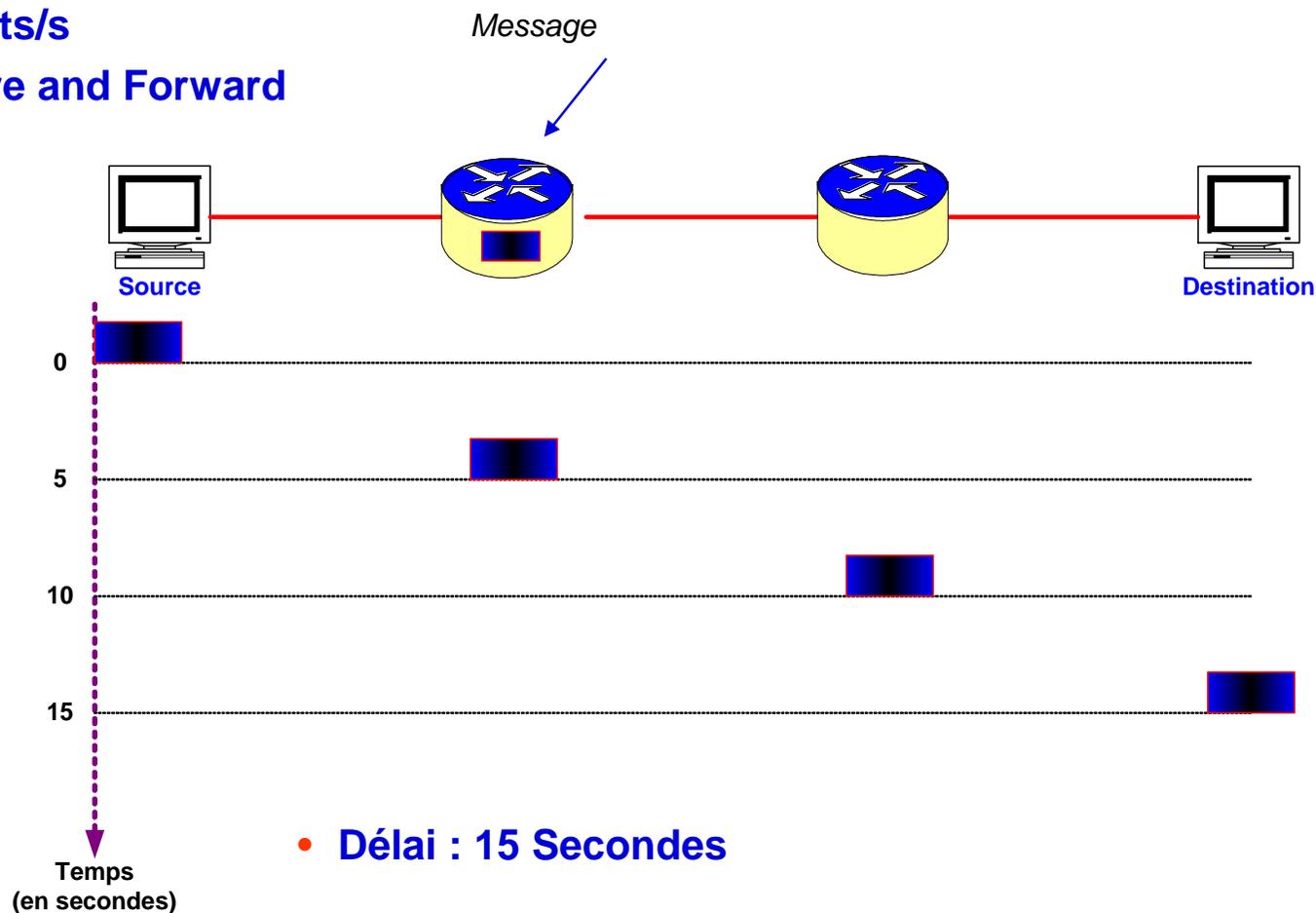
- Le client dispose maintenant de l'adresse IP pour www.cisco.com/fr/



SOLUTIONS

Tester votre compréhension

- Message de 7.5×10^6 bits
- Deux Commutateurs
- Trois liens de 1.5 Mbits/s
- Transmission en Store and Forward

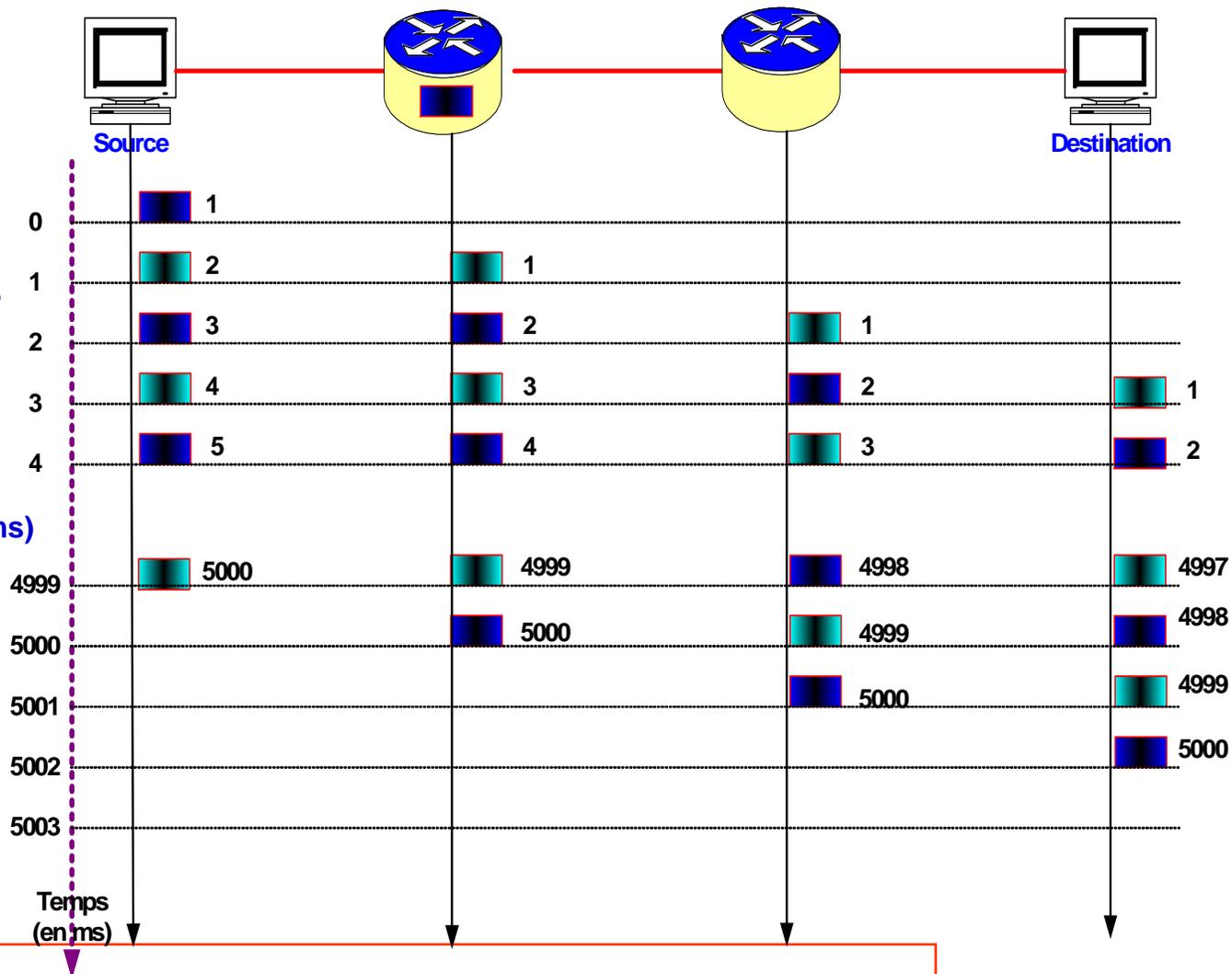


[back](#)

***N.B :** pendant que le message transite sur un des trois liens, il ne se passe rien d'autre sur les autres liens*

Tester votre compréhension

- Message de 7.5×10^6 bits
- **Segmenté** en 5000 paquets de 1500 bits
- Deux Commutateurs
- Trois liens de 1.5 Mbits/s



[back](#)

- Délai : 5.002 secondes
- Réduction d'un facteur : 3
- Commutation de paquet : transmission en parallèle des paquets sur les liens

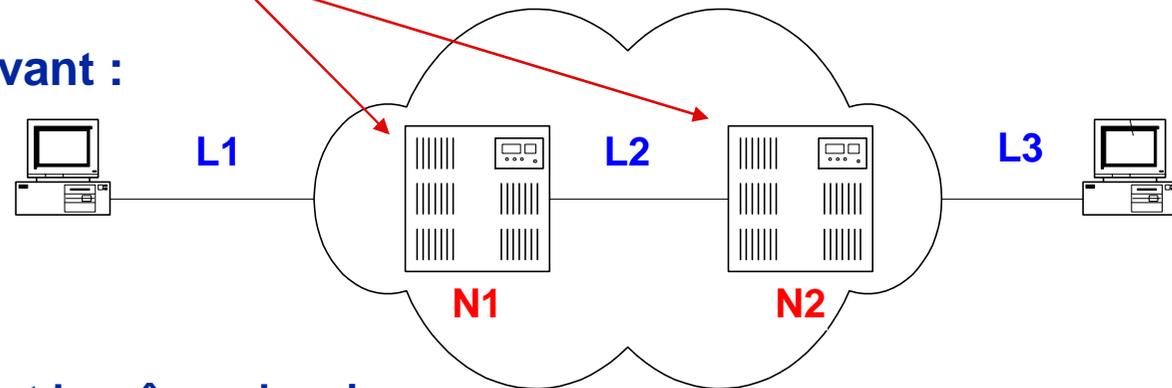
http://media.pearsoncmg.com/aw/aw_kurose_network_2/applets/message/messagesegmentation.html

ED

Comparaison entre commutation de circuit et commutation de paquets

Store and forward (stocker, traiter et retransmettre)

On considère le réseau suivant :



Hypothèses :

- tous les paquets empruntent le même chemin
- temps de transfert sur le support est nul
- le réseau possède deux nœuds
- le message est découpé en p paquets

Soit :

L : Longueur du message

D : Débit de la ligne

T_c : Le temps de transit de message

1. Calculer T_c

[solution](#)

2. Dans la commutation par paquet,

2.1 : calculer le temps d'émission d'un paquet sur le lien (t_p)

solution

2.2 : tracer le chronogramme de transit du message sur le réseau dans le cas ou $p = 4$

solution

2-3 : calculer le temps total de transit pour une fragmentation en p paquets avec N nœuds sur le réseau (T_p)

solution

2-4 : Soit H la longueur des informations de service (en-tête), calculer la nouvelle expression de T_p

solution

2 -5 : Montrer graphiquement la variation du temps de transit en fonction du nombre de paquets et de la taille de l'en-tête

solution

Commentaires ?

Solution

Commutation de circuit :

- Le message est transmis intégralement directement sur le lien
- Hypothèse : Temps de transfert de paquet est nul sur le support

1 - Le temps de transit du message sur le réseau correspond au temps d'émission sur le lien, soit :

$$T_c = \frac{L}{D}$$

L : Longueur du message
D : débit de la ligne

[back](#)

2 - Commutation de paquet :

Le message est découpé en p paquets émis successivement sur le lien

Chaque nœud du réseau retransmet immédiatement le paquet dès sa réception

(store et forward)

2 - 1 : Si le message est découpé en p paquets, le temps d'émission d'un paquet sur le lien est :

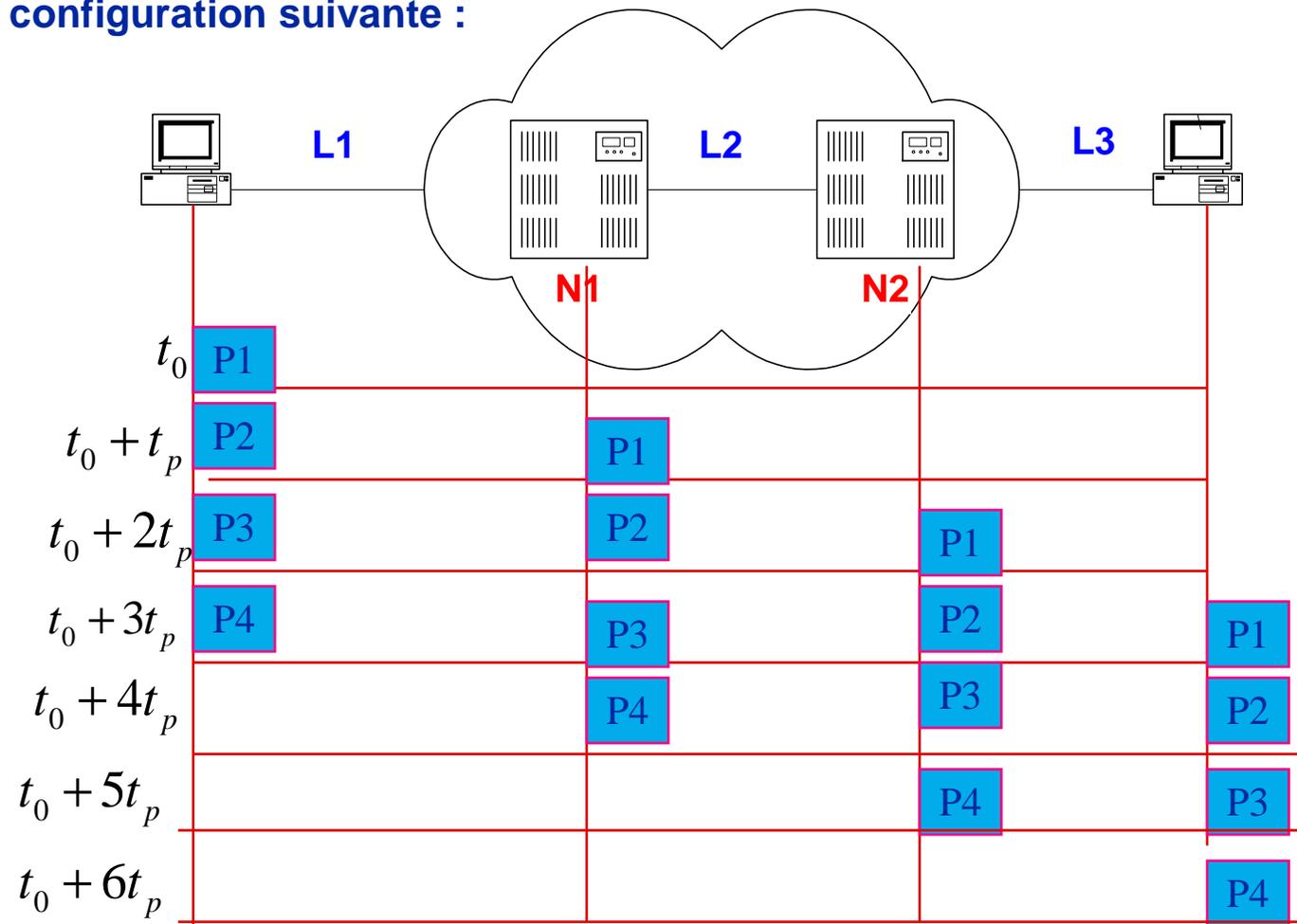
$$t_p = \frac{T_C}{p} = \frac{L}{pD}$$

[back](#)

2-2 : Commutation de paquet :

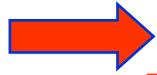
Pour déterminer le temps au bout duquel un message de **P** paquets arrive à sa destination après avoir traversé **N** nœuds

on suppose la configuration suivante :



[back](#)

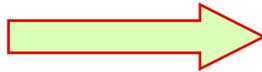
2-3 : temps total de transit



$$T_P = 4t_p + 2t_p$$

$$\left\{ \begin{array}{l} 4t_p = T_C \\ 2t_p = N.t_p = N \frac{T_C}{p} \end{array} \right.$$

N : le nombre de nœuds



$$T_P = T_C (1 + N/p) = \frac{L}{D} \left(1 + \frac{N}{p}\right)$$

[back](#)

2 -4 : Nouvelle valeur du temps total de

Si **H** est la longueur des informations de service

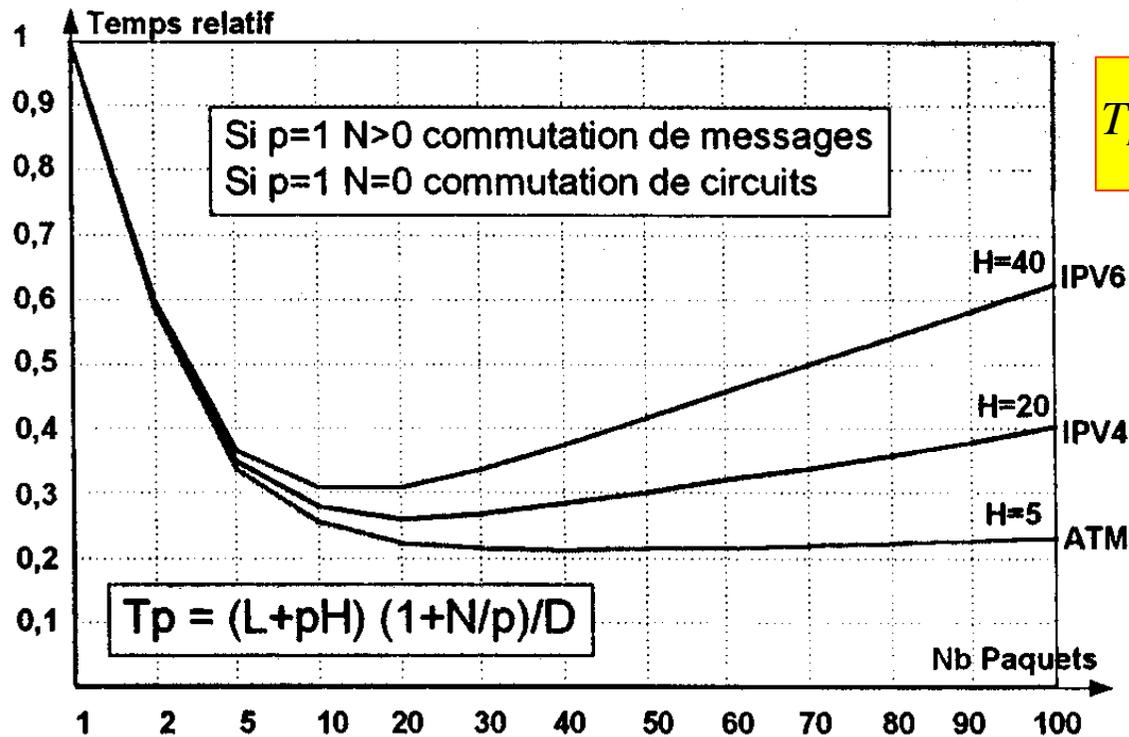
$$T_C = \frac{L + pH}{D}$$

$$T_P = T_C (1 + N/p) = \frac{(L + pH)}{D} \left(1 + \frac{N}{p}\right)$$

[back](#)

Temps relatif :

rapport entre le temps de transit effectif pour une fragmentation de p paquets et le temps de transit lorsque le message contient en seul paquet (commutation de messages)



$$T_p = \frac{(L + pH)}{D} \left(1 + \frac{N}{p}\right)$$

Evolution du temps de transit en fonction du nombre de paquets.

Exemple :

message de taille 1500 octets et un réseau de 5 nœuds

Discussion

$$T_p = \frac{(L + pH)}{D} \left(1 + \frac{N}{p}\right)$$

- Le temps de transit dans le réseau est d'autant plus faible que le facteur N l'est.
 - ➔ L'aspect performance conduit donc à minimiser le nombre de nœuds traversés et par voie de conséquence à **augmenter le maillage du réseau** (augmentation de la probabilité de trouver une route plus directe) ;
- L'influence de la taille de l'en-tête de service est non négligeable,
 - ➔ la figure compare les performances en fonction d'un en-tête
 - ATM (5 octets)**
 - IPV4 (20 octets)**
 - IPV6 (40 octets).**

Cette approche conduit à définir un rapport optimal entre la charge utile du bloc de données et les données de services (**ATM environ 10**) ;
- Les résultats ci-dessus ne tiennent pas compte du **temps de traitement dans les noeuds** ;
- Notons qu'en cas d'erreur, en commutation de messages, **le message est intégralement retransmis**, en commutation de paquets **seul le paquet erroné ou depuis le paquet erroné dans la fenêtre est retransmis**.

[back](#)

FIN

Couche Hôte-Réseau (d'accès au réseau - Host-to-network layer)

- ◆ N'est pas explicitement définie dans le modèle TCP/IP
 - **Interface d'accès au réseau** qui doit permettre à un ordinateur d'envoyer des paquets IP
 - **Indépendance** de la technologie d'un sous-réseau
- ◆ Des méthodes de transporter des paquets IP ont été définies pour de nombreuses technologies de réseau
 - IP sur Ethernet
 - IP sur des lignes séries
 - IP sur ATM
 - ...

[Retour](#)

