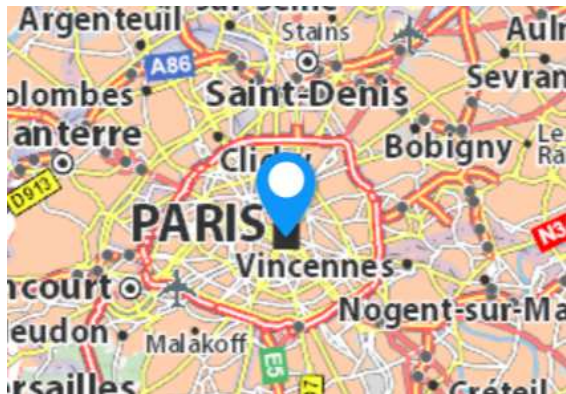


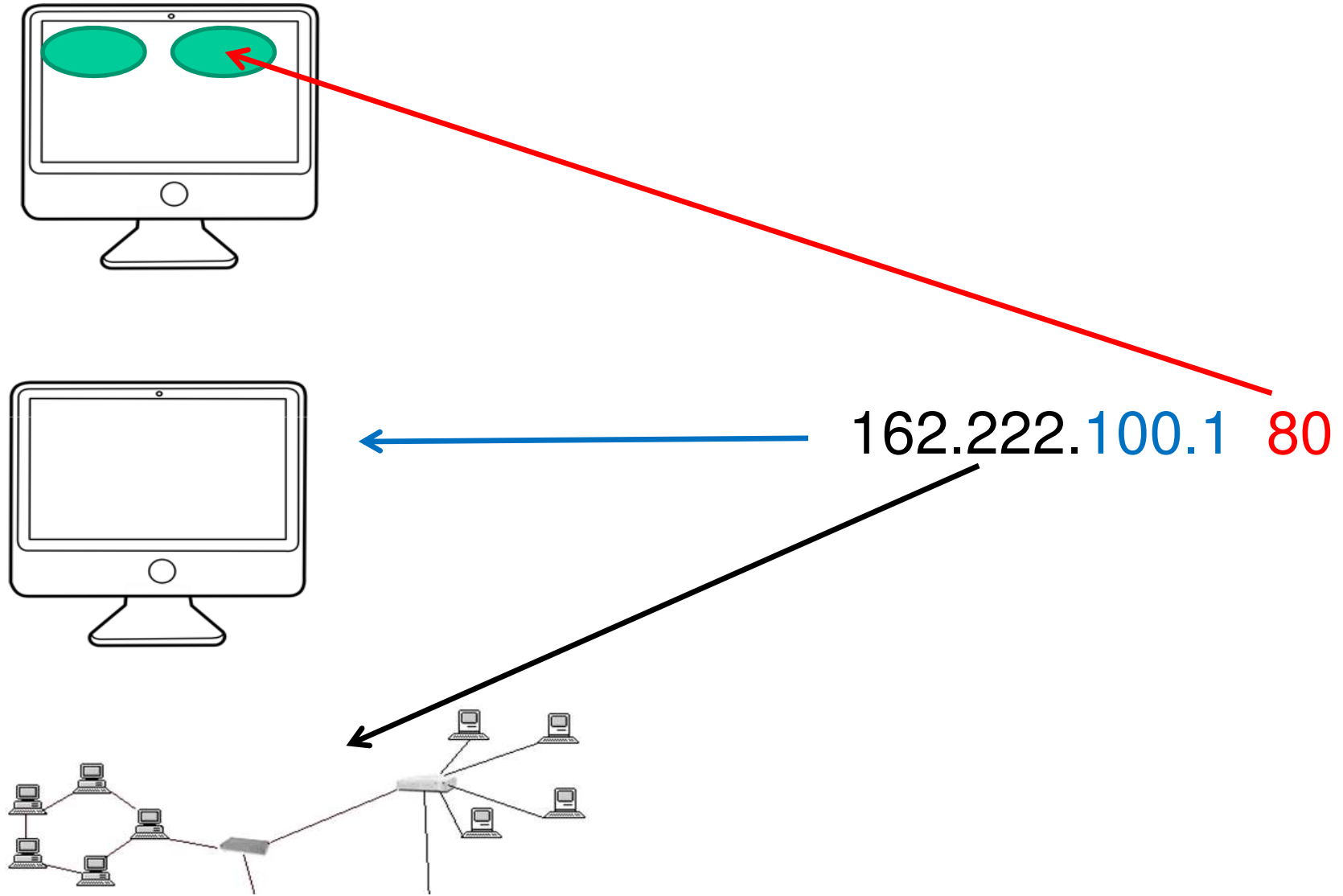
Réseaux Informatiques

Rappels

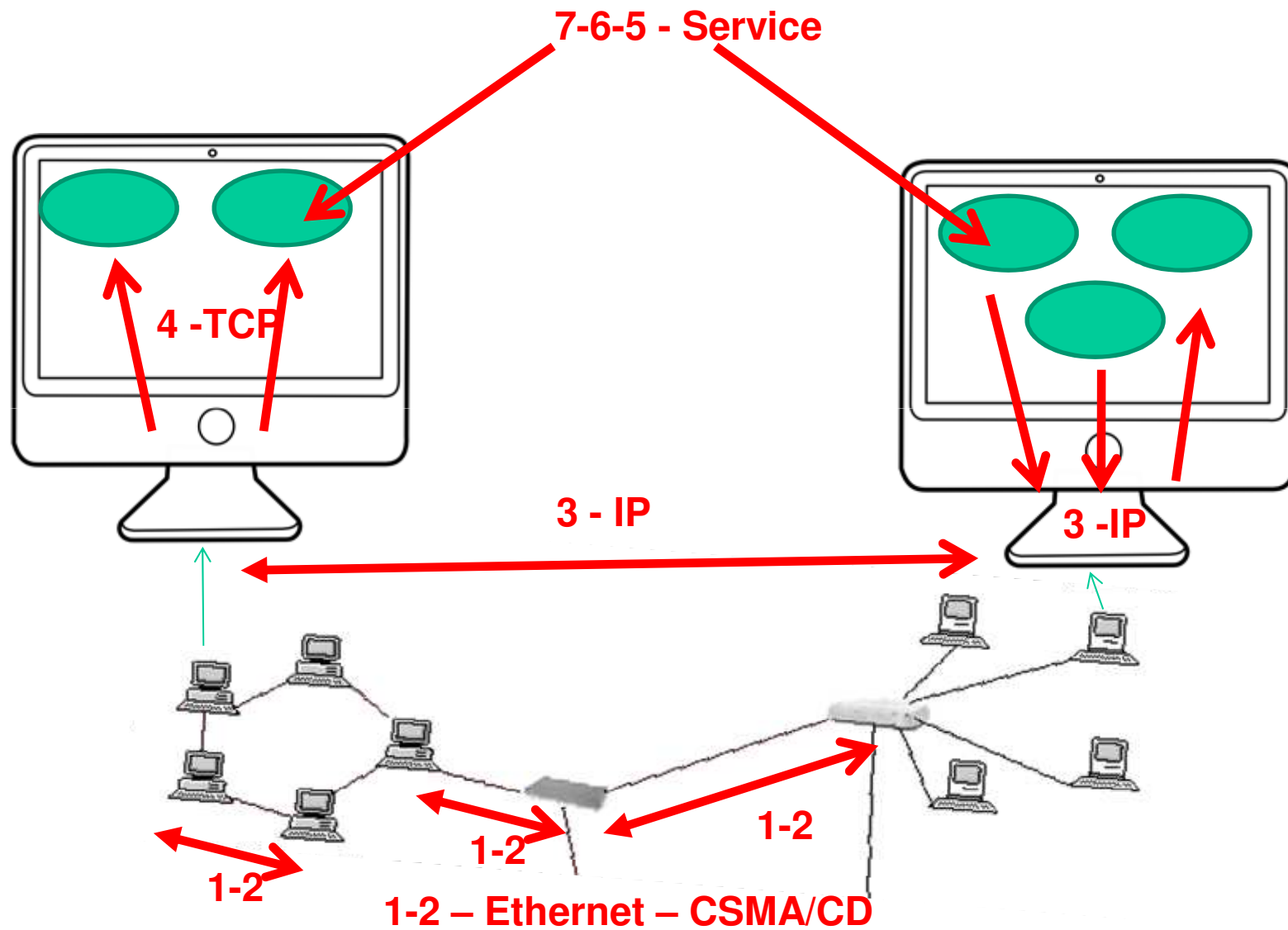
Rappels



Rappels

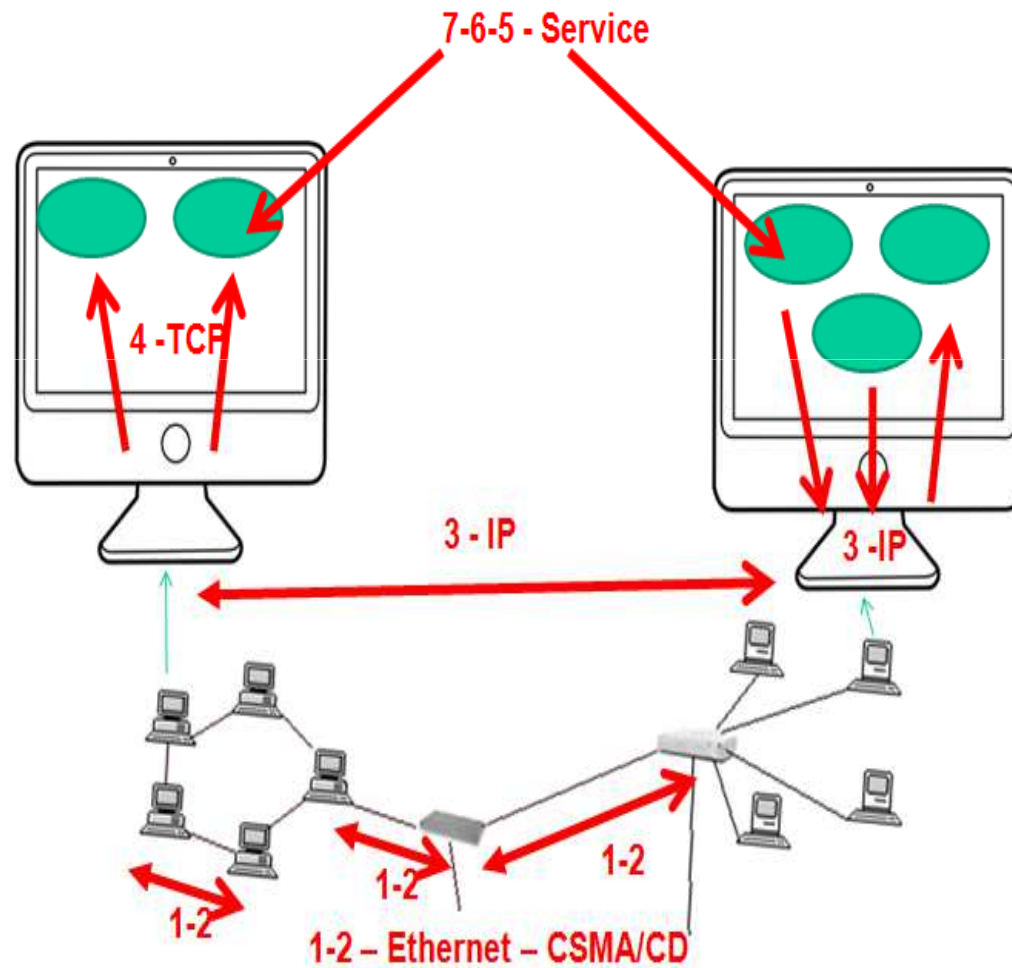


Rappels



Introduction générale

Le modèle OSI

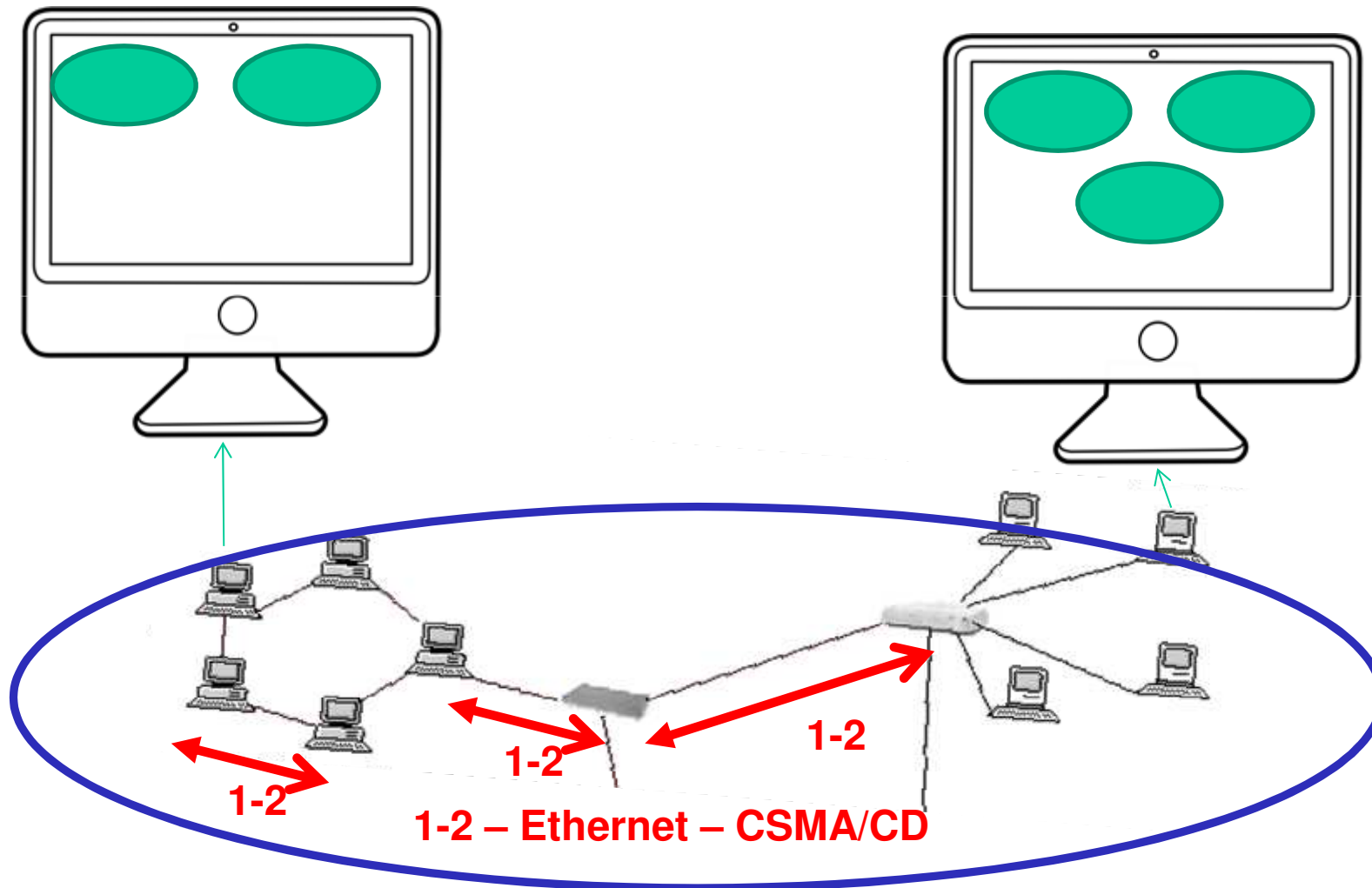


Type de problème	Couche	
Applications réseaux	7	Application
Accès aux ressources	6	Présentation
	5	Session
Echange via plusieurs réseaux	4	Transport
	3	Réseau
Echange entre 2 machines	2	Liaison
	1	Physique

Le protocole Ethernet

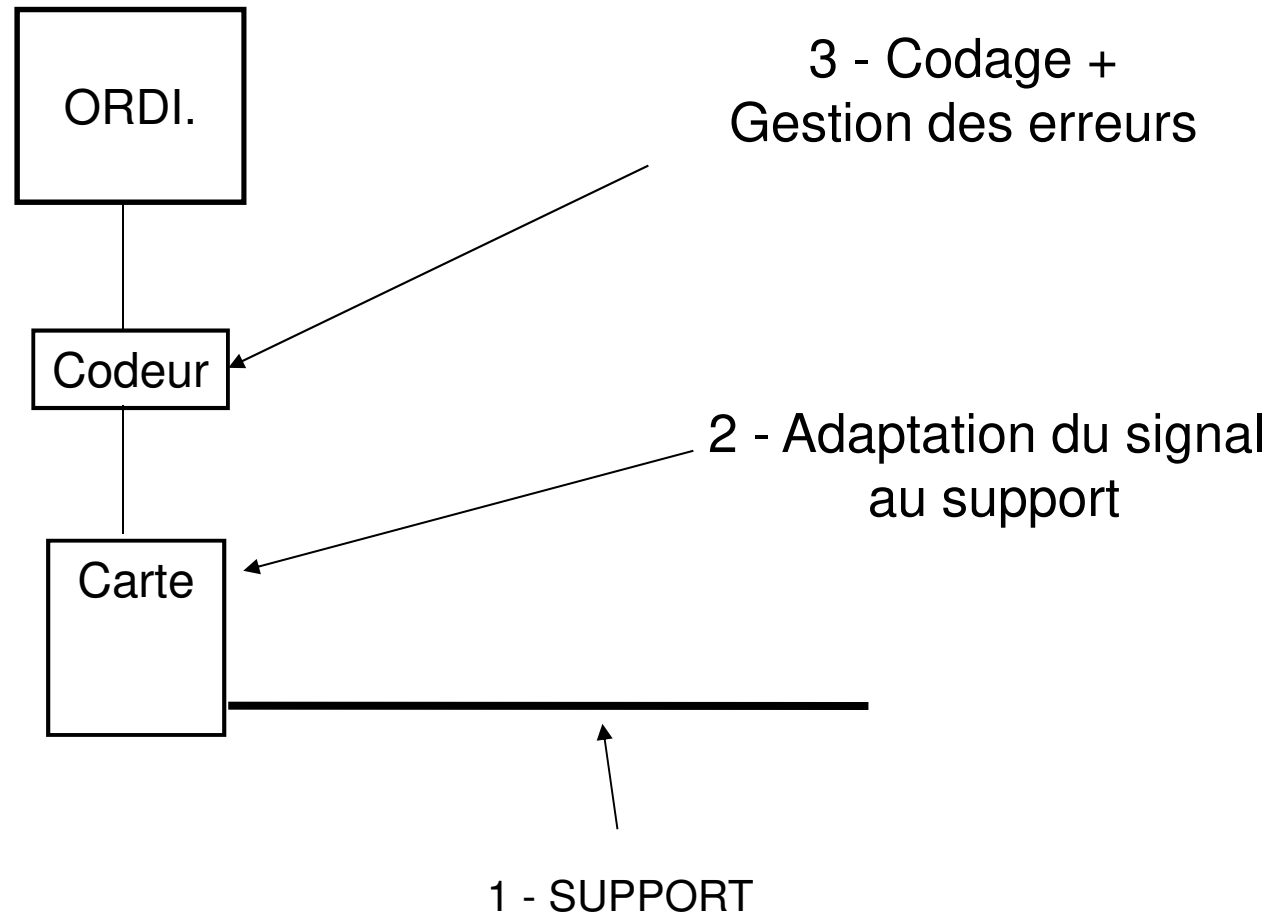
Couches 1 et 2

Couches 1 et 2



Le protocole Ethernet

E
T
H
E
R
N
E
T



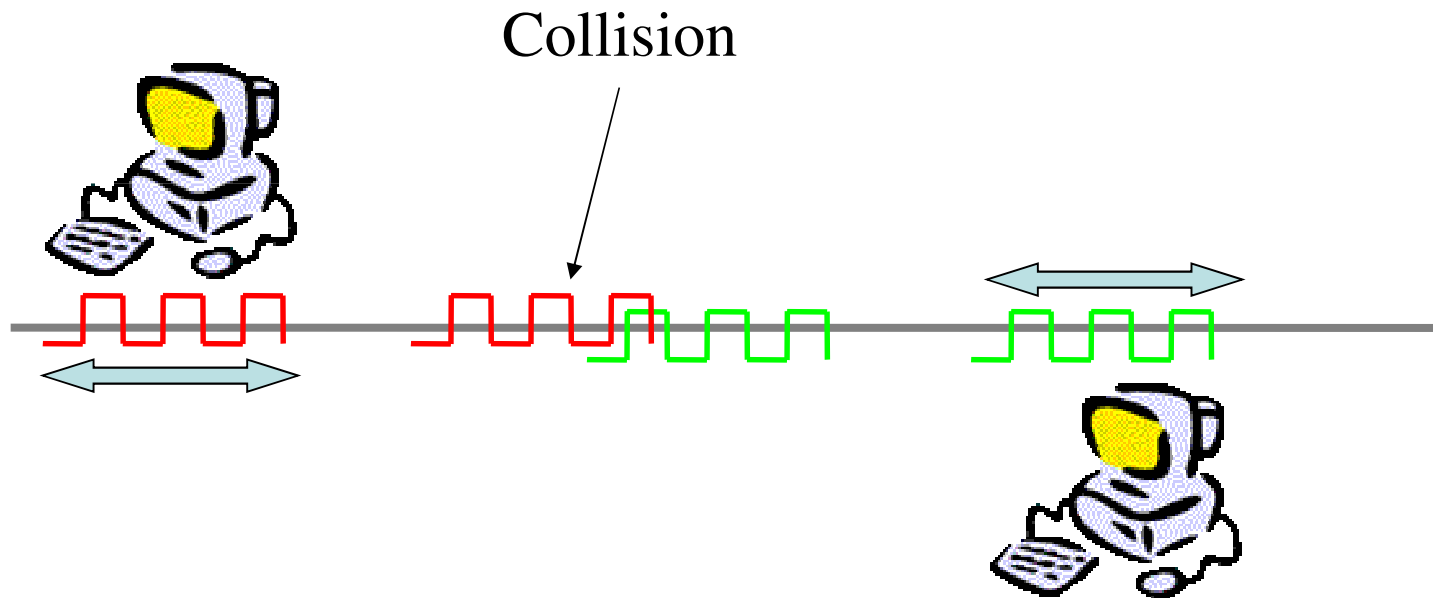
Le protocole Ethernet

802.1 Protocole sans accusé de réception demandé											
802.2 LLC (Liaison de données) Type 1 (sans connexion) Diffusion (Datagramme) Pas de lien logique établi Pas d'acquittement Type 2 (orienté connexion) Lien logique Acquittement Contrôle de flux et d'erreurs Type 3 (trame simple) Datagramme Acquittement											
<table><tr><td colspan="4">A C C E S A U M E D I A</td></tr><tr><td>802.3 CSMA/CD ETHERNET</td><td>802.4 BUS JETON</td><td>802.5 ANNEAU JETON</td><td>802.11 CSMA/CA</td></tr></table>				A C C E S A U M E D I A				802.3 CSMA/CD ETHERNET	802.4 BUS JETON	802.5 ANNEAU JETON	802.11 CSMA/CA
A C C E S A U M E D I A											
802.3 CSMA/CD ETHERNET	802.4 BUS JETON	802.5 ANNEAU JETON	802.11 CSMA/CA								

Le protocole Ethernet

IEEE 802.3

Transmission par inondation



Réseau en bus / technique d'accès : CSMA/CD

Le protocole Ethernet

Le protocole Ethernet a été conçu pour les réseaux locaux.

Objectif = acheminer des trames d'un poste à l'autre sur un câble unique.

1 – Adressage : 6 octets = Adresse MAC ou adresse physique

2 – Méthode d'accès : CSMA/CD (IEEE 802.x)

3 - Echange des données : Trame précédée d'un fanion

4 - Contrôle de la liaison : Rien de prévu

5 - Contrôle des erreurs : Ignore les trames en cas d'erreur

Le protocole Ethernet

Préambule	<i>Adresse destination</i>	<i>Adresse source</i>	<i>Type</i>	<i>Données + Bourrage</i>	<i>CRC</i>
8 octets	<i>6 octets</i>	<i>6 octets</i>	<i>2 octets</i>	<i>46 à 1500 octets</i>	<i>4 octets</i>

La trame Ethernet standard

**RFC 894 - Un Standard pour la Transmission des Datagrammes IP
sur les Réseaux Ethernet**

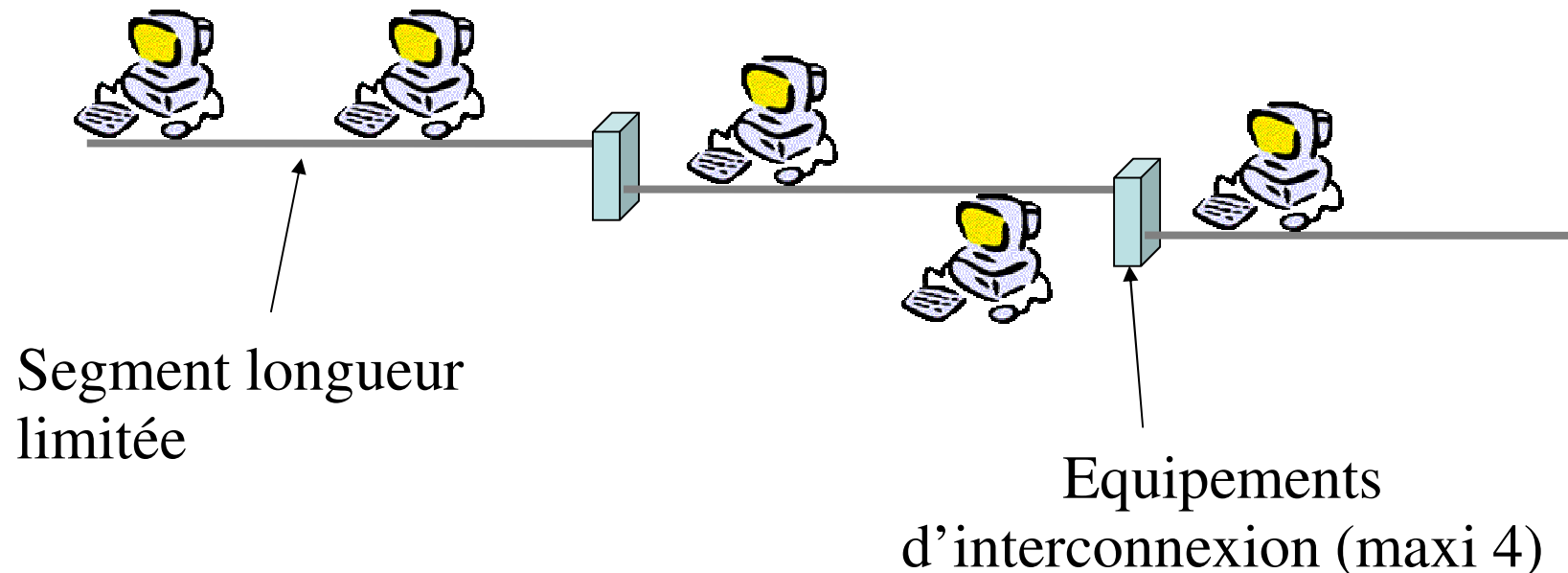
Le protocole Ethernet

Traitement d'une trame à la réception

- 1 – Vérification longueur trame , si < 64 octets alors rejet
- 2 – Vérification CRC , si erreur détecté alors rejet
- 3 – Vérification @destinataire , si @ différente alors rejet
- 4 – Extraction protocole, si protocole non actif alors rejet
- 5 – Transfert des données vers couche supérieure

Le protocole Ethernet

Pour allonger la distance de connexion on doit ajouter des équipements d'interconnexion, qui sont des amplificateurs de signaux électriques.



Le protocole Ethernet

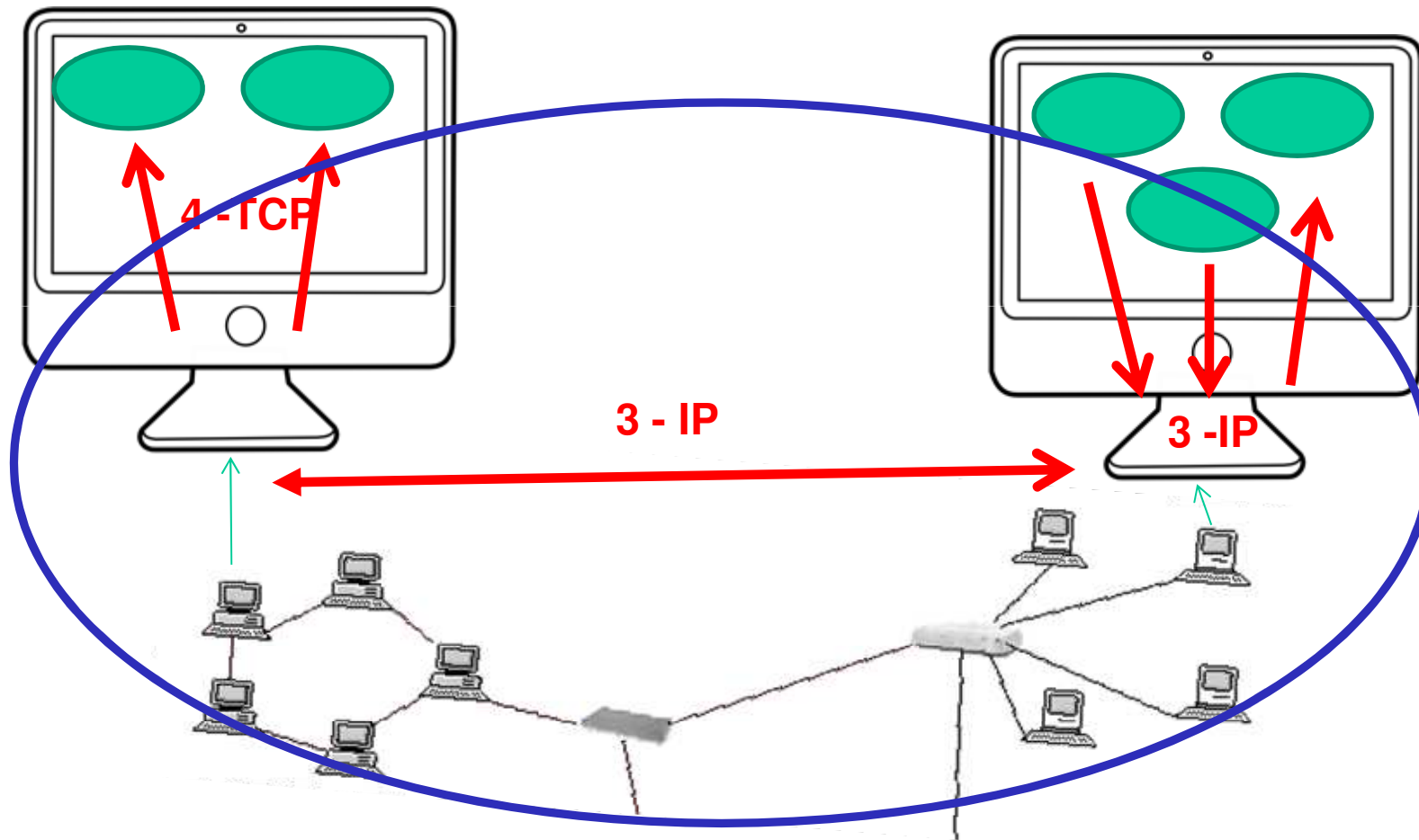
Les principaux équipements sont :

- Les répéteurs, permettant de régénérer un signal
- Les concentrateurs (**hubs**), permettant de connecter entre eux plusieurs hôtes
- Les commutateurs (**switchs**) permettant de relier divers éléments tout en segmentant le réseau
- ...
- Les routeurs, ne concernent pas le protocole Ethernet, ils permettent de relier des réseaux locaux différents.

Le protocole IP

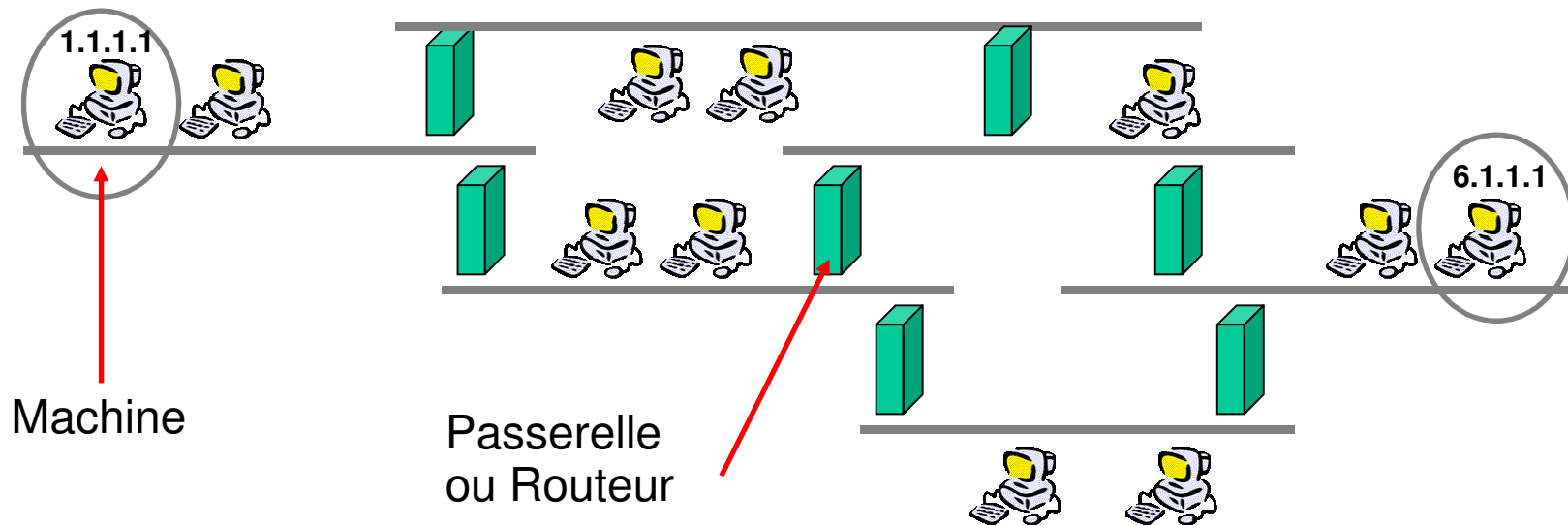
Couche 3

Couches 3 et 4



Protocole IP

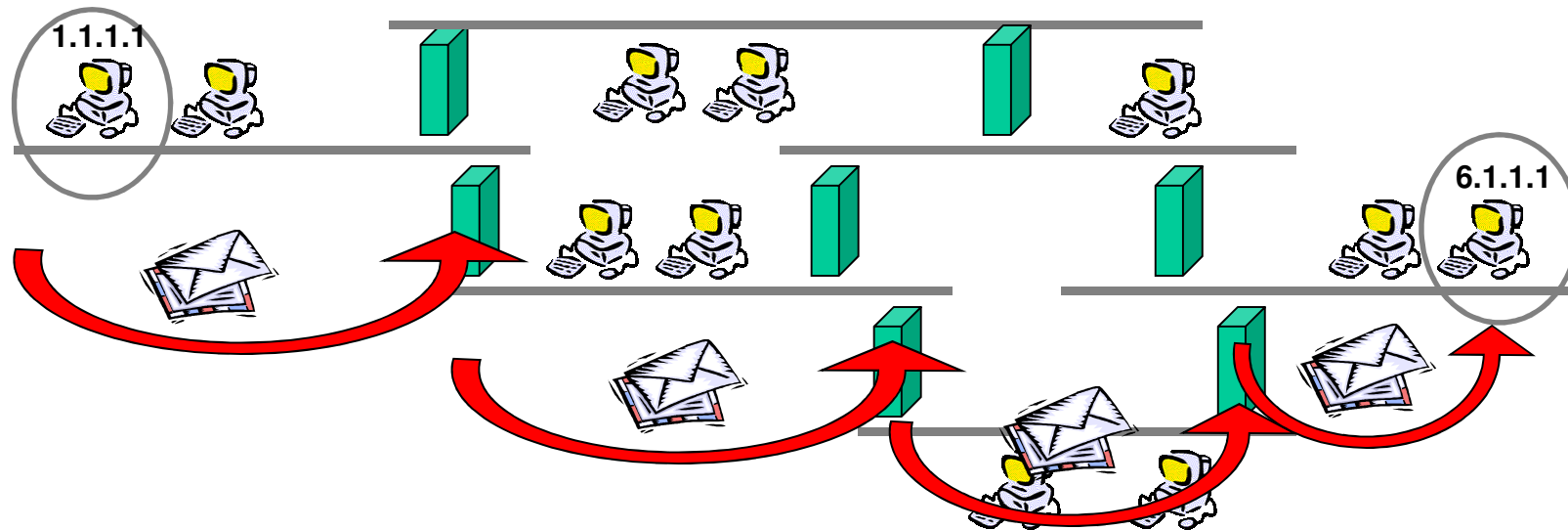
Routage IP V.4



Pour la transmission des paquets, machine et passerelle participent au routage (Un routeur possède deux ou plusieurs connexions réseaux tandis qu'une machine possède généralement qu'une seule connexion.)

Protocole IP

Routage IP V.4



Pour une machine deux possibilités :

- 1 – Elle voit le destinataire et lui transmet le paquet via Ethernet
- 2 – Elle ne voit pas le destinataire et transmet alors le paquet à la passerelle. Le paquet IP transite alors de passerelle en passerelle jusqu'à ce que l'une d'entre elle le délivre à son destinataire.

Protocole IP

Routage IP V.4

Pour déterminer où se trouve le destinataire, la technique consiste à masquer la partie adresse machine.

Le masque = valeur binaire pour extraire une partie de l'adresse de destination

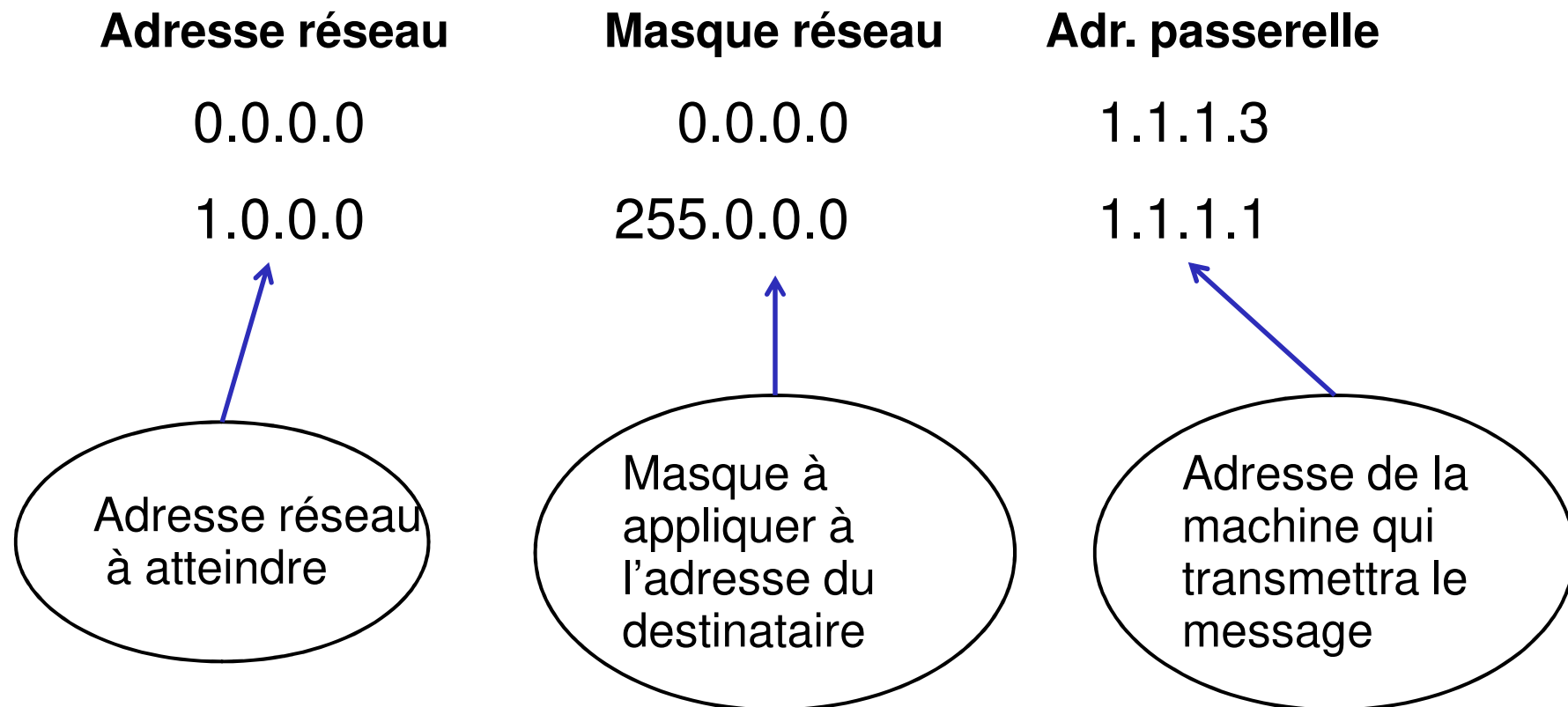
Exemple

	6.	1.	1.	1	= adresse destination
et	255.	0.	0.	0	= masque
	6 .	0 .	0 .	0	= résultat (adresse réseau)

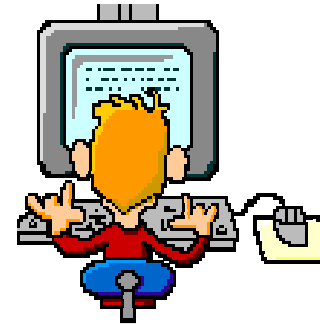
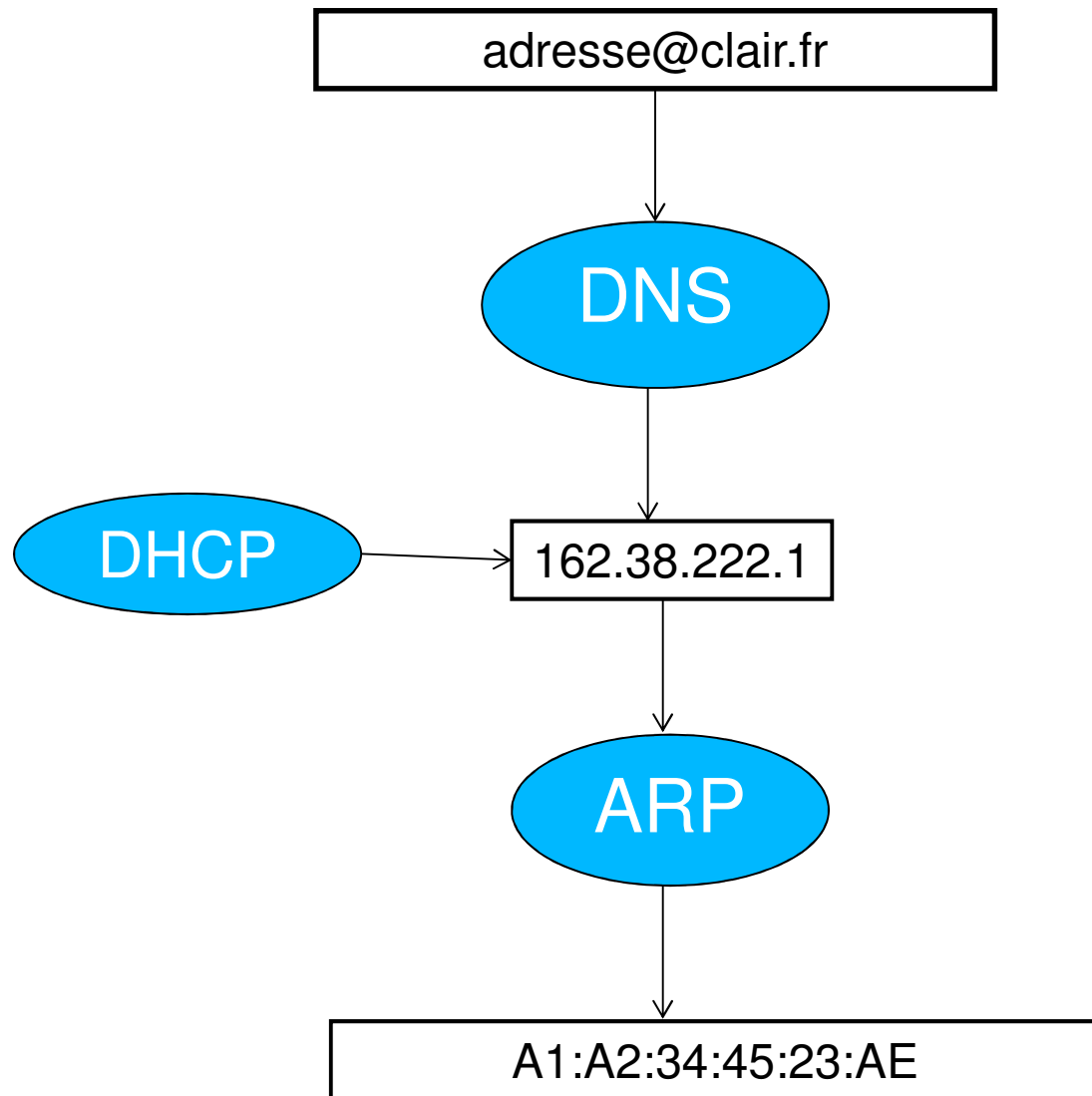
Protocole IP

Routage IP V.4

Pour prendre la décision, la machine s'appuie sur une **table de routage**.



Protocole IP



7 Application

6 Présentation

5 Session

4 Transport

3 Réseau

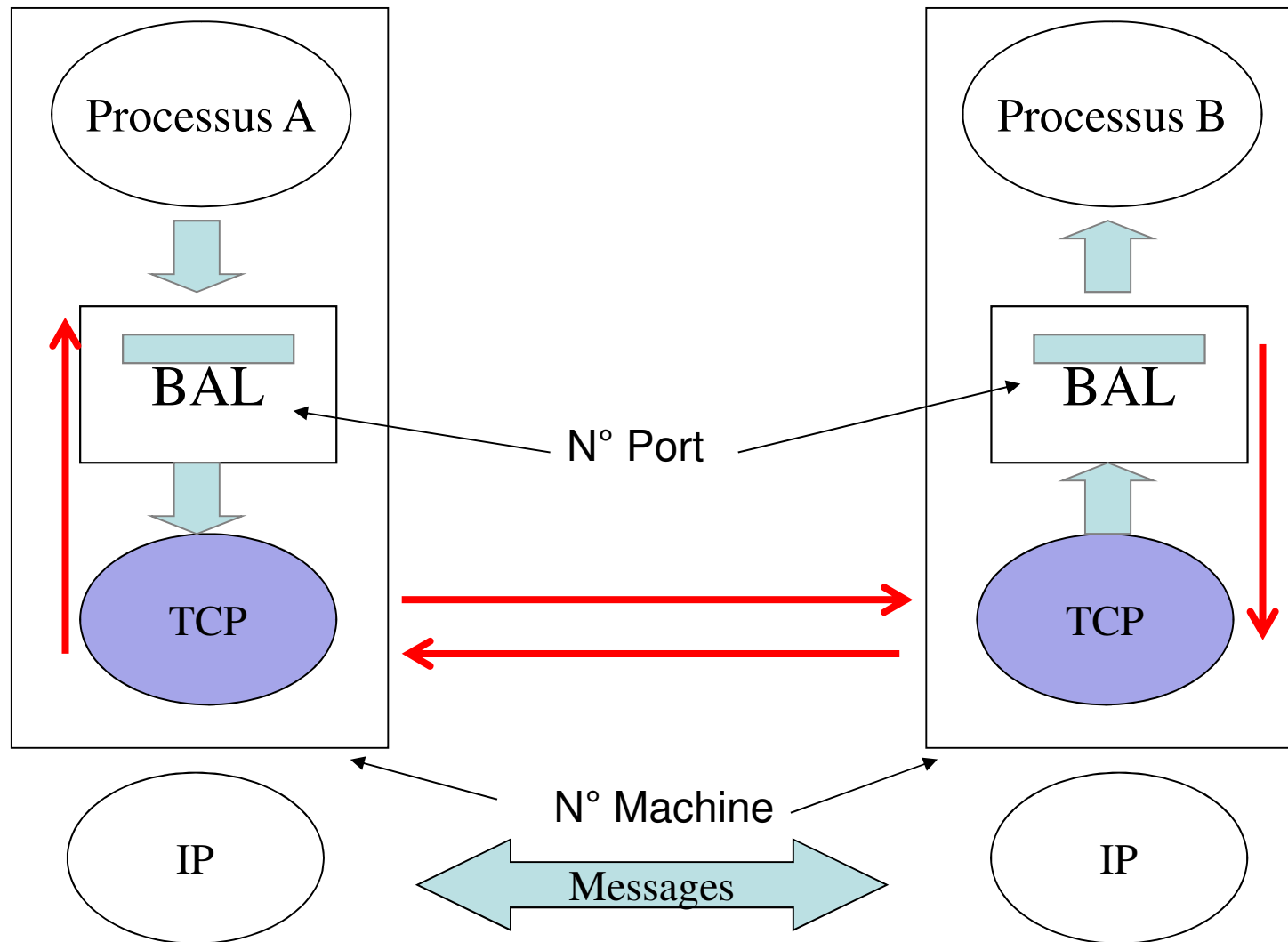
2 Liaison

1 Physique

Le protocole TCP

Couches 4

Protocole TCP

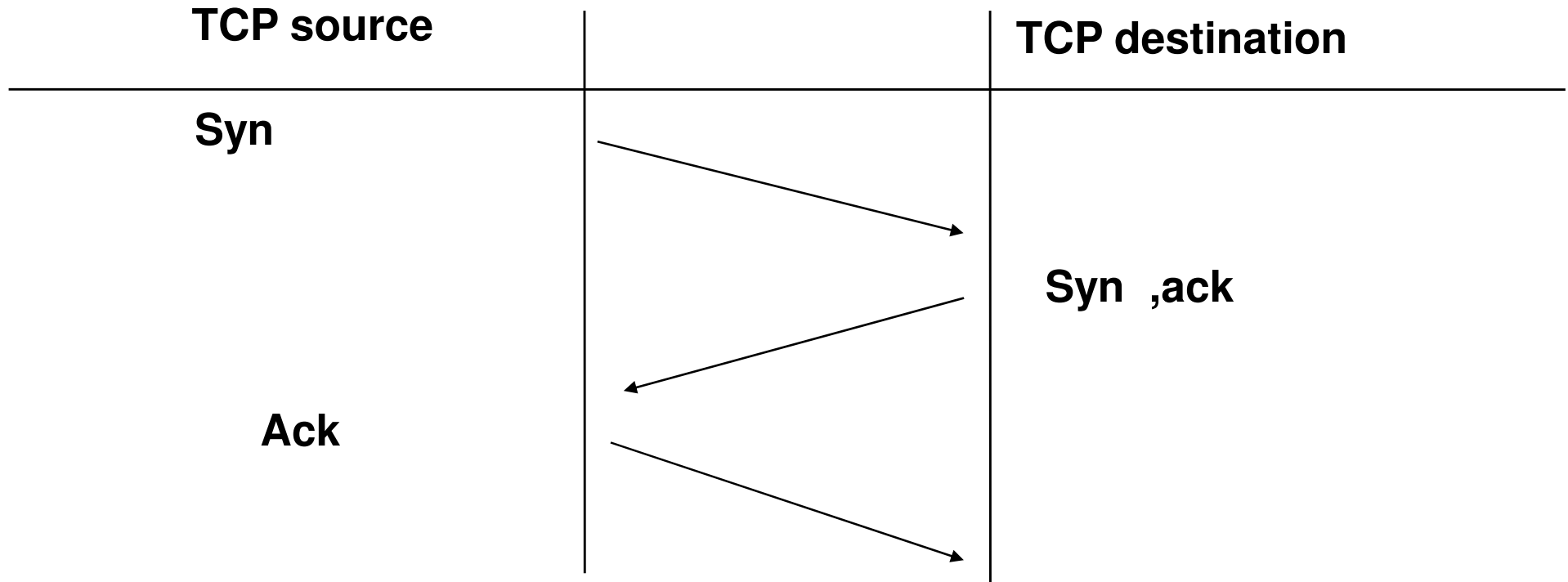


Adressage = identification d'un processus sur une machine donnée =
Numéro de machine (couche 3) + numéro de processus (N° PORT)

Protocole TCP

TCP - Connexion

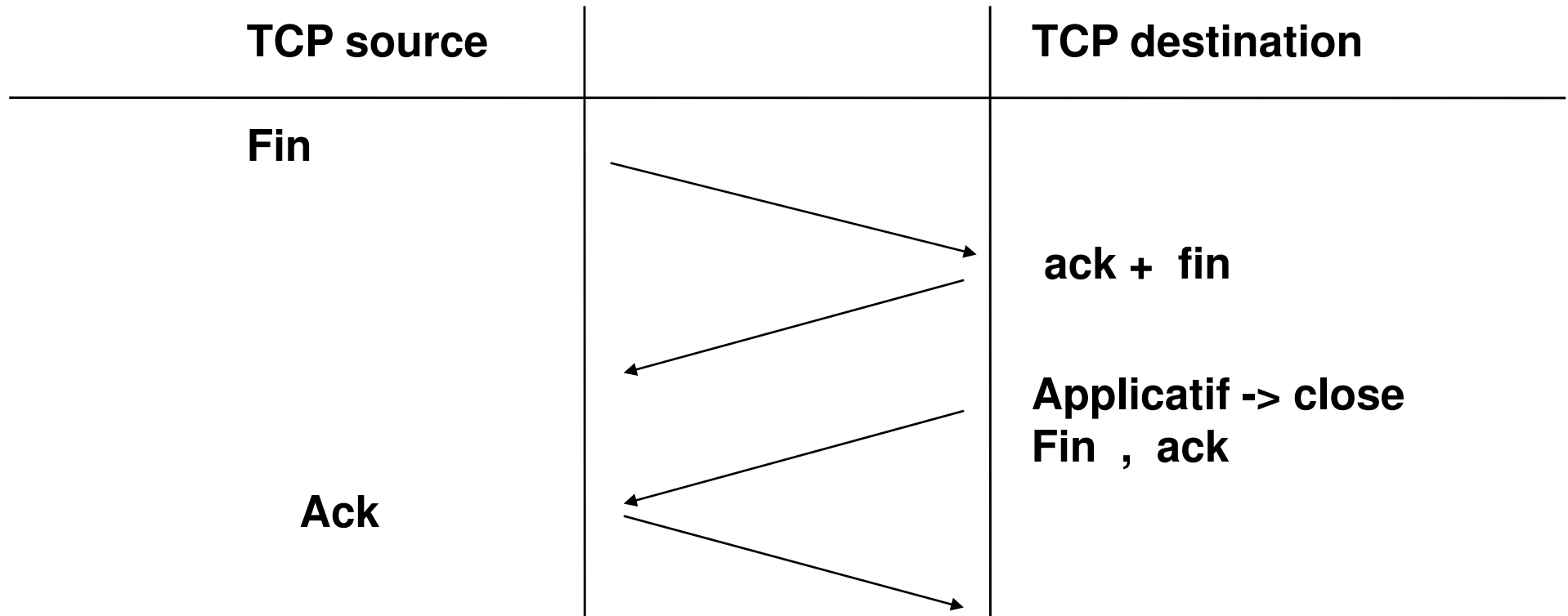
Une connexion TCP est établie en trois temps de manière à assurer la synchronisation nécessaire entre les extrémités:



Protocole TCP

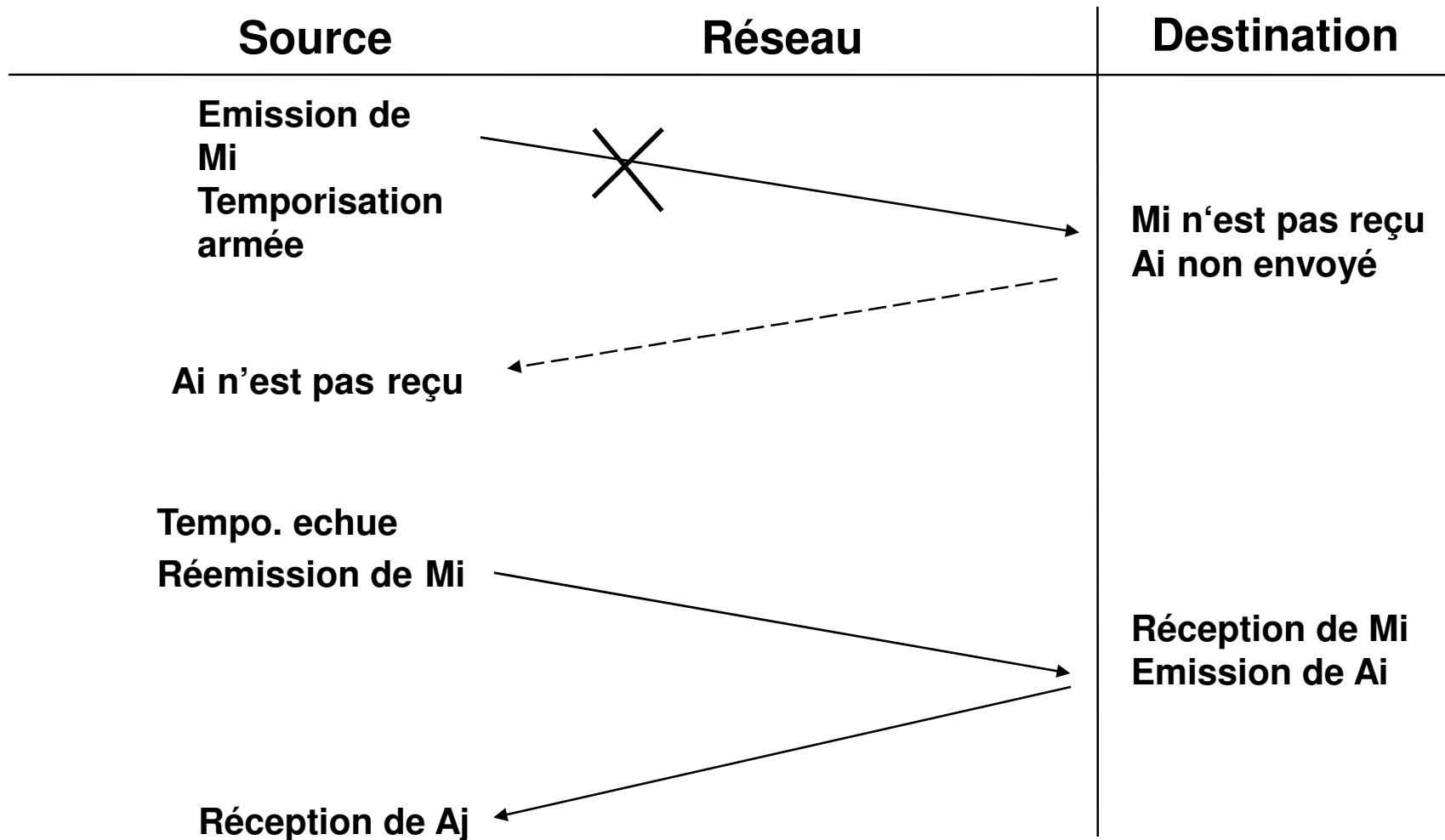
TCP - Déconnexion

Une connexion TCP est libérée en un processus dit "trois temps modifié":



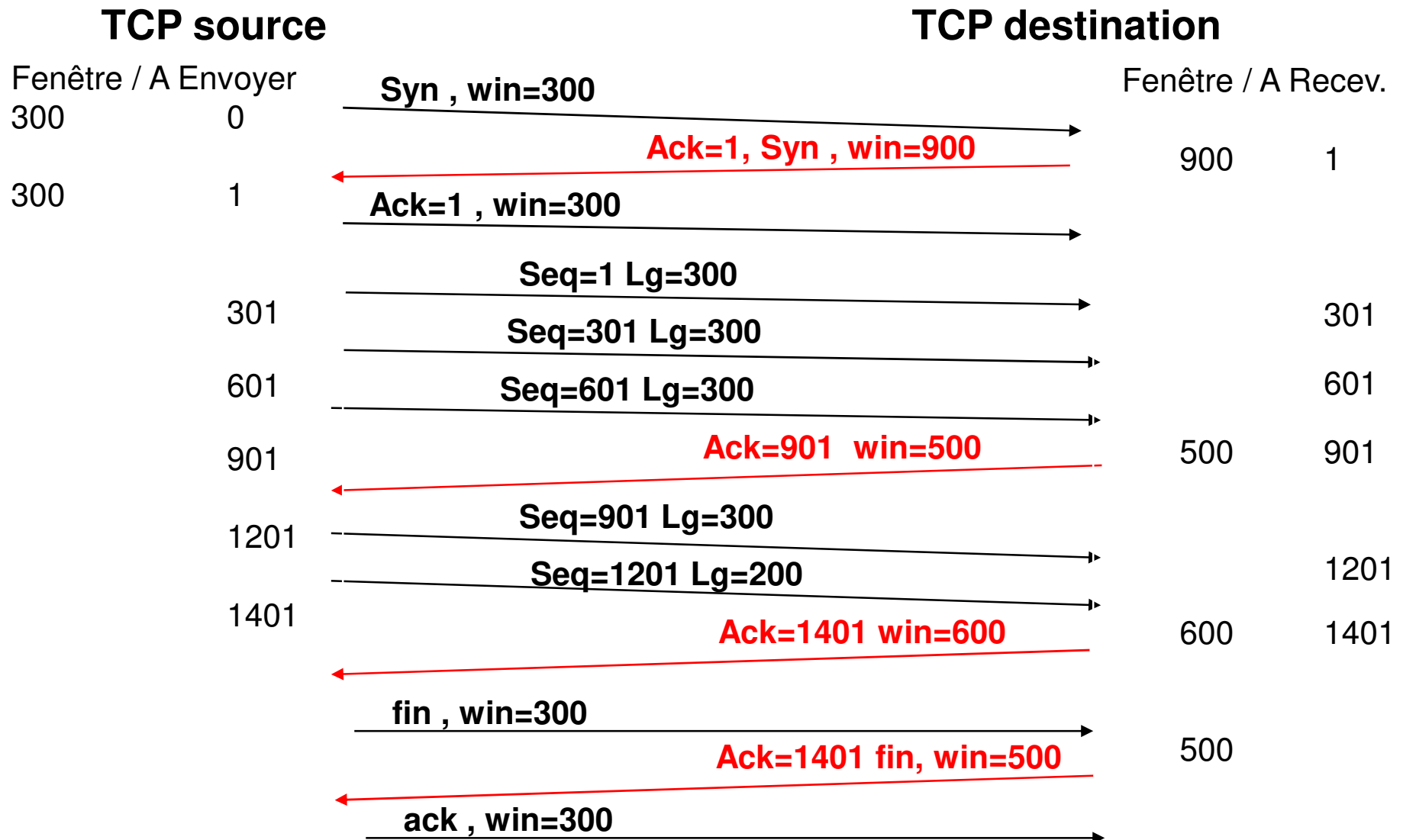
Protocole TCP

TCP - Acquittements



Protocole TCP

TCP – Exemple d'échange



En résumé

TCP – Exemple complet - Récapitulatif

Configuration locale

Adresse IP : 10.10.20.2

Adresse MAC : 00-60-08-61-04-7b

Passerelle : 10.10.1.1

Adresse MAC : 00-01-02-af-f5-e2

Serveur DNS : 10.10.1.2

Configuration distante

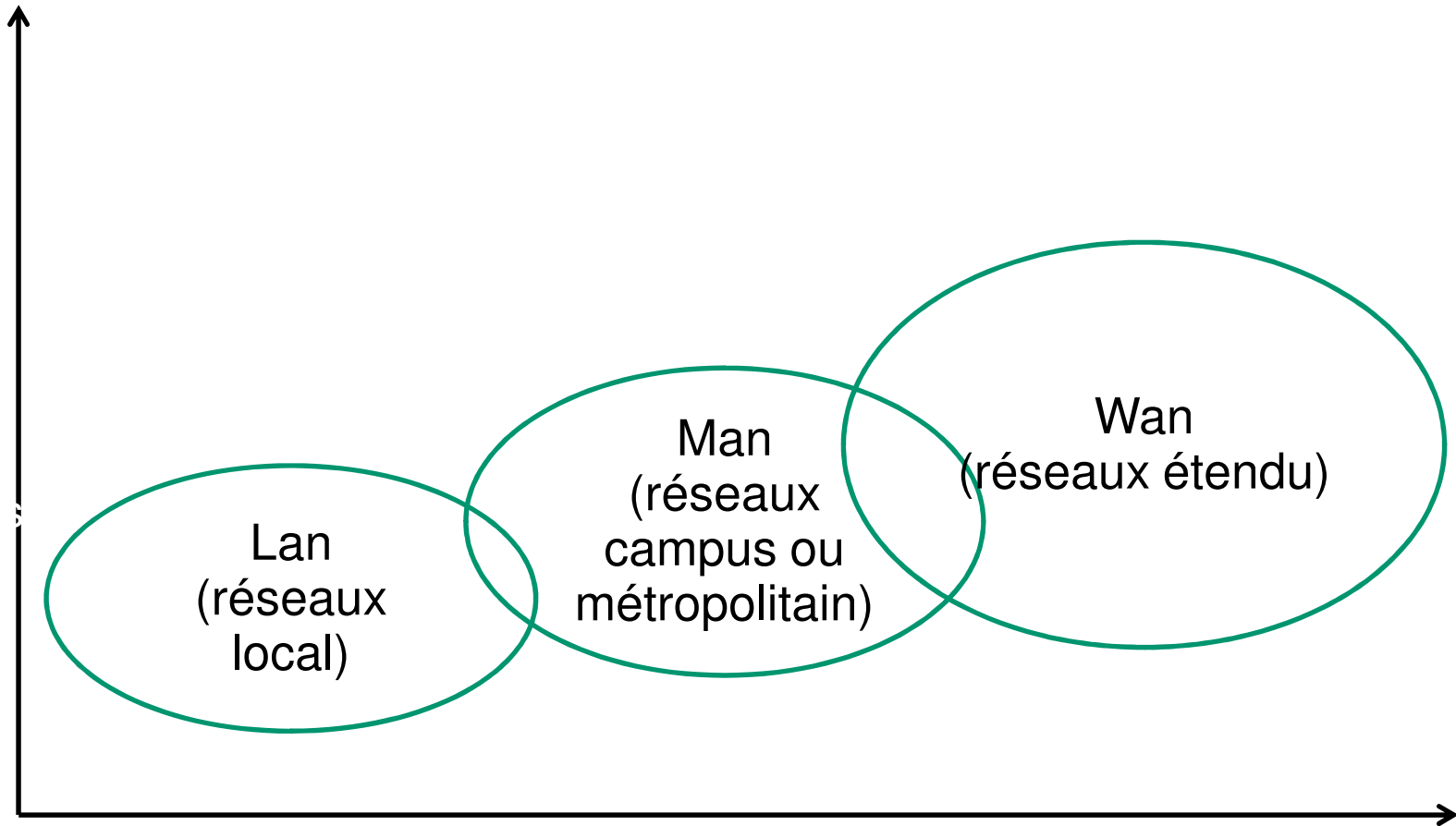
Adresse IP : 173.168.12.1

N°Protocole	Source	Destination	Signification
ARP	00-60-08-61-04-7b	Toutes machines	Qui connaît l'adresse physique de 10.10.1.1 ?
ARP	00-01-02-af-f5-e2	00-60-08-61-04-7b	L'adresse physique de 10.10.1.1 est 000102aff5e2
DNS	10.10.20.2 + 1234	10.10.1.2 + 53	Adresse IP de www.google.fr ?
DNS	10.10.1.2 + 53	10.10.20.2 + 1234	Adresse IP de www.google.fr est 173.168.12.1
TCP	10.10.20.2 + 1234	173.168.12.1 + 80	syn = demande connexion TCP au service HTTP
TCP	173.168.12.1 + 80	10.10.20.2 + 1234	ack + syn : acceptation et demande connexion
TCP	10.10.20.2 + 1234	173.168.12.1 + 80	ack = acceptation connexion
HTTP	10.10.20.2 + 1234	173.168.12.1 + 80	Get /http://...
TCP	173.168.12.1 + 80	10.10.20.2 + 1234	ack
HTTP	173.168.12.1 + 80	10.10.20.2 + 1234	Document HTML
TCP	10.10.20.2 + 1234	173.168.12.1 + 80	ack + fin
TCP	173.168.12.1 + 80	10.10.20.2 + 1234	ack: acceptation de déconnexion
TCP	173.168.12.1 + 80	10.10.20.2 + 1234	fin → on termine la déconnexion
TCP	10.10.20.2 + 1234	173.168.12.1 + 80	ack

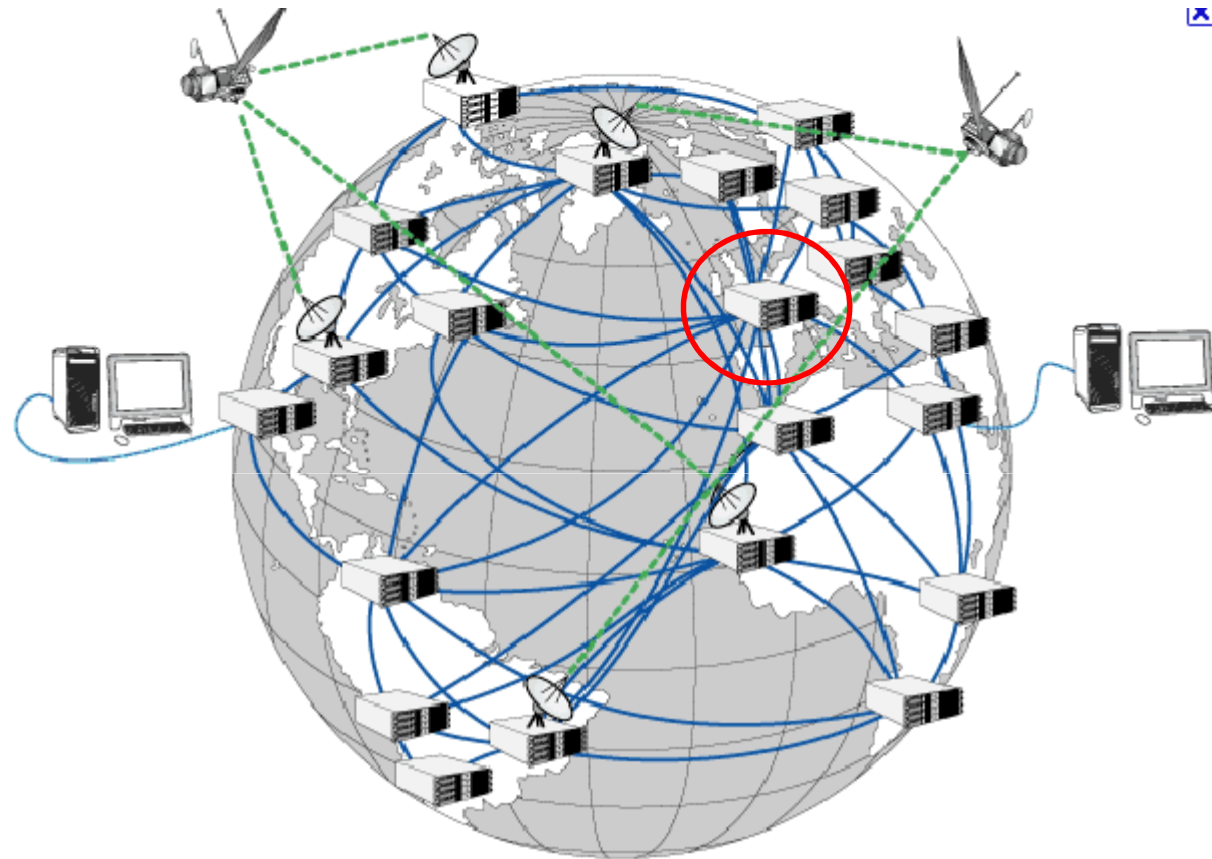
Compléments Couches 3 et 4

Le routage dynamique

Interconnexion de Réseaux (Couche 3)



Interconnexion de Réseaux (Couche 3)



Interconnexion de Réseaux (Couche 3)

Limites du routage statique

- configuration des routeurs peut être difficile, voire impossible lorsque le réseau comporte beaucoup de routeurs et que sa topologie évolue fréquemment.
- surveillance et reconfiguration permanente des routeurs en cas de problème (panne d'un routeur, ligne coupée, etc.),
- Si la route est rétablie, il faut recommencer la manipulation.

Interconnexion de Réseaux (Couche 3)

Routage

Il existe des protocoles qui permettent aux routeurs de s'échanger automatiquement les informations de routage dont ils disposent afin que les tables s'adaptent aux évolutions du réseau.

→ Routage dynamique

Interconnexion de Réseaux (Couche 3)

Pourquoi le routage dynamique ?

L'idée générale du routage dynamique est de délocaliser la tâche de mise à jour au niveau des routeurs.

En effet, chaque appareil :

- Connaît les adresses des réseaux auxquels il est directement relié
- Peut établir un diagnostic sur l'état des liaisons.
- Peut partager ces infos avec ses voisins.

On trouve 2 types de routages dynamiques :

- RIP (basé sur le nombre de routeurs)
- OSPF (basé sur les vitesses des routes)

Protocole RIP

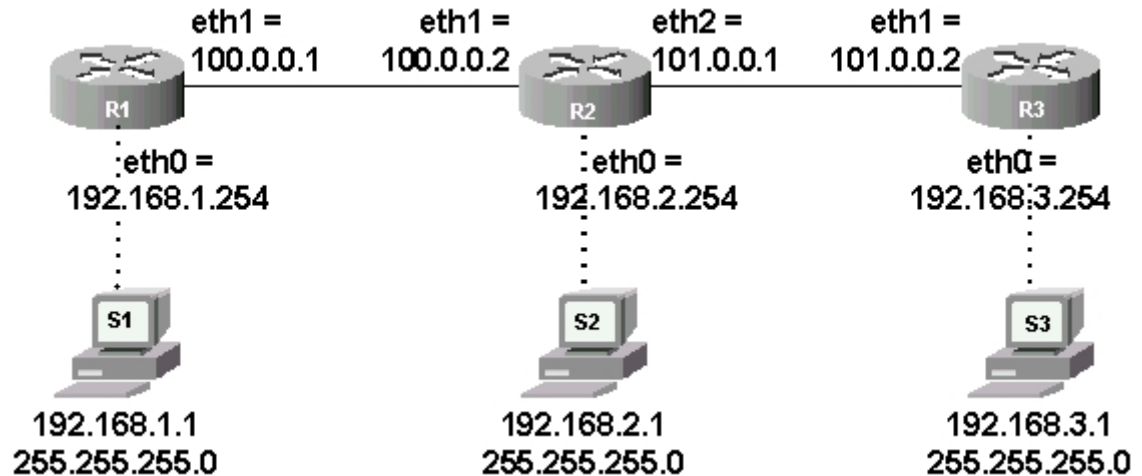
Principe - La notion de distance – métrique

Plusieurs routes mènent à la même destination.

Le routeur doit alors choisir la route qu'il considère la meilleure vers une destination donnée.

Le protocole par RIP considère que la distance la plus courte est la distance correspondant au plus petit nombre de routeurs à traverser (*hop* ou nombre de sauts) avant d'atteindre le destinataire (→ Notion de métrique).

Protocole RIP



Exemple

1 - Etat initial : toutes les interfaces réseau sont actives, aucune route statique n'est définie et le routage RIP est inactif.

2 - Table de routage de R2

Adresse/Préfixe	Moyen de l'atteindre	Distance
100.0	100.0.0.2	1
101.0	100.0.0.1	1
192.168.2	192.168.2.254	1

Protocole RIP

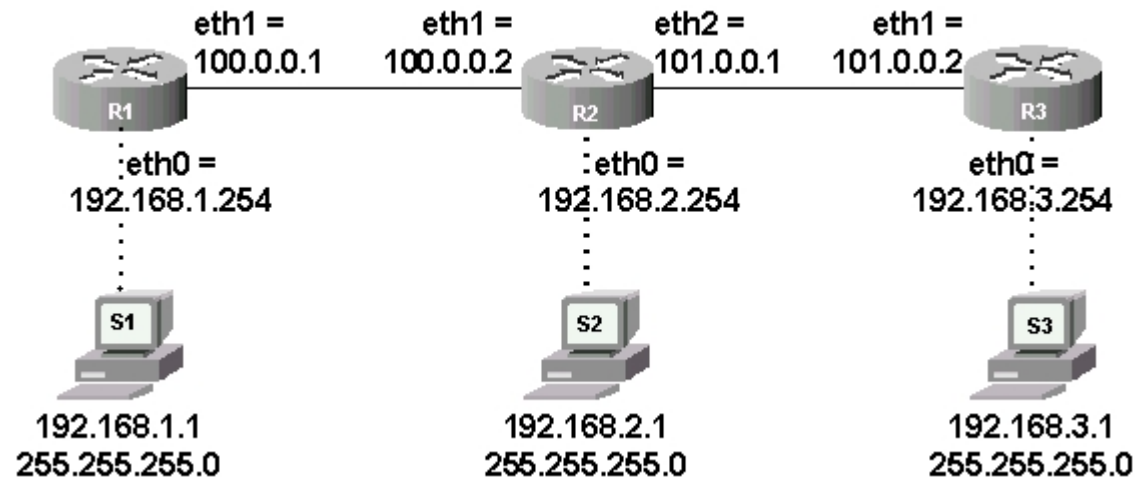
Quelles informations de routage s'échanger ?

Un routeur RIP transmet à ses voisins :

- les adresses réseau qu'il connaît (soit les adresses de ses interfaces, soit les adresses découvertes via les autres routeurs)
- la distance pour les atteindre (**métrique**).

→ Couples adresse/distance = *vecteurs de distance*.

Protocole RIP



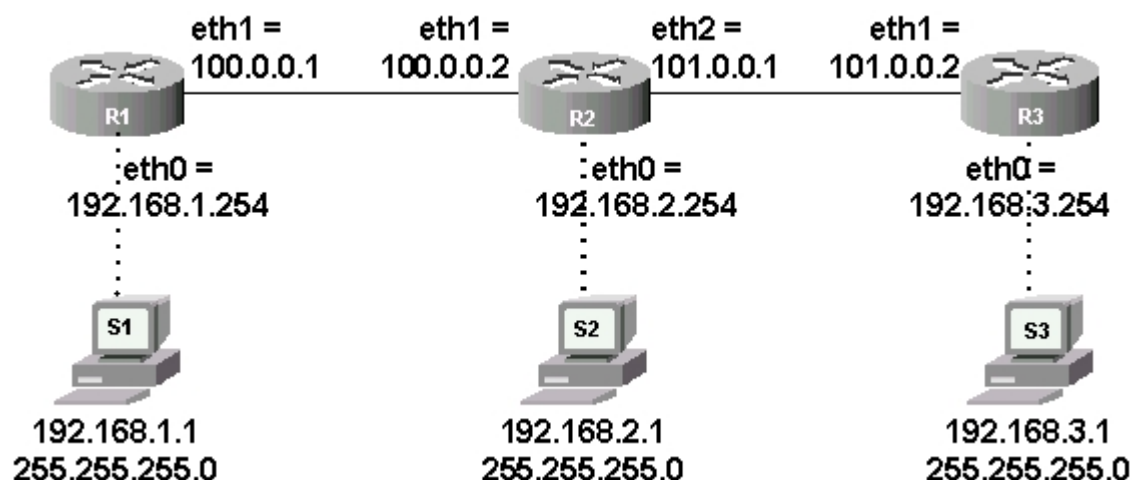
3 – Transmission des données à R2 de R1

R1 transmet à R2 le vecteur de distance { 192.168.1, 1 }

→ "je suis le routeur d'adresse IP 100.0.0.1 et je connais un moyen d'atteindre le réseau 192.168.1 en un saut".

Aucune information sur le réseau commun aux deux routeurs (100.0) n'est transmise car R1 considère que R2 connaît déjà ce réseau

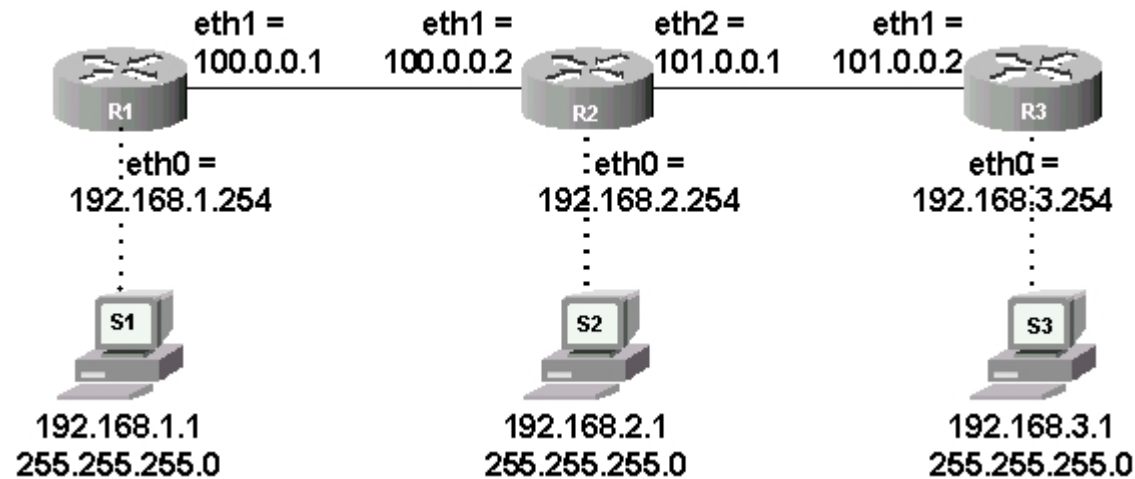
Protocole RIP



4- Table de routage de R2

Adresse/Préfixe	Moyen de l'atteindre	Distance
100.0	100.0.0.2	1
101.0	101.0.0.1	1
192.168.2	192.168.2.254	1
192.168.1	100.0.0.1	2

Protocole RIP

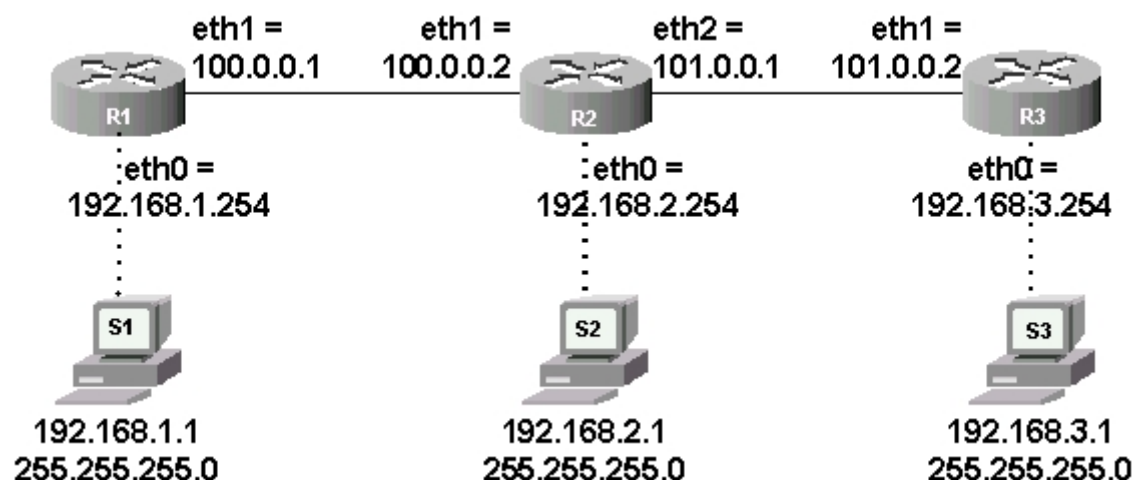


5 – Transmission des données à R2 de R3

R3 transmet à R2 le vecteur de distance {192.168.3, 1}

→ "je suis le routeur d'adresse IP 101.0.0.2 et je connais un moyen d'atteindre le réseau 192.168.3 en un saut« .

Protocole RIP



6 - Table de routage finale de R2

Adresse/Préfixe	Moyen de l'atteindre	Distance
100.0	100.0.0.2	1
101.0	101.0.0.1	1
192.168.2	192.168.2.254	1
192.168.1	100.0.0.1	2
192.168.3	101.0.0.1	2

Protocole RIP

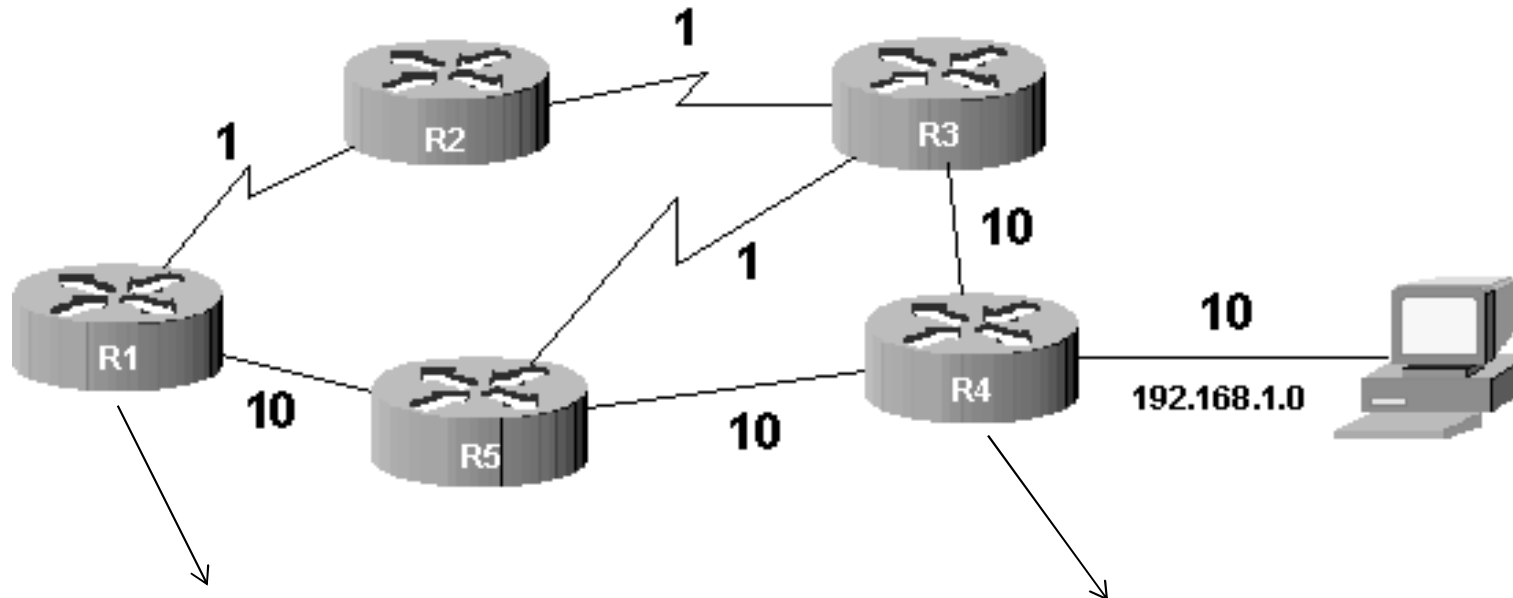
Conclusion

RIP est peu utilisé en exploitation car :

- limité à 15 routeurs soit 16 segments de réseau.
- gros consommateur de bande passante (toutes les 30 secondes, l'intégralité de la table RIP est diffusée même si elle n'a subi aucune modification).
- la métrique utilisée ne garantit pas que le routage soit optimal (débit ou coût en particulier).
- délai long avant que tous les routeurs ne possèdent des tables cohérentes dans certaines situations.

→ Développement d'un protocole de routage beaucoup plus efficace : OSPF

Autres limites de RIP



Destination	Passerelle	Dist
R2	R1 (direct)	1
R5	R1 (direct)	1
R3	R2	2
R4	R5	2
192.168.1.0	R5	3

Destination	Passerelle	Dist
R5	R4 (direct)	1
R3	R4 (direct)	1
192.168.1.0	R4(direct)	1
R1	R5	2
R2	R3	2

Protocole OSPF

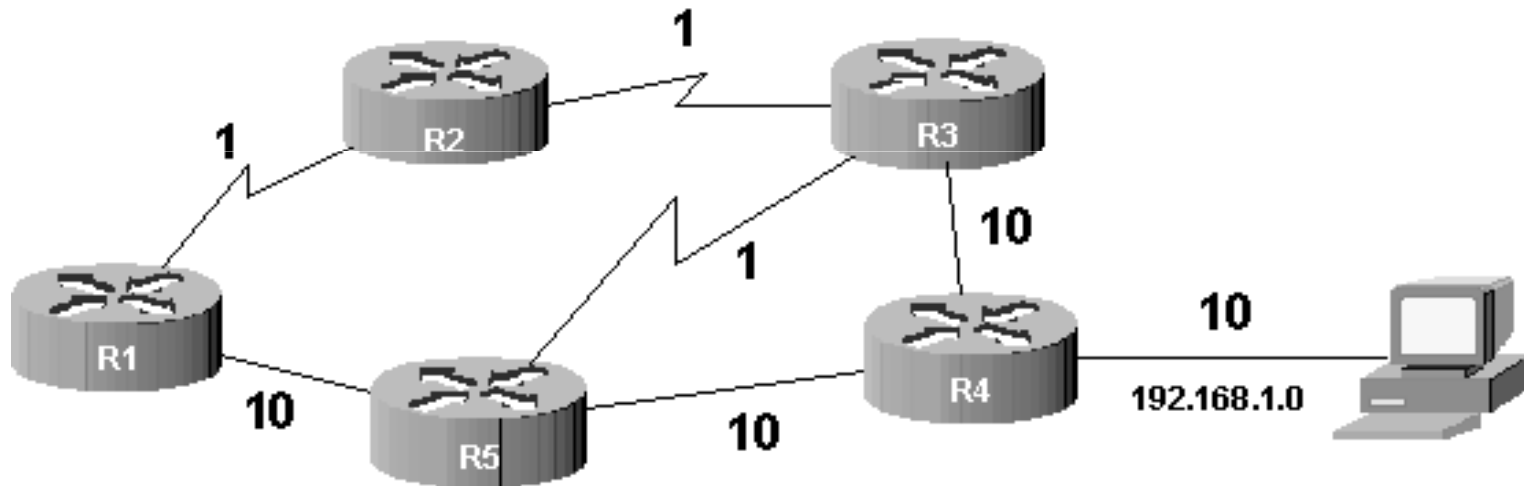
Principe de fonctionnement de OSPF :

Il attribue un coût à chaque liaison (dénommée *lien* dans le jargon OSPF) afin de privilégier l'élection de certaines routes. Plus le coût est faible, plus le lien est intéressant. La formule de calcul est : coût = référence / bande passante du lien.

Type de réseau	Coût par défaut
Ethernet > = 100 Mbps	1
FDDI	1
Ethernet 10 Mbps	10
E1 (2,048 Mbps)	48
T1 (1,544 Mbps)	65
64 Kbps	1562
56 Kbps	1785
19.2 Kbps	5208
Type de réseau	Coût par défaut

Protocole OSPF

Différence entre RIP et OSPF



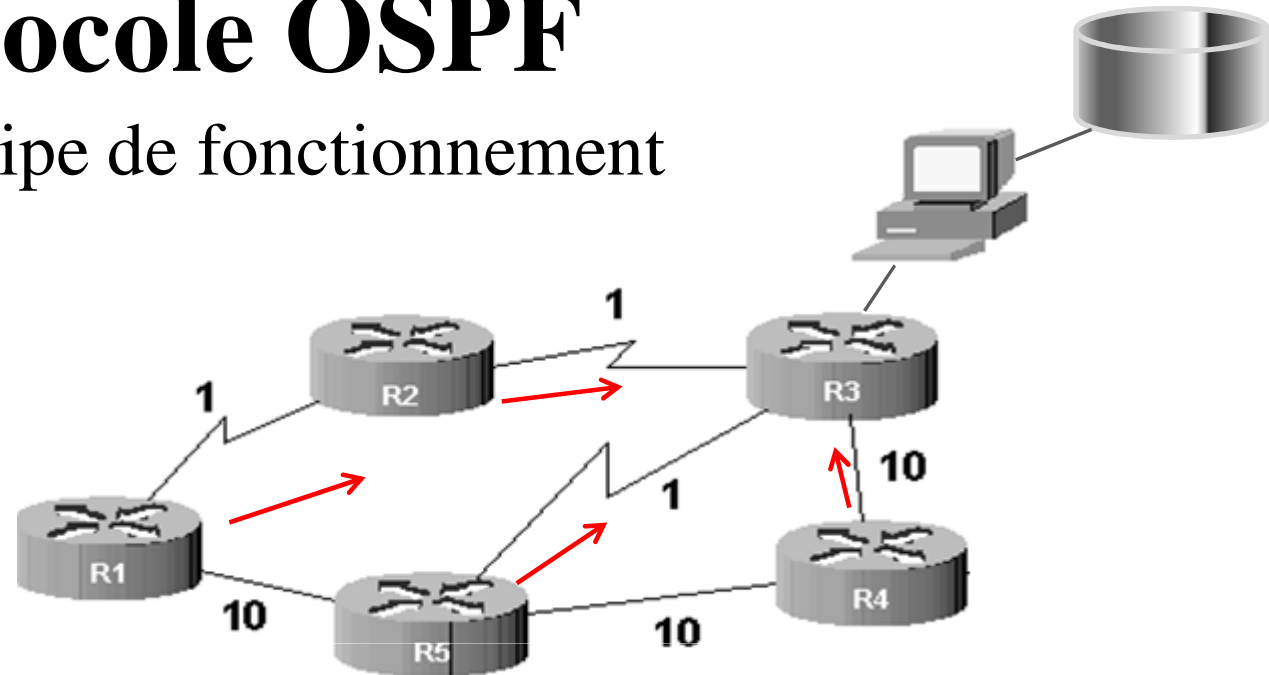
Pour aller de R1 à 192.168.1.0 :

RIP choisirait le chemin : R1-R5-R4

OSPF choisirait le chemin : R1-R2-R3-R4

Protocole OSPF

Principe de fonctionnement



1 – La base de données topologique

Tous les routeurs d'un même réseau travaillent sur une base de données partagée, à laquelle ils transmettent les informations.

On construit alors un graphe des routes et on détermine les meilleures routes entre chaque routeur.

Protocole OSPF

Principe de fonctionnement

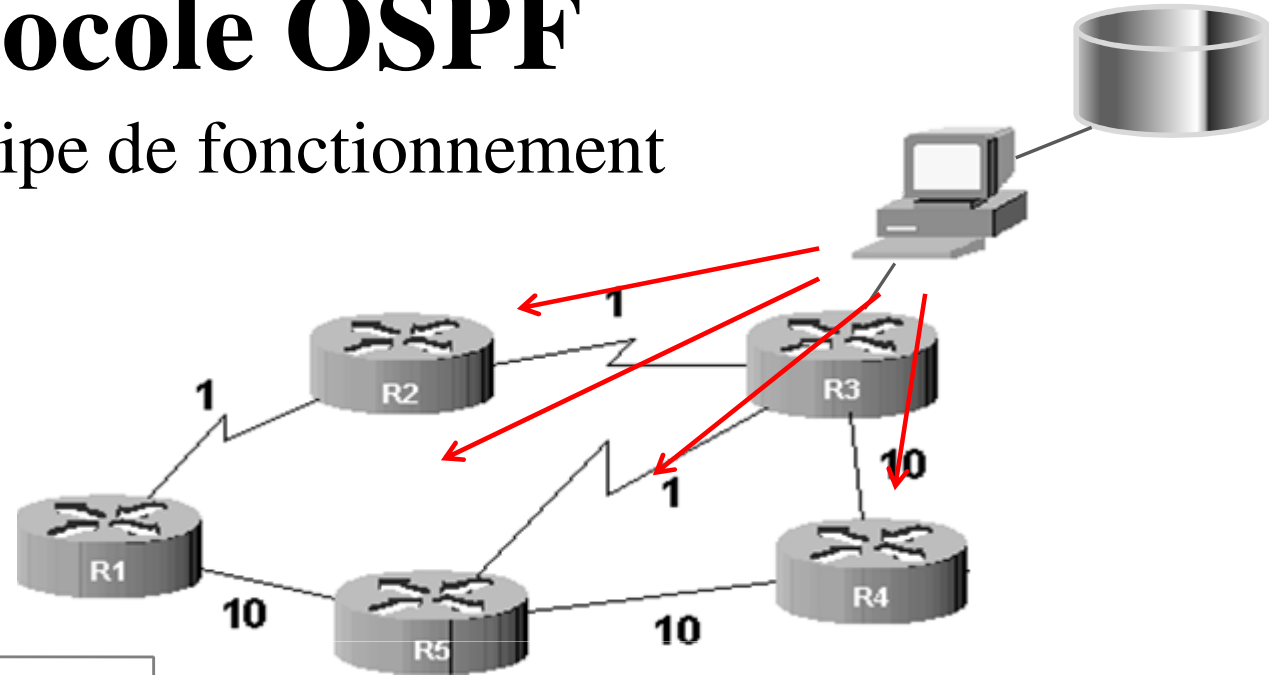


TABLE	R1	
Dest	Passerell	Coût
192.168.1.10	R2	22

2 – La mise à jour des tables

Les différentes routes sont renvoyées aux routeurs pour qu'ils mettent à jour leur table de routage.

Protocole OSPF

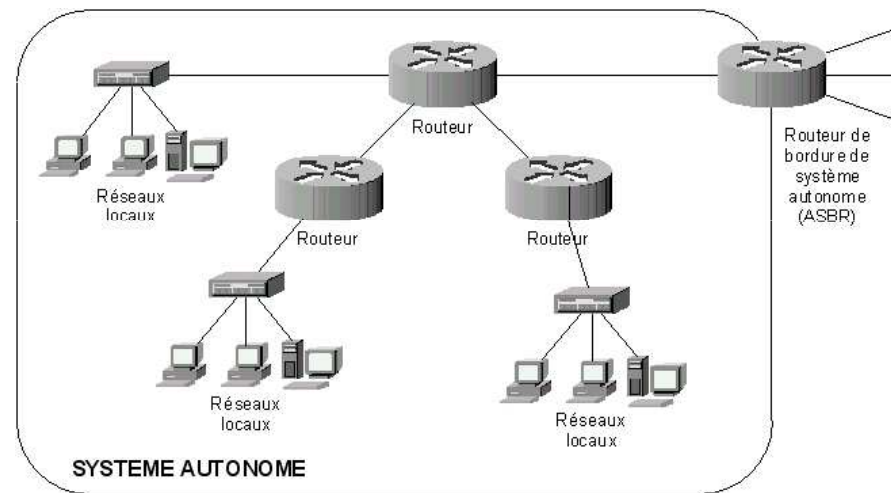
Limites de OSPF

Ce protocole peut être gourmand en puissance de calcul et en mémoire lorsque le réseau comporte beaucoup de routes ou qu'il y a de fréquentes modifications de topologie.

Protocole BGP

RIP et OSPF sont des protocoles IGP (*Interior Gateway Protocol*), c'est-à-dire qu'il agissent au sein d'un AS (système autonome)..
Un Système

Autonome et un ensemble de réseaux gérés par une même autorité et utilisant une même protocole



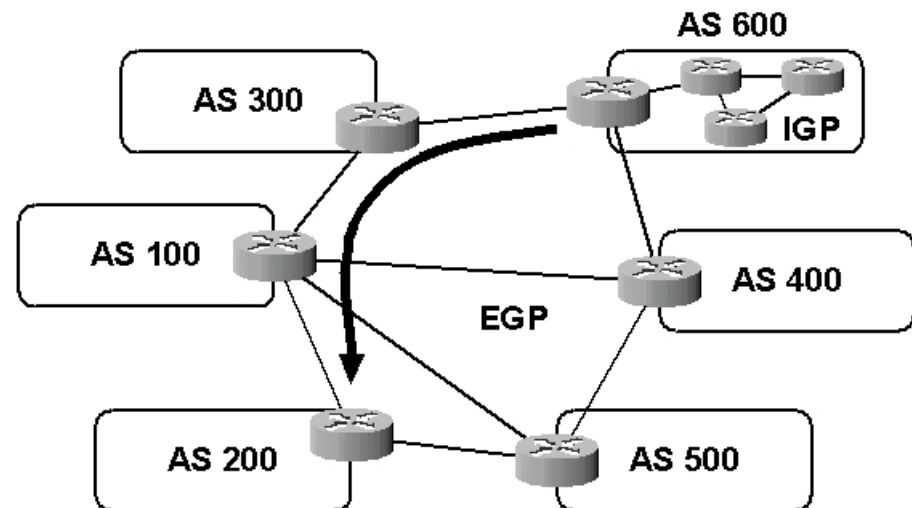
Les multinationales, les opérateurs de télécom ou les fournisseurs d'accès à Internet détiennent un système autonome.

Protocole BGP

Il faut désigner un ou plusieurs routeurs, à la frontière de votre AS, pour propager les informations (ASBR Routeur de Bordure de Système Autonome).

Pour assurer le routage entre les AS, il faut un protocole de type EGP (*Exterior Gateway Protocol*) sur les ASBR.

Dans le cas d'Internet, c'est généralement BGP (*Border Gateway Protocol*) qui réalise cela.



Protocole BGP

BGP et politique de routage

Vous préférerez que votre trafic transite par tel AS, car l'administrateur est un pote ou que vous avez négocié un tarif avantageux.

Vous ne voulez pas qu'il transite par tel AS car vos flux sont incompatibles avec la législation en vigueur dans le pays ou pour toute autre considération politique, économique ou de sécurité.

Il faut alors définir une politique de routage sur Internet :

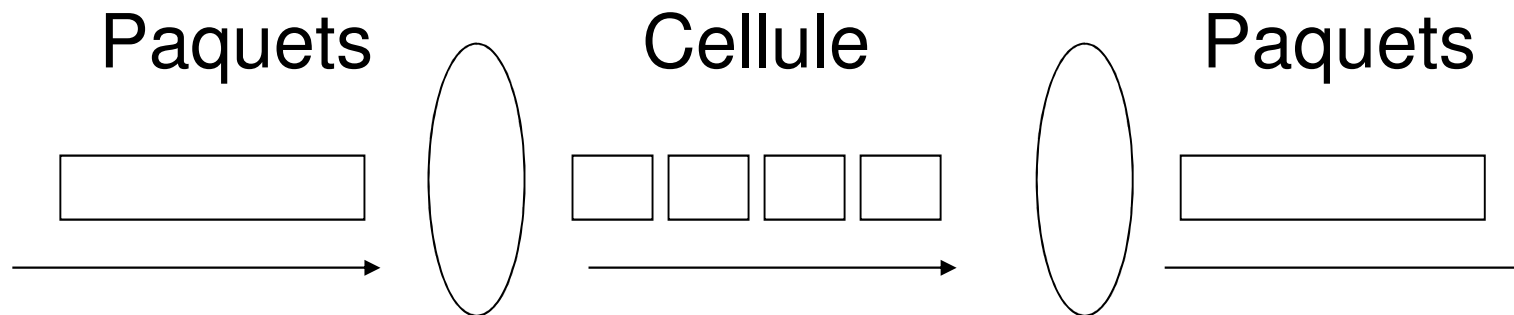
→ BGP rend ce service.

Protocole ATM

Réseaux ATM

La cellule : principe de base

Transmettre les données dans des paquets de petite taille ➔ les cellules



Réseaux ATM

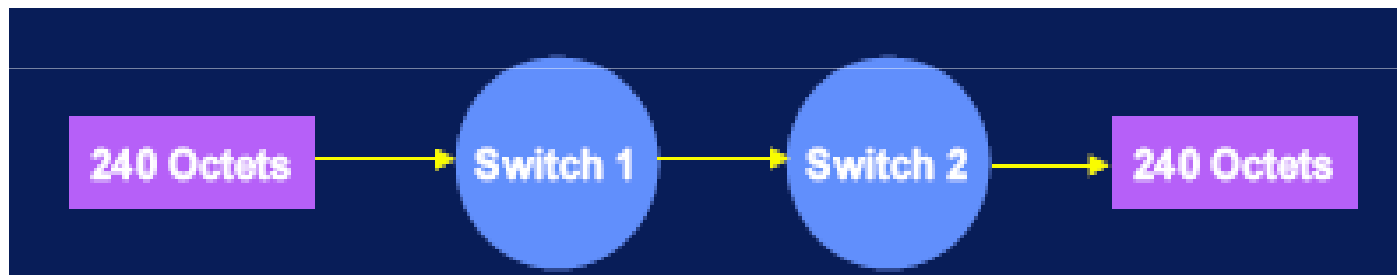
Intérêt des cellules de petite taille

- Moins de perte de place, les cellules sont toutes remplies par de l'information
- Optimisation du temps d'insertion de la cellule dans le support
- Optimisation du temps de routage et du délai de transmission

Réseaux ATM

Exemple d'après : l'UREC

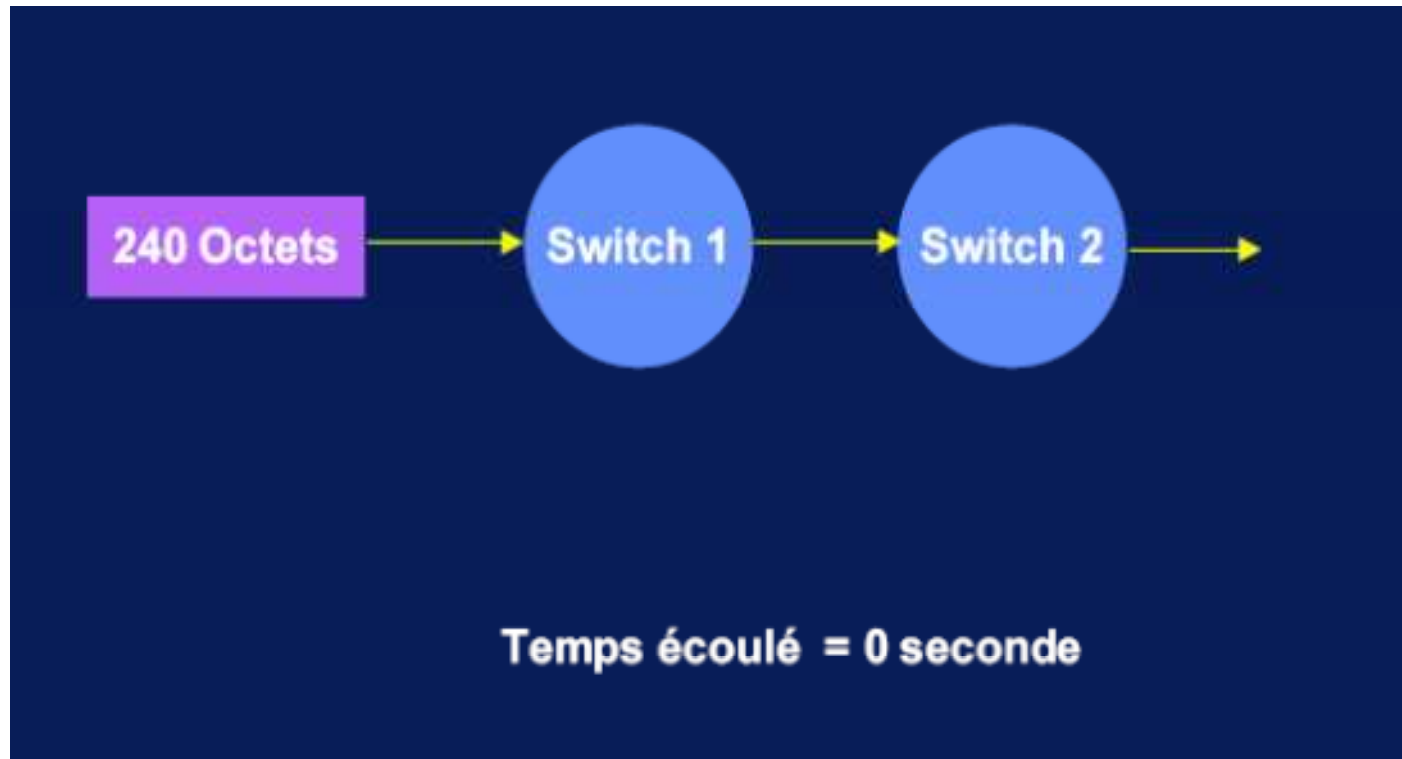
Objectif = transmettre une trame de 240 octets à travers 2 commutateurs.



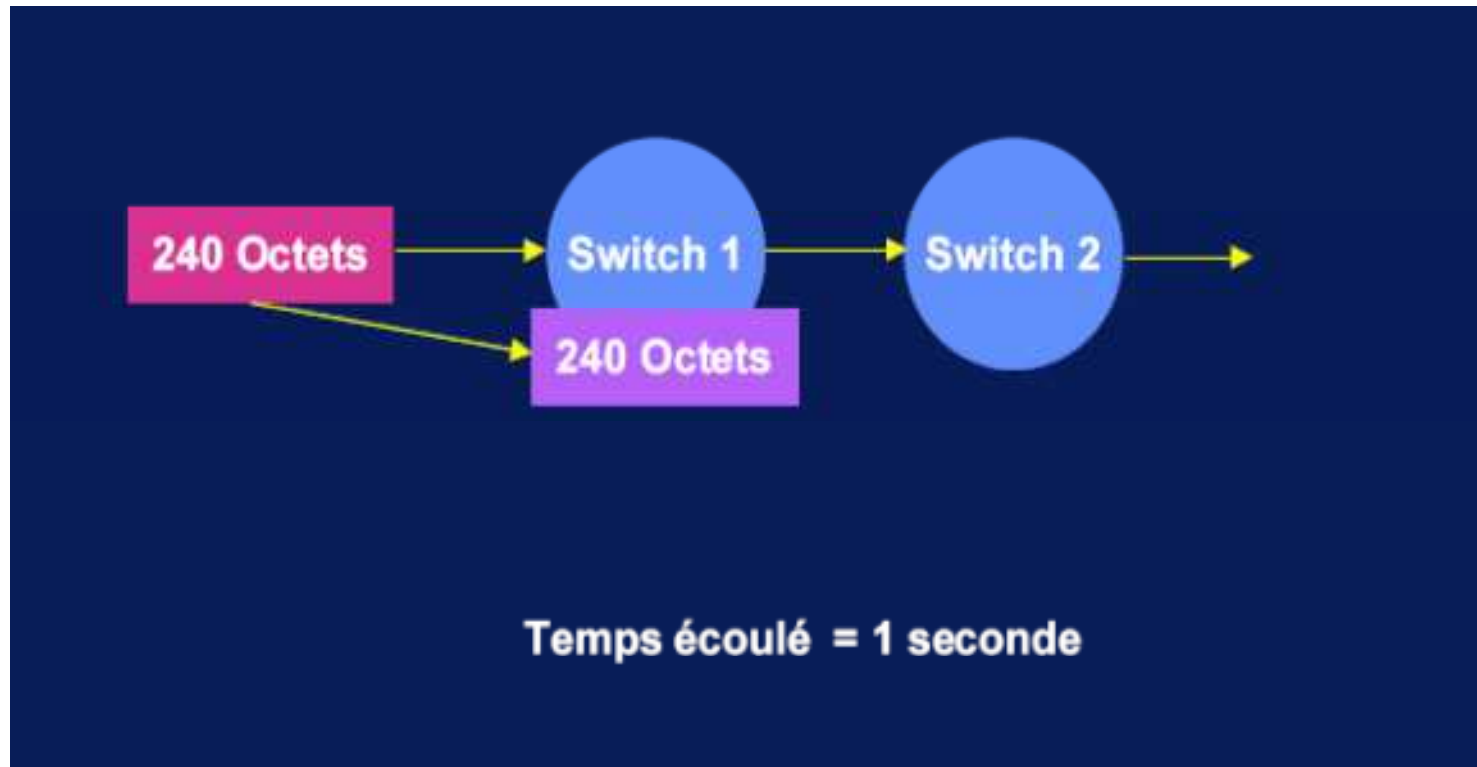
Hypothèses :

- Lien à 240 octets/s
- Commutateurs infiniment rapides

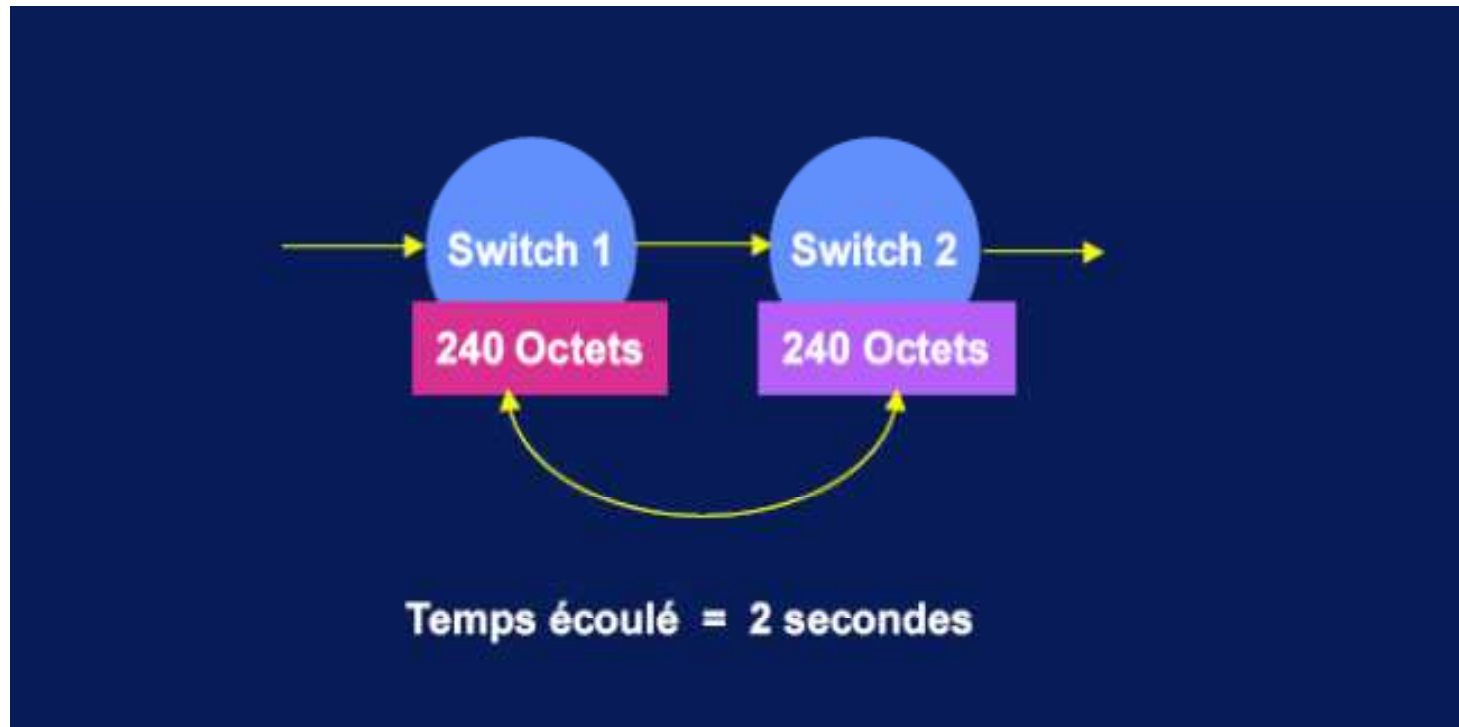
Réseaux ATM



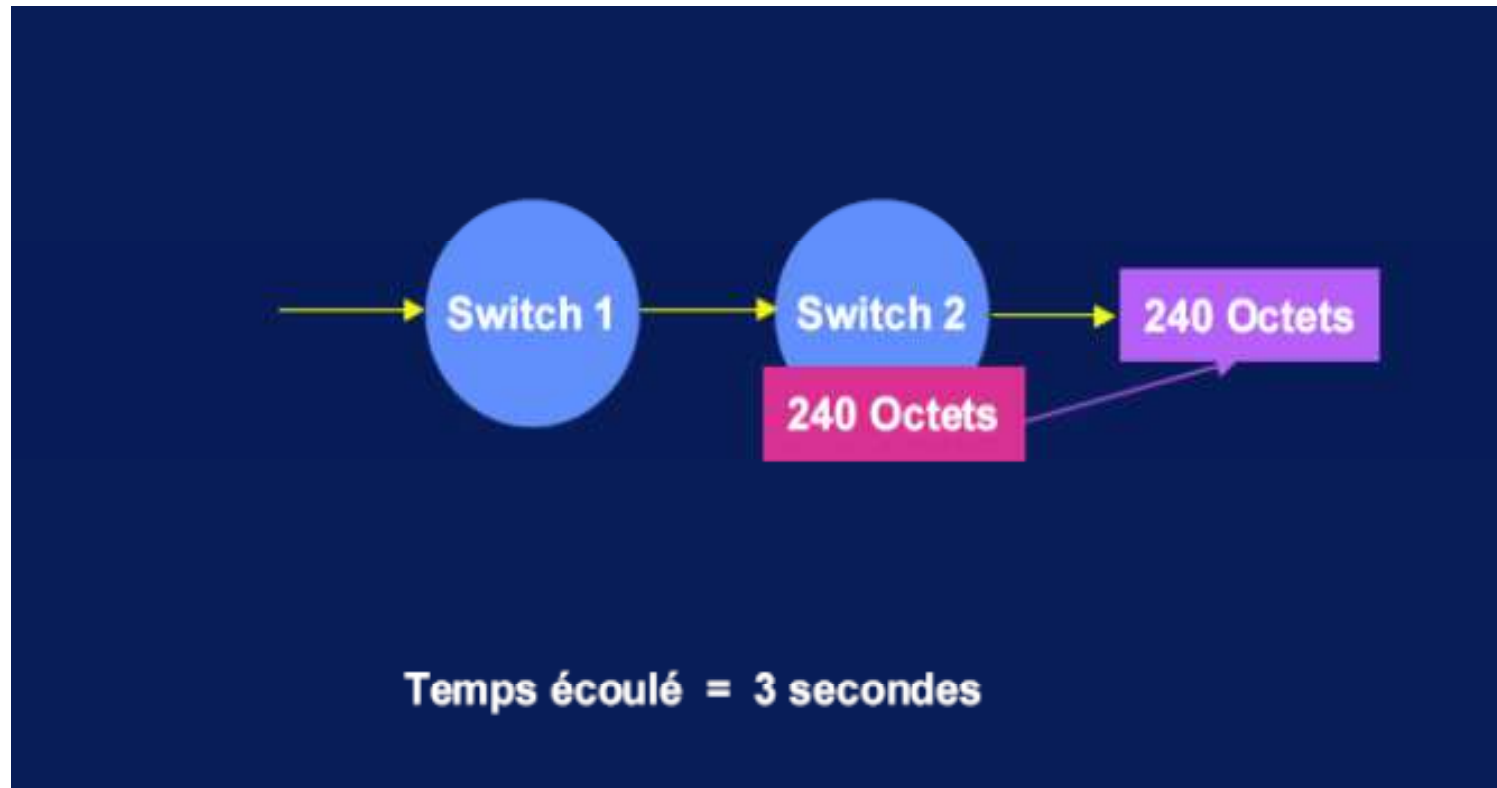
Réseaux ATM



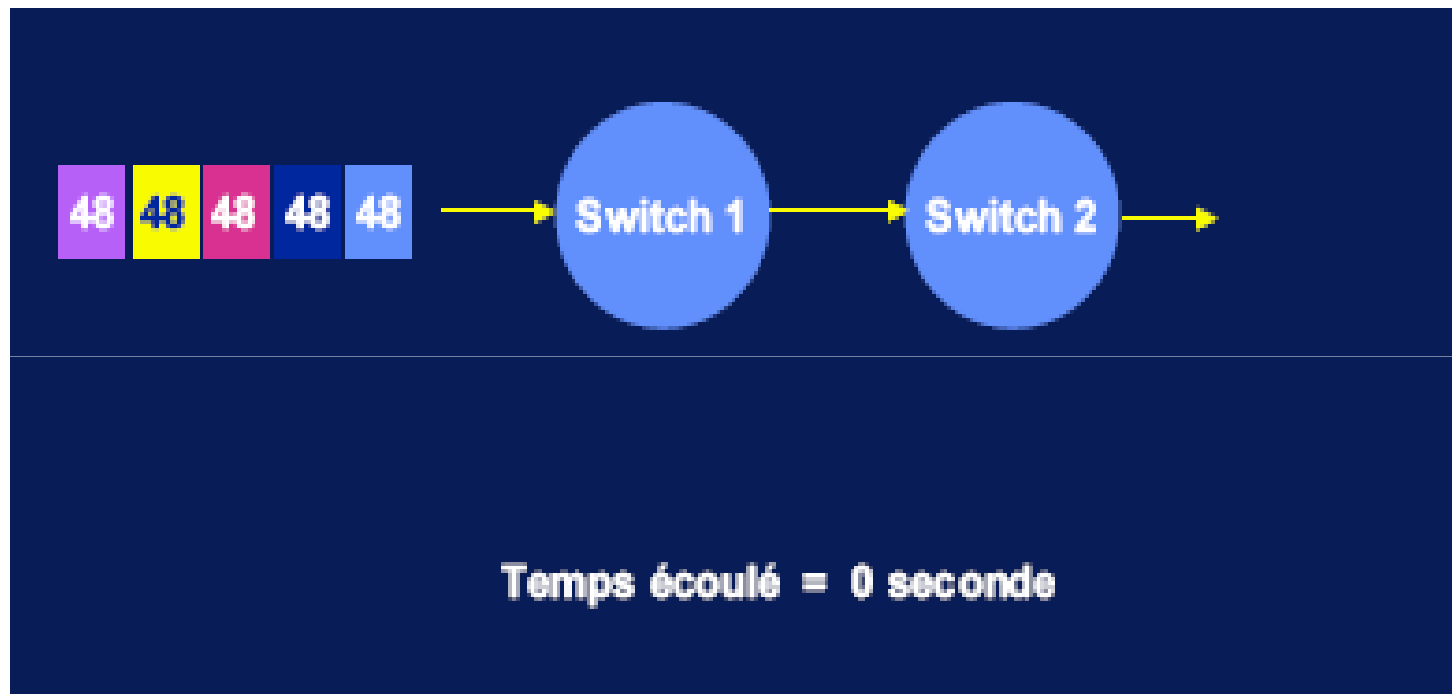
Réseaux ATM



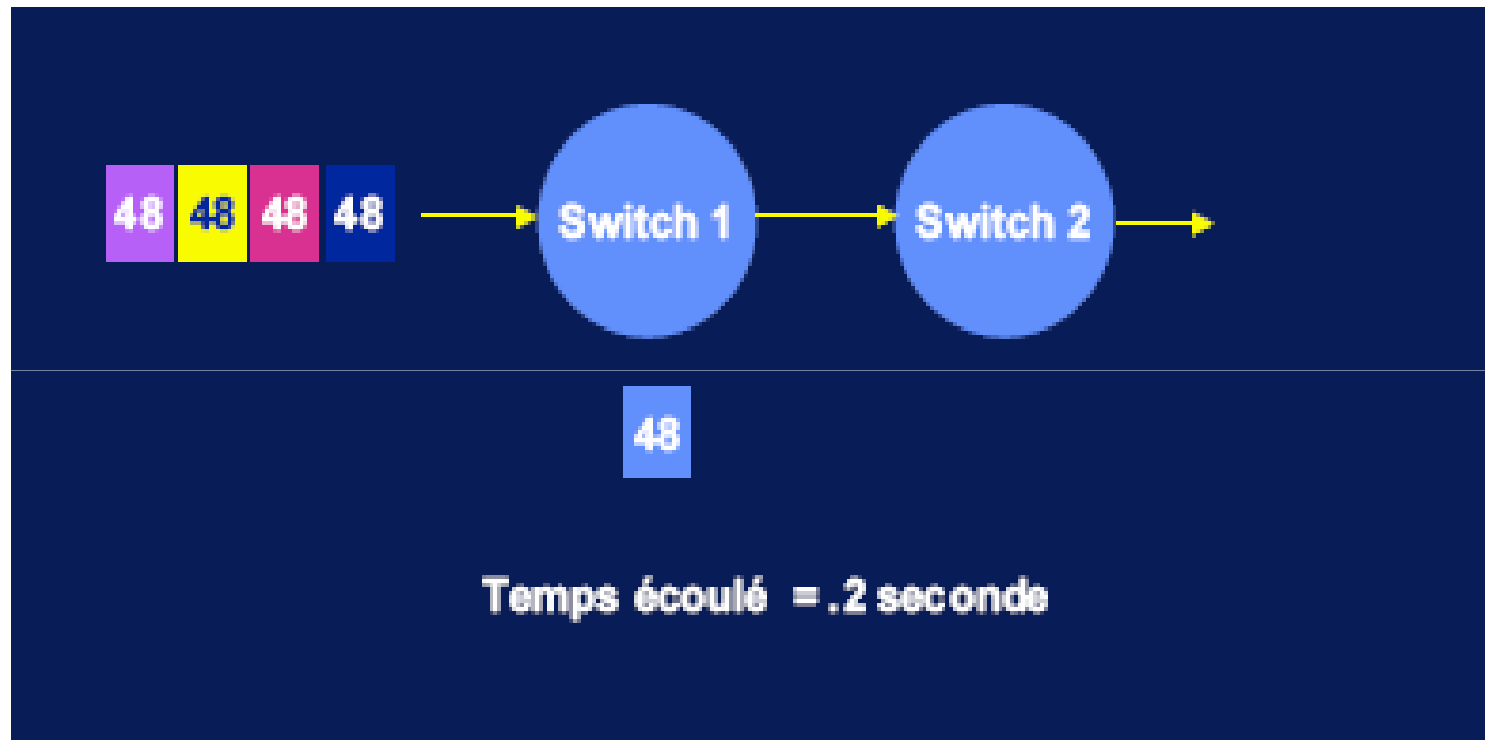
Réseaux ATM



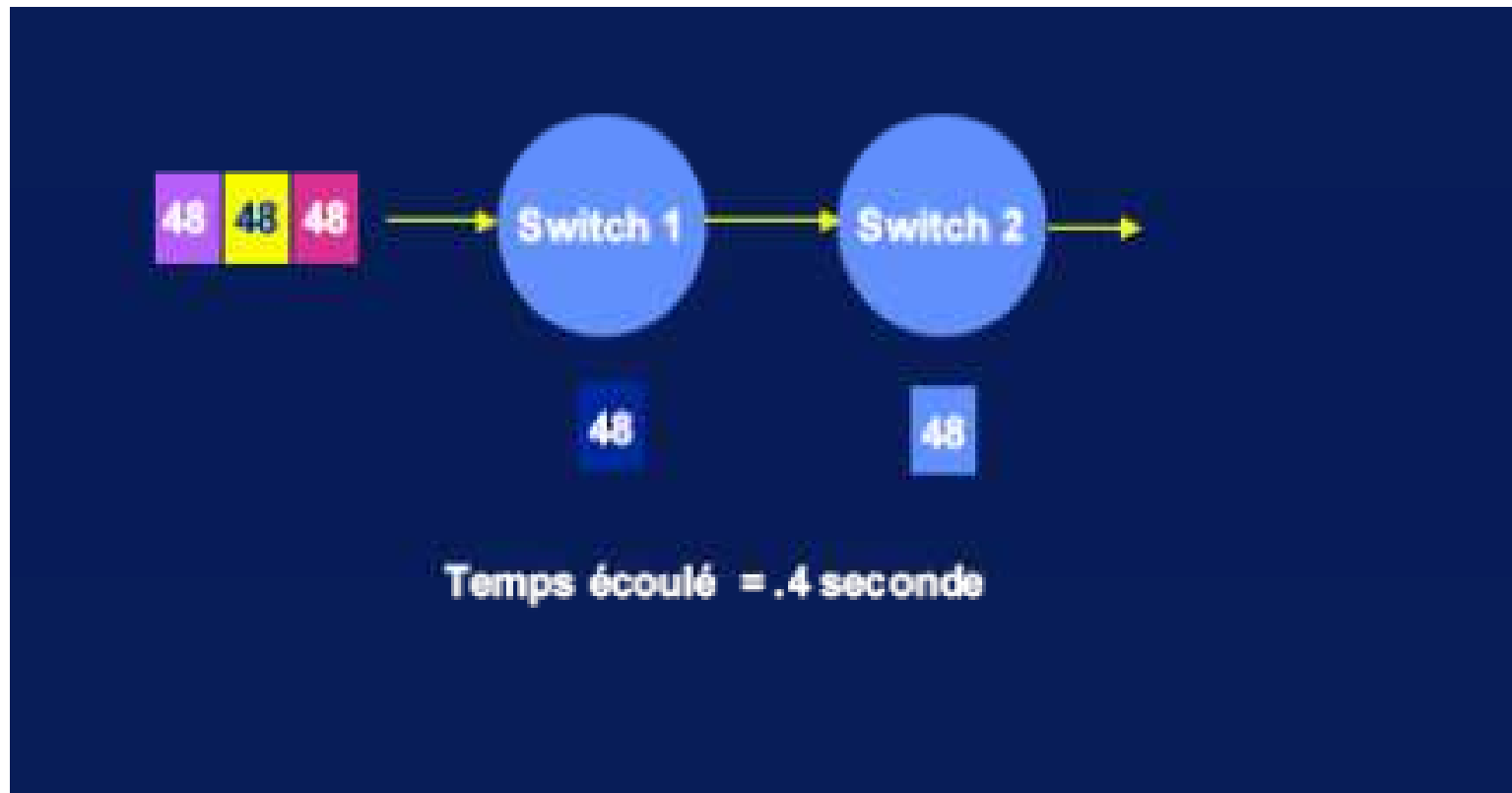
Réseaux ATM



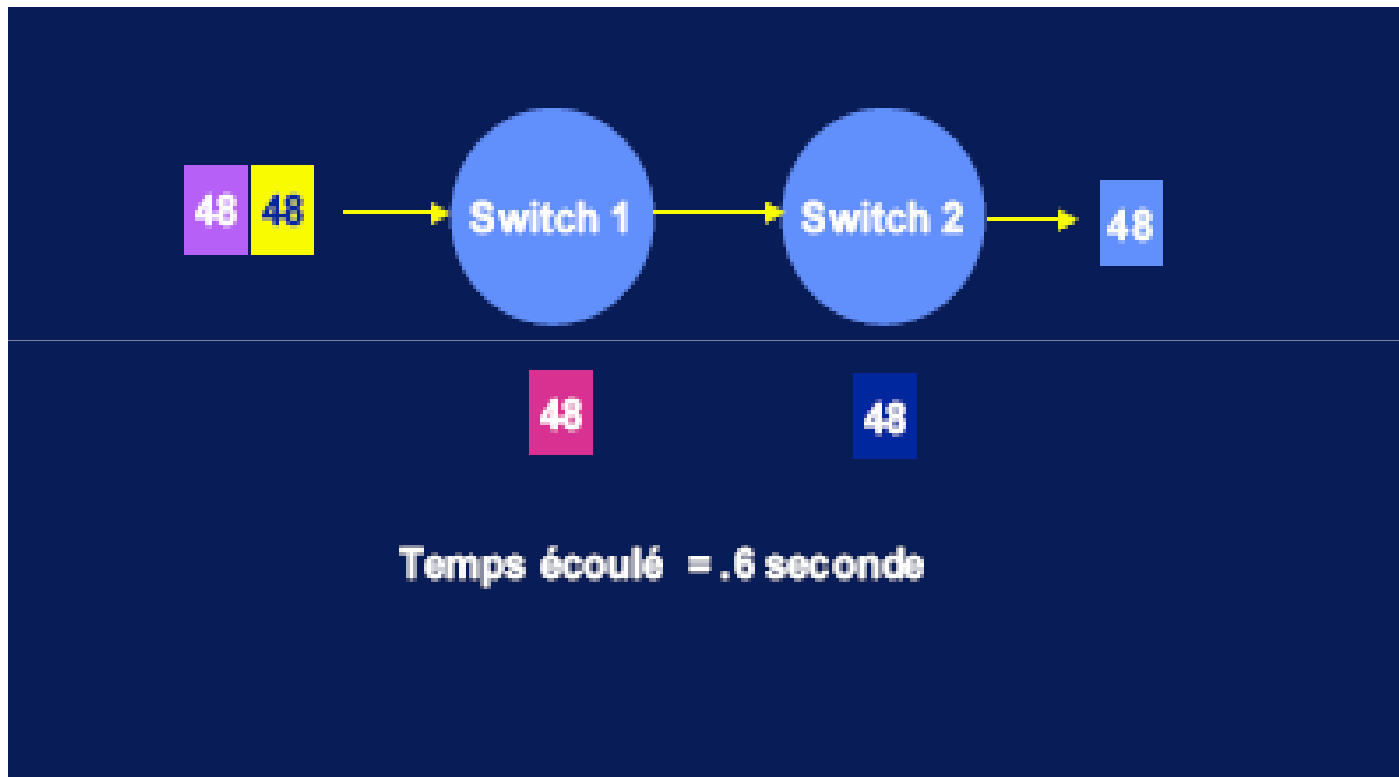
Réseaux ATM



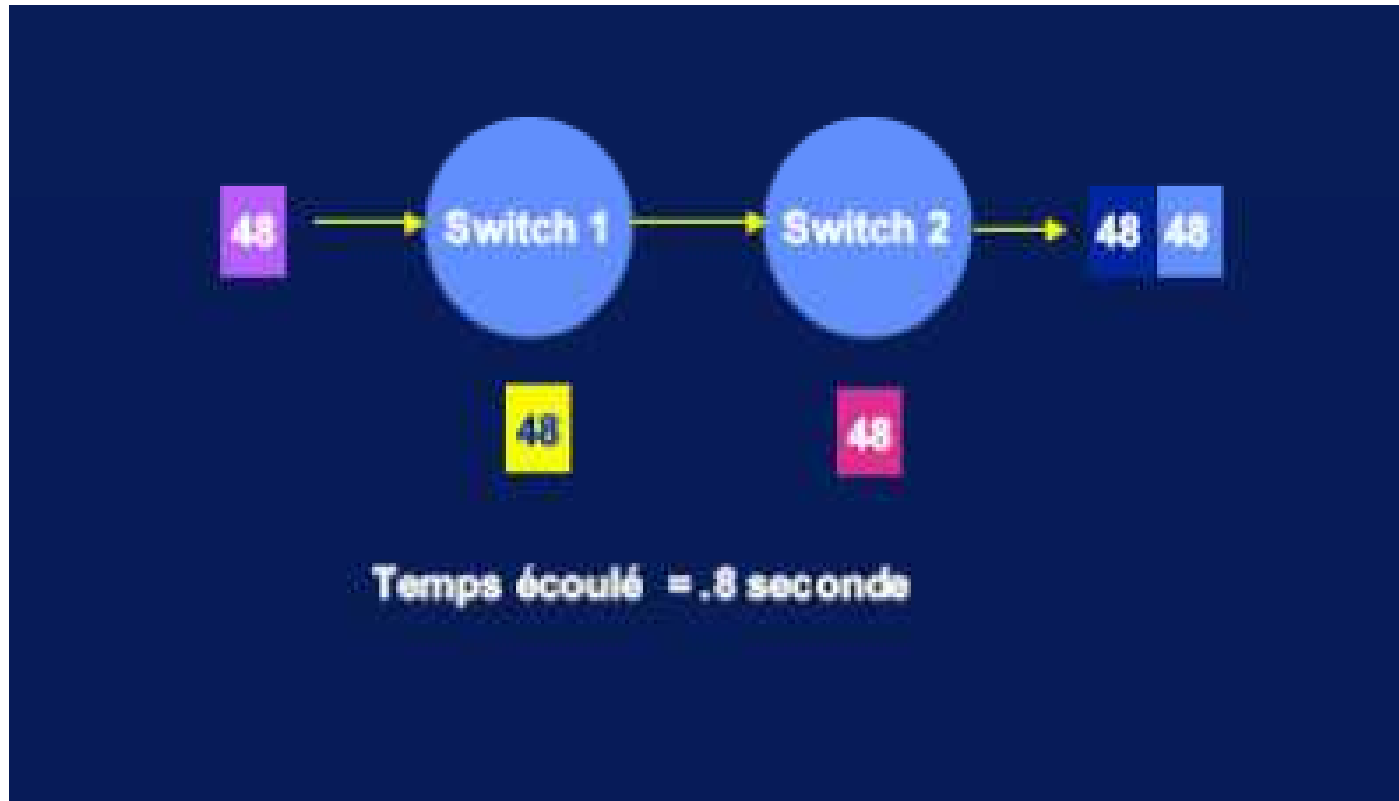
Réseaux ATM



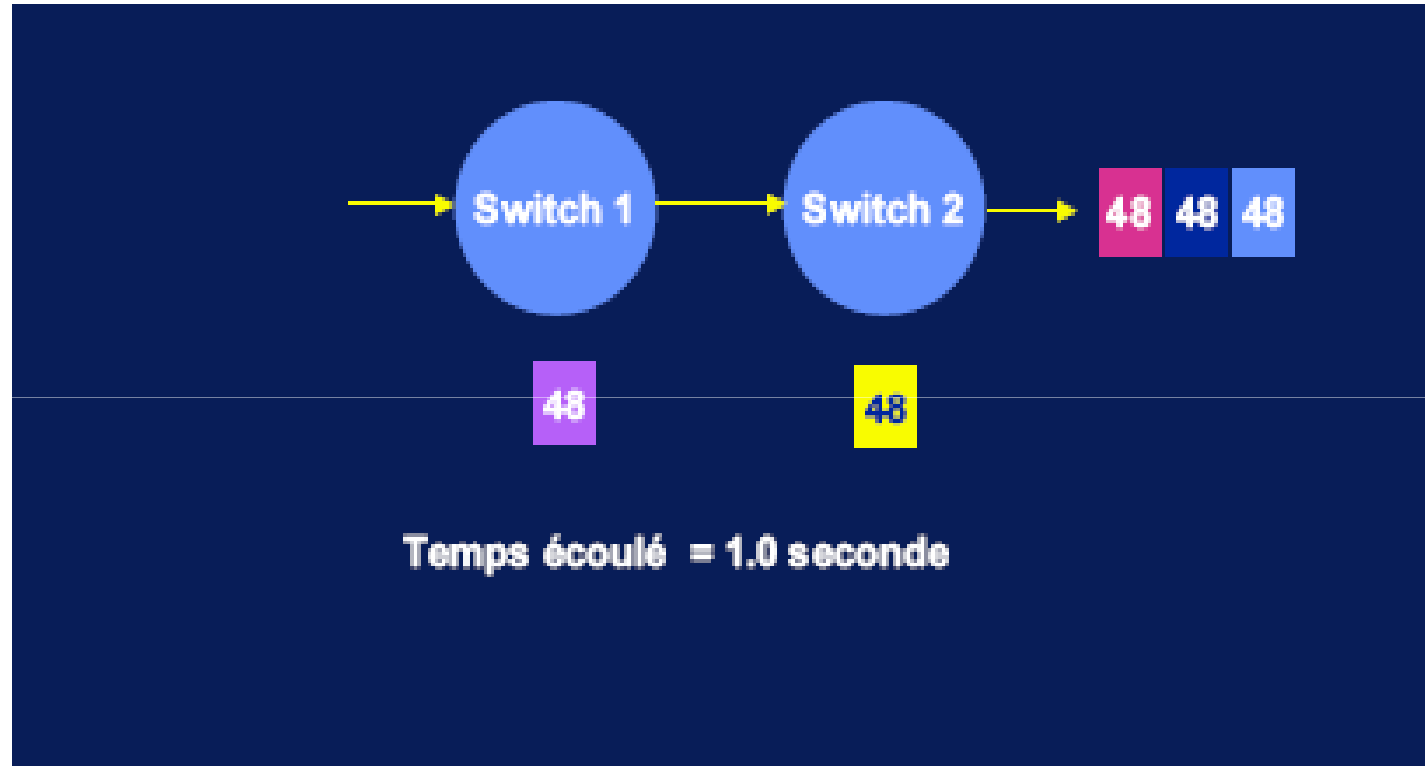
Réseaux ATM



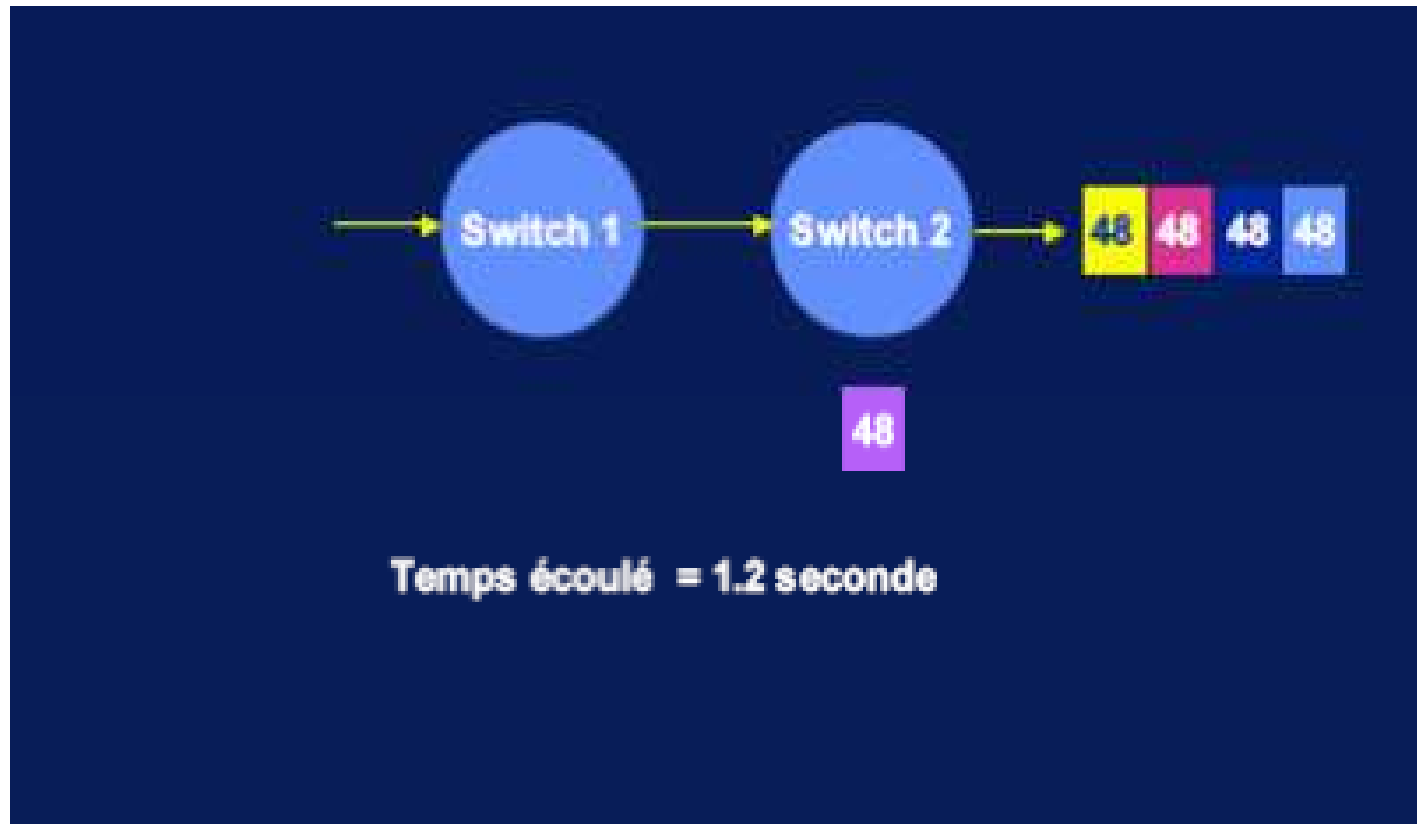
Réseaux ATM



Réseaux ATM



Réseaux ATM



Réseaux ATM



Temps écoulé = 1.4 seconde

Réseaux ATM

Fonctions de base

- Paquets de taille fixe , longueur 53 octets (Cellule)
 - ➔ Meilleure utilisation des supports
- Réseau orienté connexion
 - ➔ Amélioration des temps de transfert
- Négociation des services à l'établissement de la connexion
 - ➔ Réduction du temps de traitement des cellules
- Utilisation de supports de transmission fiable (fibre optique)
 - ➔ pas de détection ni correction d'erreurs nécessaires

Réseaux ATM

Format de cellule

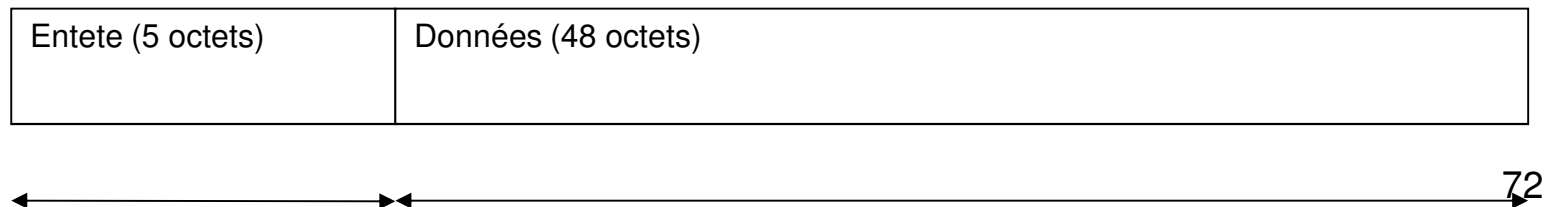
Taille fixe 53 octets (5 d'entête et 48 de données)

Raison du choix pour la longueur des données

Proposition américaine: 64 octets

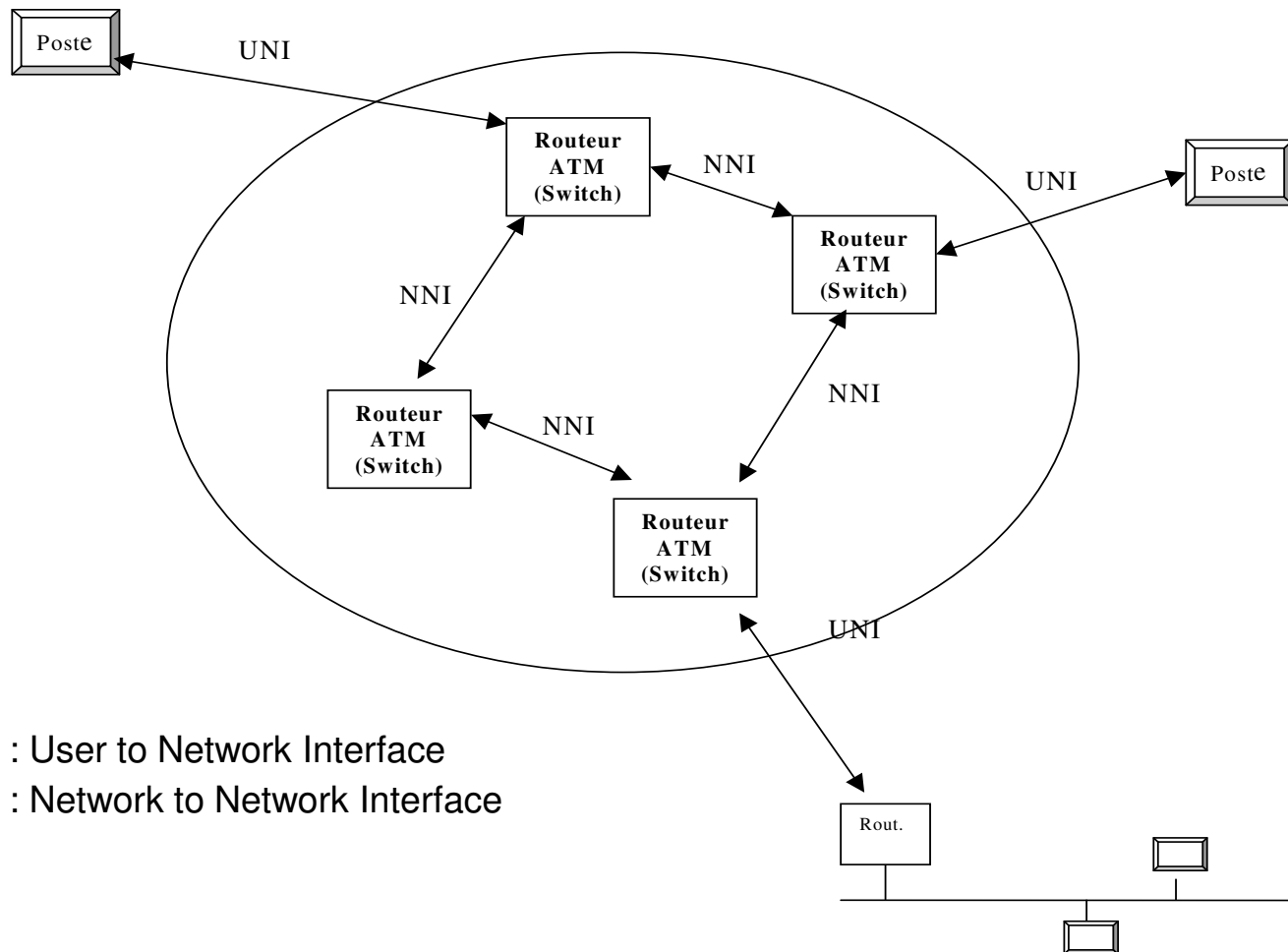
Proposition européenne: 32 octets

➔ Choix final 48 octets



Réseaux ATM

Le réseau ATM

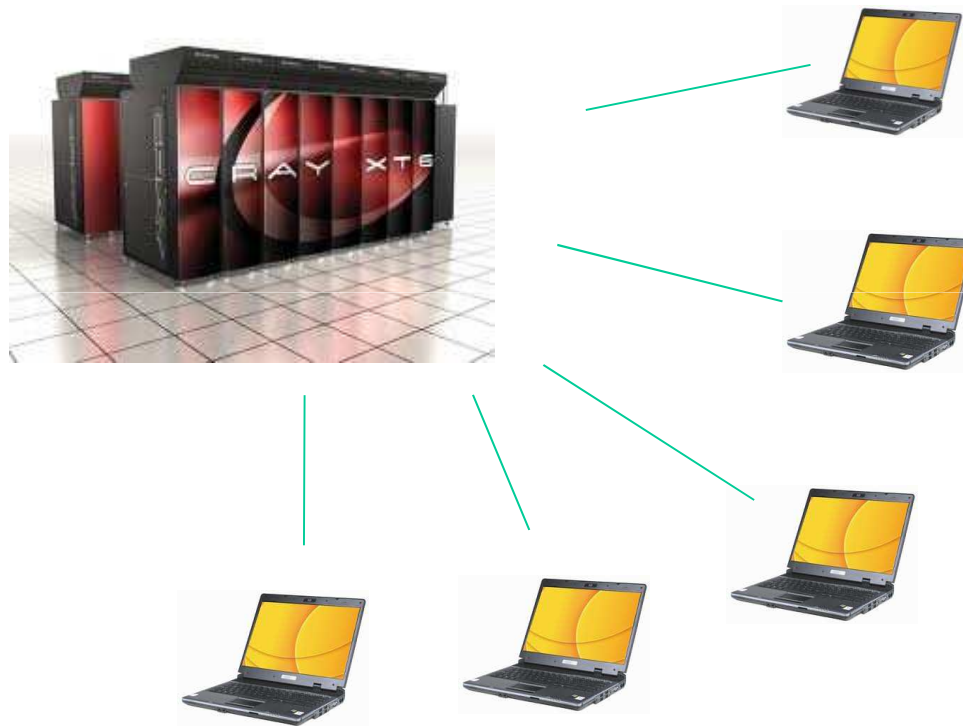


Les couches applicatives

5 à 7 du modèle OSI

Introduction

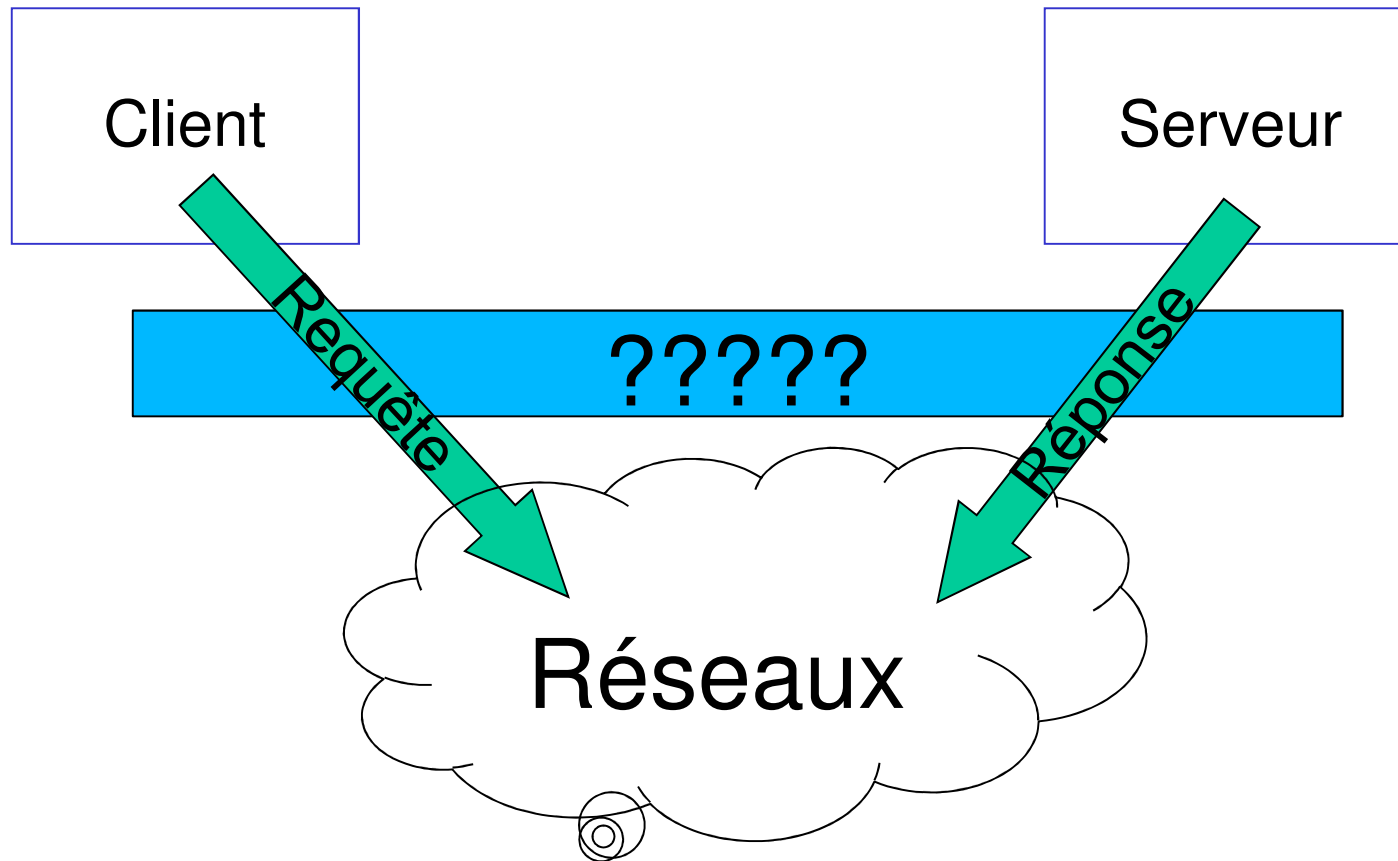
L'objectif des réseaux est de mettre en relation des machines afin que ces dernières se partagent leurs ressources. Il faut, pour cela, créer des applications, capables de dialoguer entre elles.



On parle d'application Client/Serveur ou de services

Introduction

Ces applications doivent fonctionner via un réseau déjà existant.

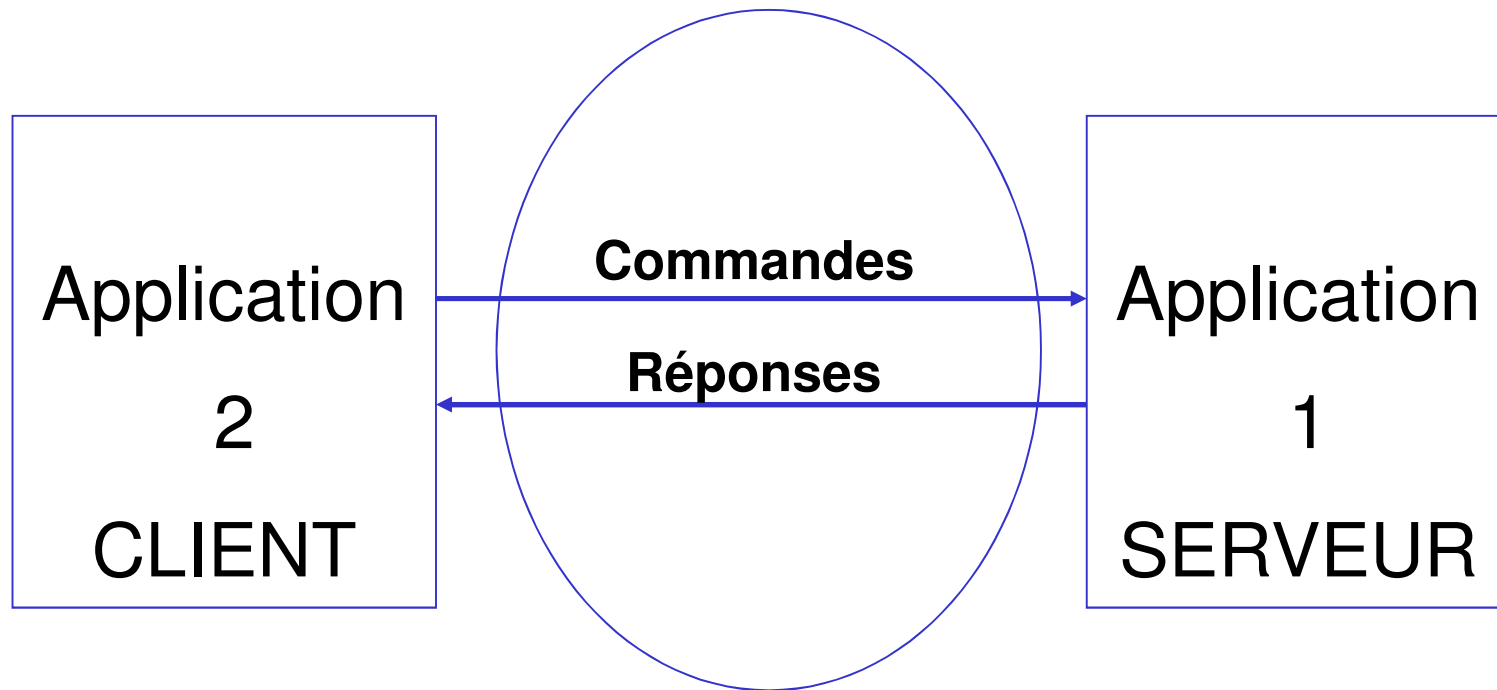


Comment interfacer les applications et les couches réseau ?

→ Interface Socket

Introduction

Ces applications doivent « dialoguer » en parfaite synchronisation. Pour cela l'une attendra les requêtes et l'autre les lui transmettra puis inversement.



Quelle doit être la structure des applications ?

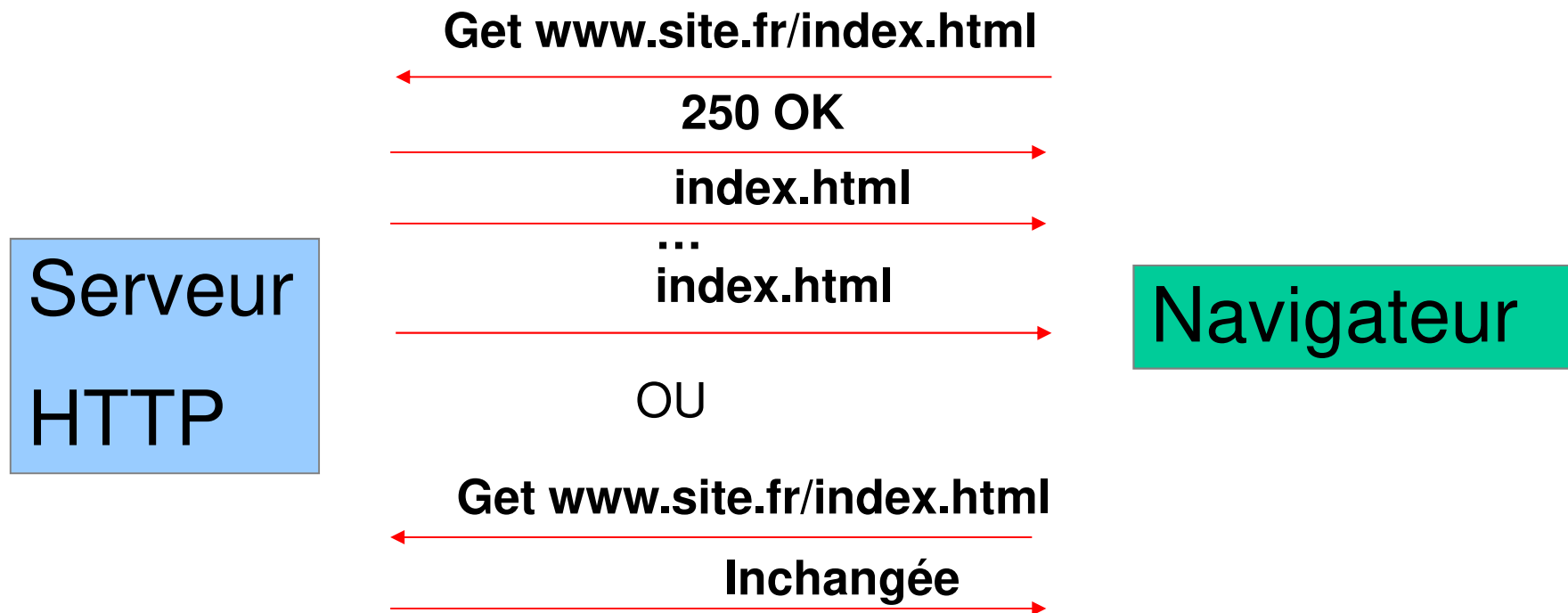
Quels doit être le format des données?

→ **RFC**

Introduction

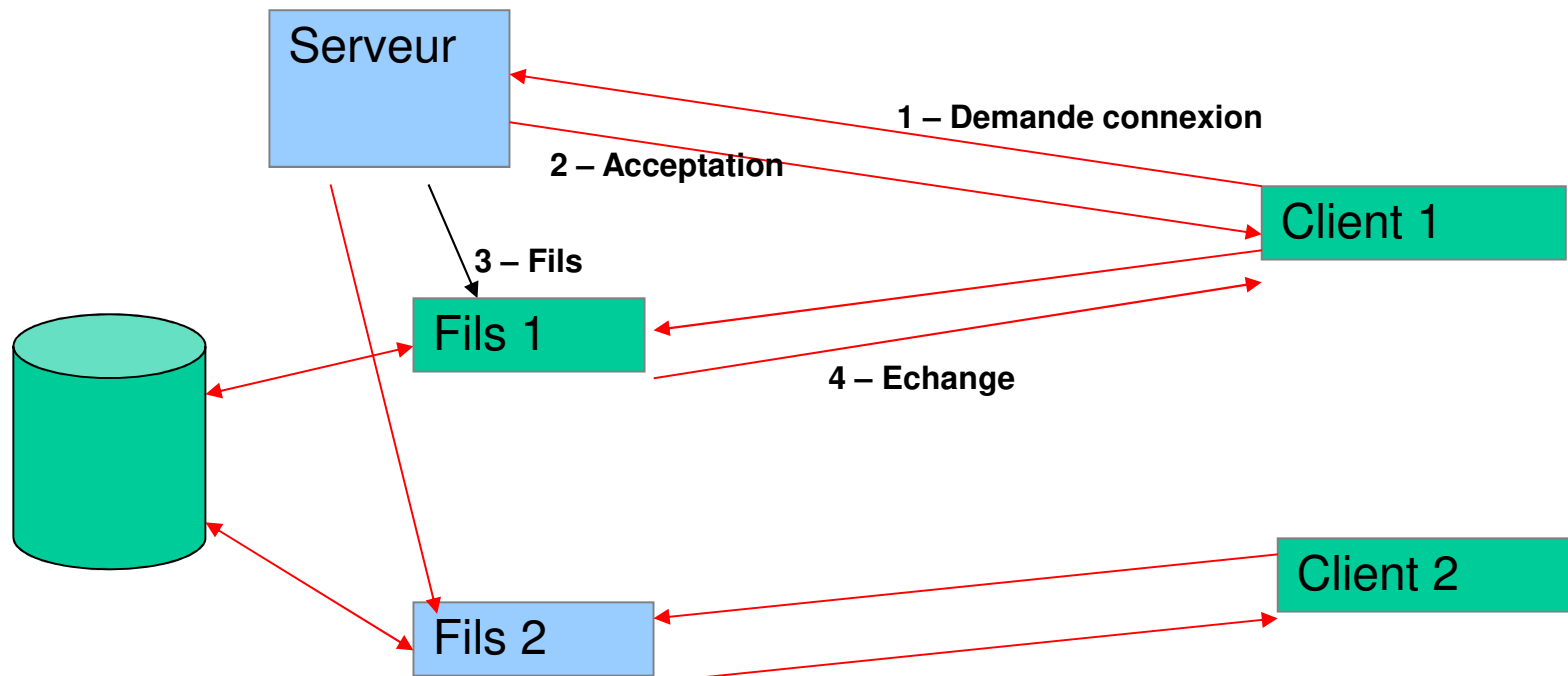
Exemple de dialogue - HTTP

Protocole de communication : HTTP 'HyperText Transfer Protocol'



Introduction

L'application serveur, doit communiquer avec plusieurs client simultanément.



Comment créer cette simultanéité ?
→ **Fork() – Thread**

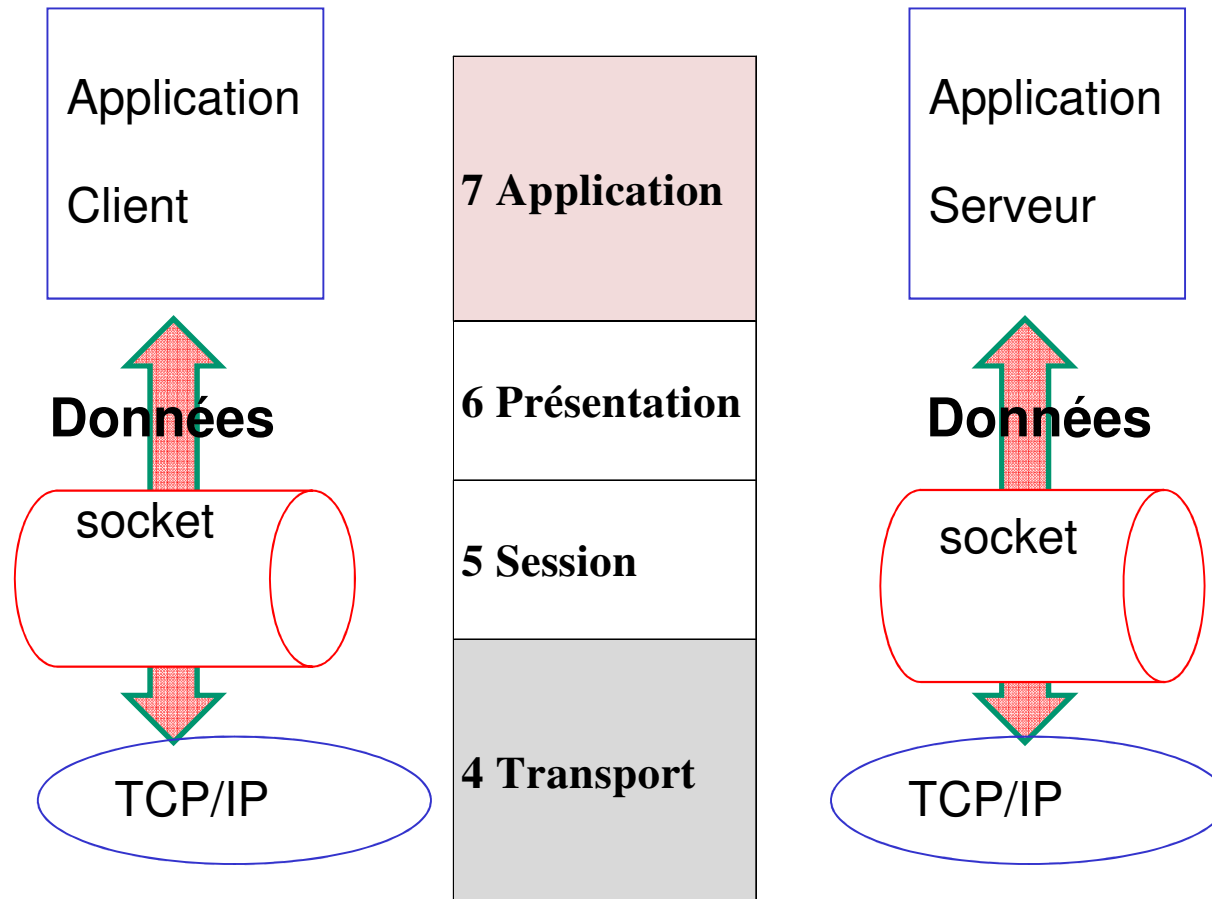
Introduction

Type de problème	Couche	Rôle	Norme
Applications réseaux	7 Application	Applications réseau	Http, Ftp, Ftam, X400
Accès aux ressources	6 Présentation	Format des données	Gestion de la syntaxe
	5 Session	Accès aux données	Sockets
Echange via plusieurs réseaux	4 Transport	Transport et contrôle de routage	TCP, UDP
	3 Réseau	Routage des paquets dans plusieurs réseaux	X25, IP
Echange entre 2 machines	2 Liaison	Contrôle de l'échange entre deux machines	HDLC, IEEE 802.x
	1 Physique	Transmission de signaux binaires	Ethernet,

Toutes ces fonctions correspondent aux couches 5,6 et 7 du modèle OSI.

Couche applicative

Le modèle OSI



Spécification des applications

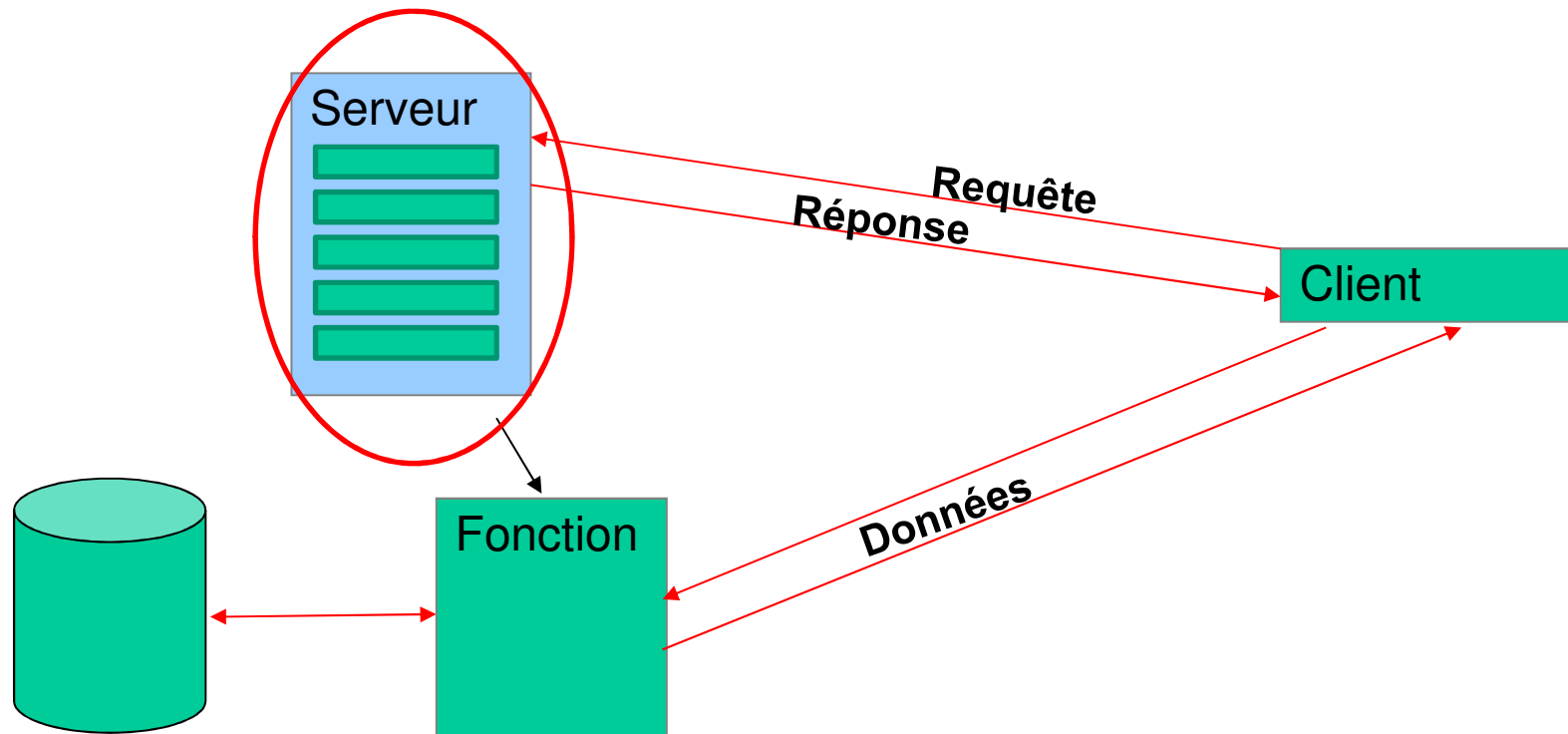
Notion de RFC

Spécification des applications

1 - Définir les fonctions accessibles (Méthodes)

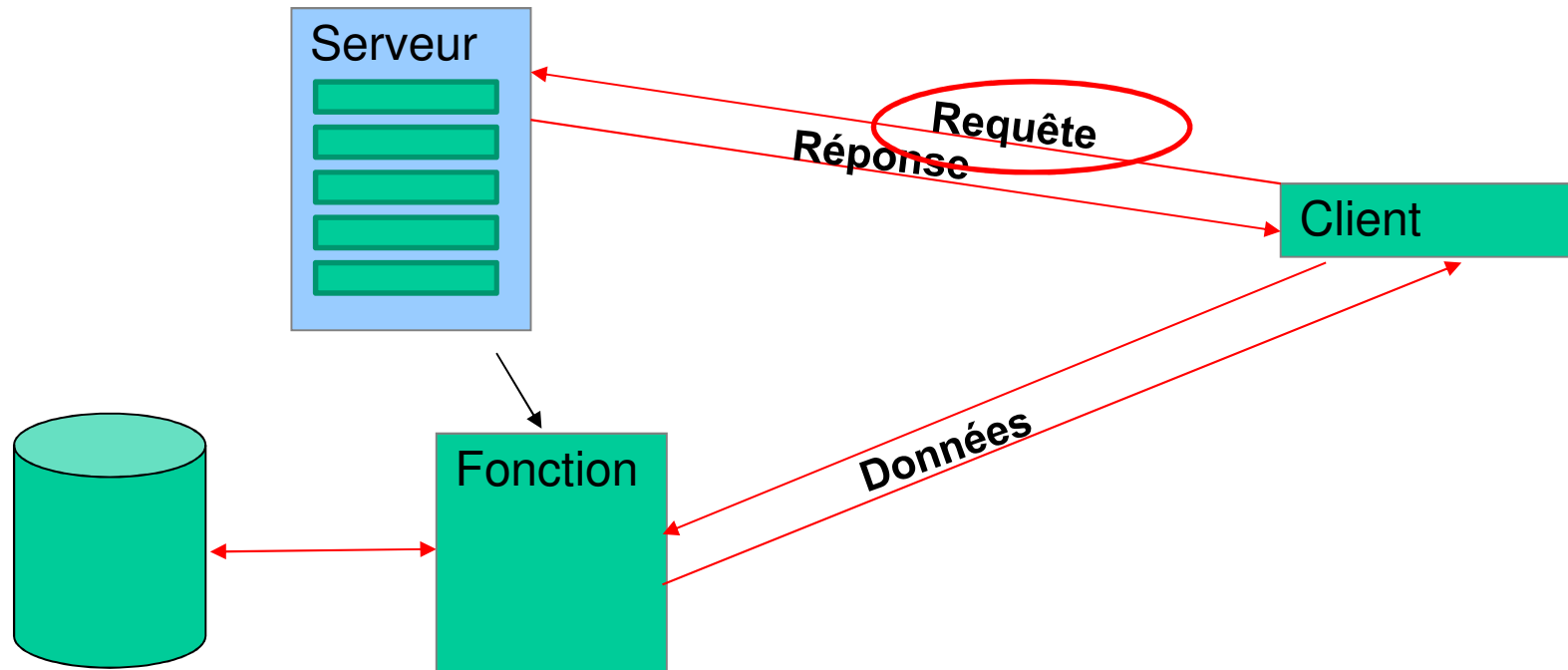
Les processus interagissent par échange de messages :

- Chaque message reçu doit déclencher une **fonction** particulière du processus serveur
- Représentation des fonctions = automates état/transition.



Spécification des applications

2 - Définir les conditions déclanchantes (Requêtes)

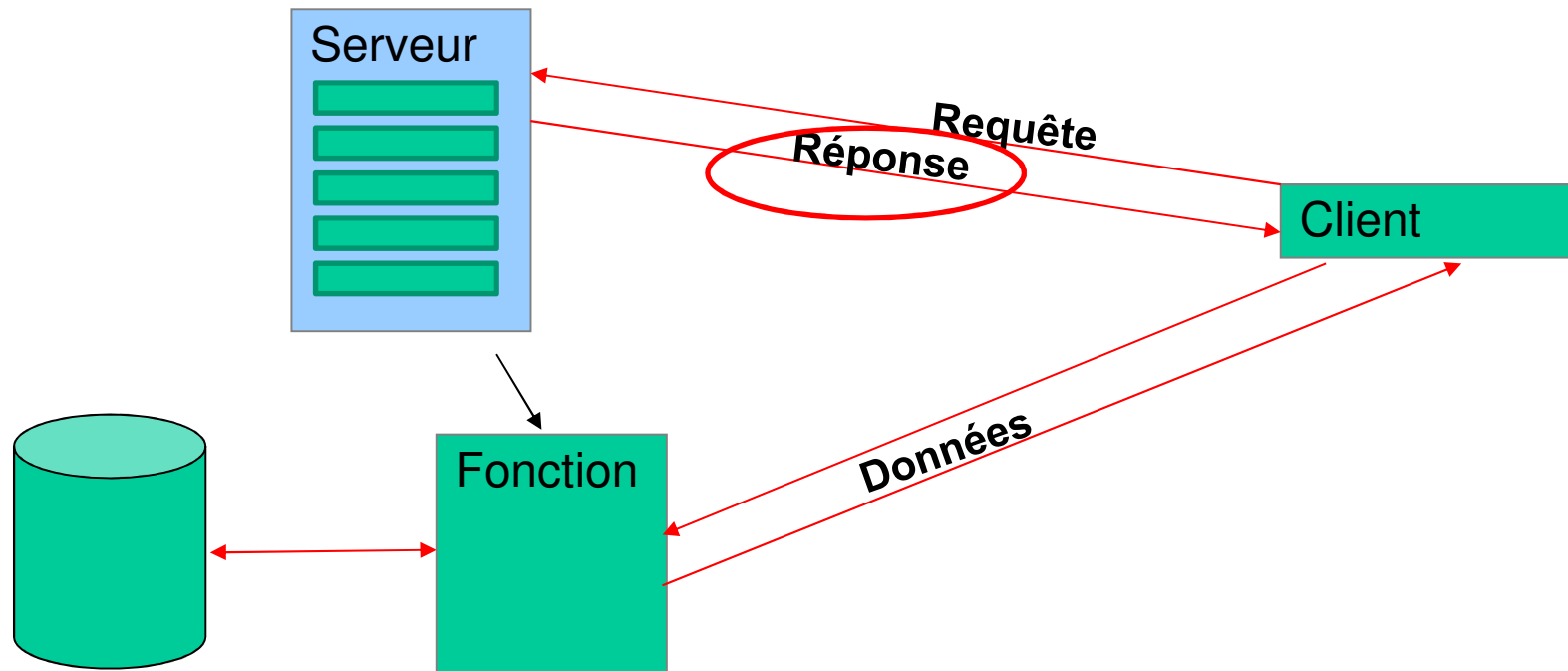


Requêtes = liste des unités de service (primitives, SDU) échangées entre le prestataire et l'utilisateur.

→ Exemple de primitives ou SDU : DELE, CWD, HELP, QUIT, STAT, RMD, MKD, PWD, TYPE, ...

Spécification des applications

3 - Définir les réponses possibles aux requêtes

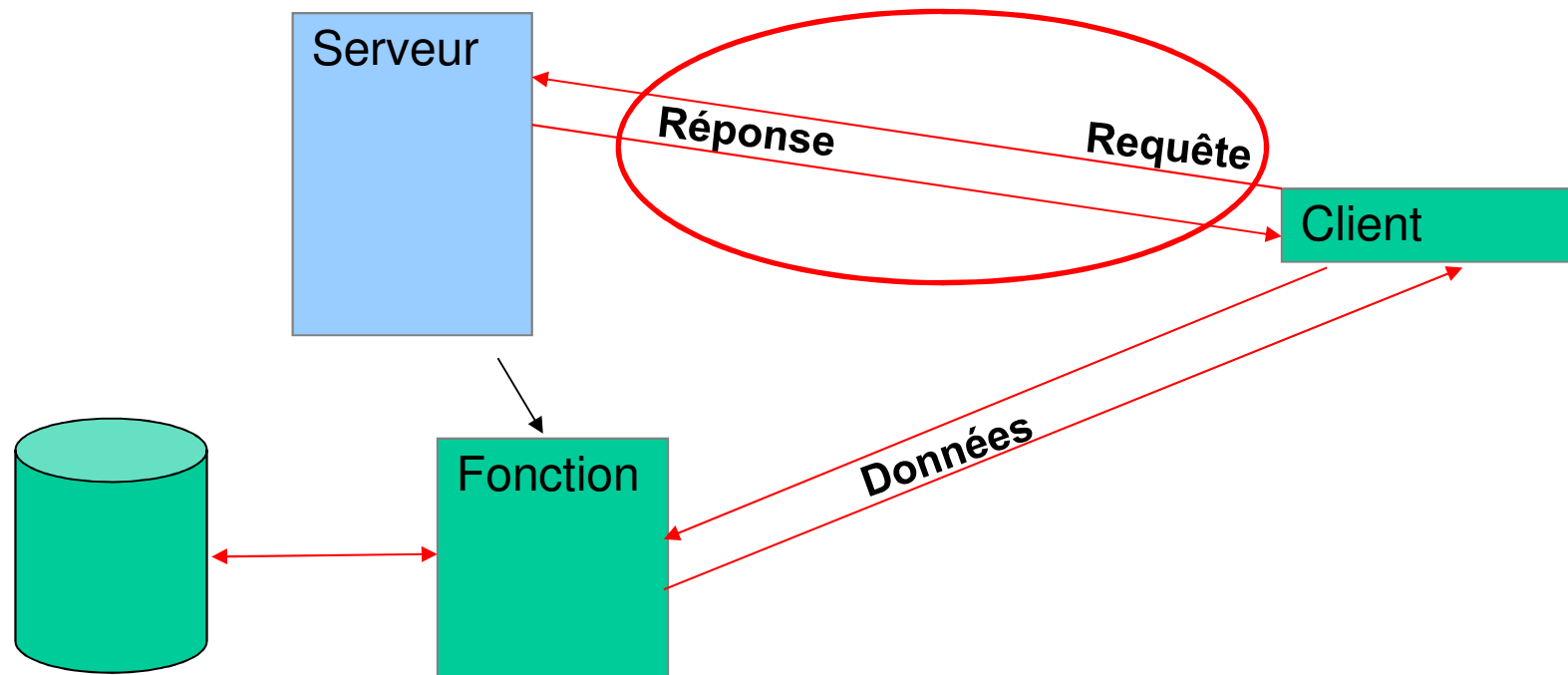


Chaque requête donne lieu à une réponse

- 1yz Réponse positive préliminaire
- 2yz Réponse positive définitive
- 3yz Réponse positive intermédiaire
- 4yz Réponse négative transitoire
- 5yz Réponse négative définitive

Spécification des applications

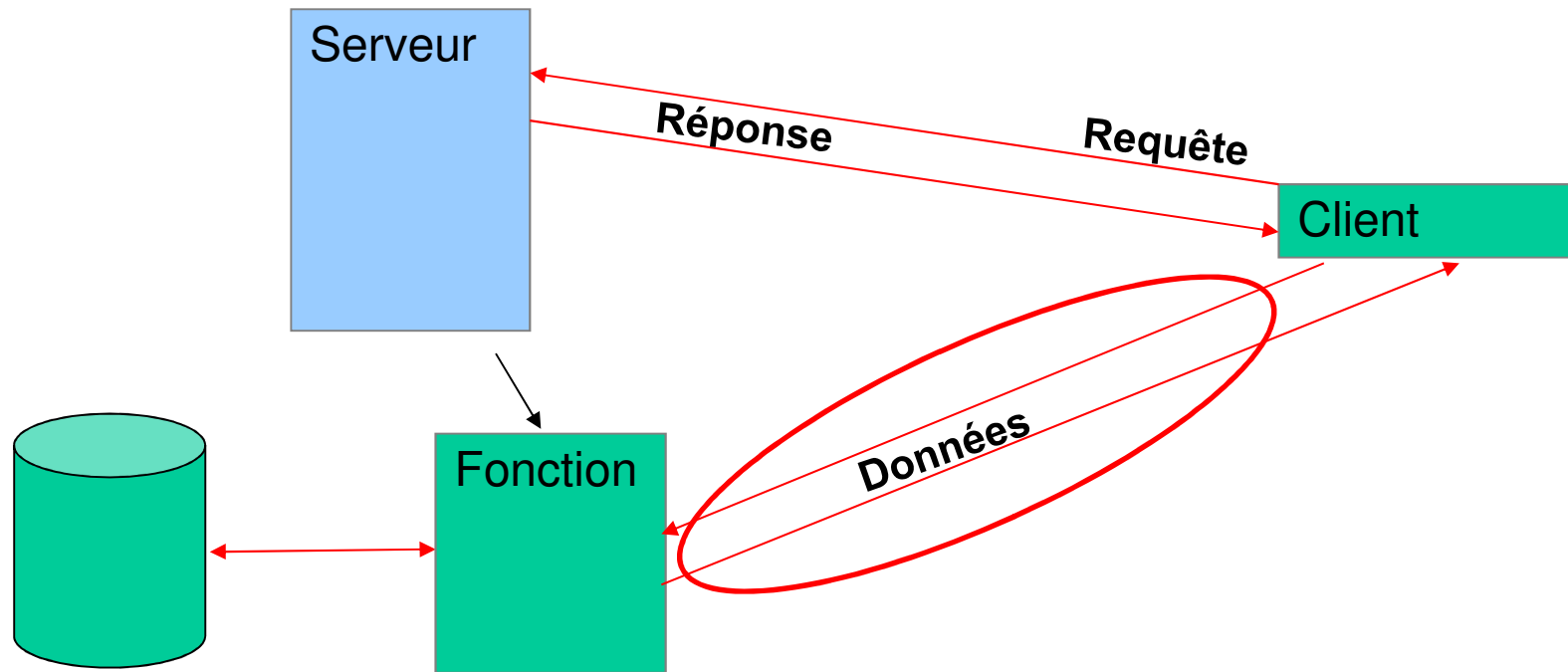
4 - Définir les échanges entre deux prestataires



- Établissement de la liste des unités de protocole (messages, PDU) échangés entre deux niveaux n.
 - Définition des suites d'échanges qu'un observateur de la voie de communication entre les deux niveaux n peut observer.
- => Idem que diagramme de séquence

Spécification des applications

5 - Définir le format des données échangées



Spécification des applications

Exemple d'échanges entre deux prestataires de même niveau
:

COMMANDES LOCALES (Utilisateur)	ACTION IMPLIQUEE
ftp (host) multics<CR>	Connexion à l'hôte S, port L, Etablissement du canal de contrôle. <---- 220 Service ready <CRLF>.
username Doe <CR>	USER Doe<CRLF>----> <---- 331 User name ok, need password <CRLF>.
password mumble <CR>	PASS mumble<CRLF>----> <---- 230 User logged in<CRLF>.

Tout ceci est décrit dans les RFC

Spécification et RFC

Un RFC est une série numérotée de documents officiels décrivant les aspects techniques d'Internet (Wikipédia)

RFC 959 STD 9	FTP - File Transfer Protocol Protocole de la couche applicative. Utilisé pour le transfert fiable de fichiers sur Internet.
RFC 854 STD 8	TELNET - Protocole TELNET Protocole de la couche applicative. Utilisé pour se connecter à un serveur distant, Terminal Virtuel Internet.
RFC 830	DNS - Système réparti pour le service des noms Internet
RFC 827	EGP - Protocole de passerelle extérieure (EGP)
RFC 826 STD 37	ETHERNET - Protocole de résolution d'adresse Ethernet Conversion des adresses de protocole réseau en adresses Ethernet à 48 bits pour la transmission sur matériel Ethernet
RFC 821 STD 10	SMTP - Simple Mail Transfer Protocol Protocole de la couche applicative. Utilisé pour envoyer des e-mails par les logiciels de messagerie électronique (KMail, Messenger, etc.)
RFC 819	DNS - Convention de désignation de domaine pour les applications d'utilisateur de l'Internet

Spécification et RFC

Les RFC sont rédigées sur l'initiative d'experts techniques, puis sont revues par la communauté Internet dans son ensemble.

Network Working Group
Request for Comments: 2616
Obsoletes: 2068
Category: Standards Track

R. Fielding
UC Irvine
J. Gettys
Compaq/W3C
J. Mogul
Compaq
H. Frystyk
W3C/MIT
L. Masinter
Xerox
P. Leach
Microsoft
T. Berners-Lee
W3C/MIT
June 1999

Plusieurs liens existent :

<http://www.rfc.fr/>

<http://abcdrfc.free.fr/>

<http://www.ietf.org/rfc.html>

Hypertext Transfer Protocol -- HTTP/1.1

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

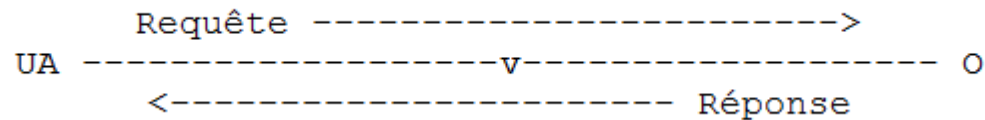
Spécification et RFC

Dans les RFC on trouve :

1 – La façon d'accéder au service (URL)

```
http_URL          = "http:" "://" host [ ":" port ] [ chem_abs ]  
  
host              = <un nom Internet d'ordinateur valide ou une adresse IP  
                  (sous forme numérique), comme définie en Section 2.1 de la  
                  RFC 1123>  
  
port              = *DIGIT
```

2 – La structure générale des échanges



Spécification et RFC

3 – La liste et la description des méthodes

Méthode	= "GET"	; Section 8.1
	"HEAD"	; Section 8.2
	"POST"	; Section 8.3
	<i>nom_de_méthode</i>	

8.1 GET

La méthode GET signifie "récupérer" le contenu quel qu'il soit de la ressource (sous forme d'une entité) identifiée par l'URI-visée. Si l'URI-visée identifie un processus générant dynamiquement des données, ce sont les données produites qui sont renvoyées dans l'entité au lieu du source de l'exécutable appelé, sauf si ce texte lui-même est la sortie du processus.

Spécification et RFC

4 – La liste et la description des réponses

Code_état	= "200" ; OK	OK
	"201" ; Created	Créé
	"202" ; Accepted	Accepté
	"204" ; No Content	Pas de contenu
	"301" ; Moved Permanently	Changement définitif
	"302" ; Moved Temporarily	Changement temporaire
	"304" ; Not Modified	Non modifié
	"400" ; Bad Request	Requête incorrecte
	"401" ; Unauthorized	Non autorisé
	"403" ; Forbidden	Interdit
	"404" ; Not Found	Non trouvé
	"500" ; Internal Server Error	Erreur interne serveur
	"501" ; Not Implemented	Non implémenté
	"502" ; Bad Gateway	Erreur de routeur
	"503" ; Service Unavailable	Indisponible

9.2 Succès 2xx

Cette classe précise que la requête du client a été correctement transmise, interprétée, et exécutée.

200 OK

La requête a abouti. L'information retournée en réponse dépend de la requête émise, comme suit:

Spécification et RFC

5 – Le format détaillé des requêtes et des réponses

Requête_complète	= Ligne_requête	; Section 5.1
	*(En-tête_générale	; Section 4.3
	En-tête_requête	; Section 5.2
	En-tête_entité)	; Section 7.1
	CRLF	
	[Corps_entité]	; Section 7.2
Réponse_complète	= Ligne-état	; Section 6.1
	*(En-tête_générale	; Section 4.3
	En-tête_réponse	; Section 6.2
	En-tête_entité)	; Section 7.1
	CRLF	
	[Corps entité]	; Section 7.2

Spécification et RFC

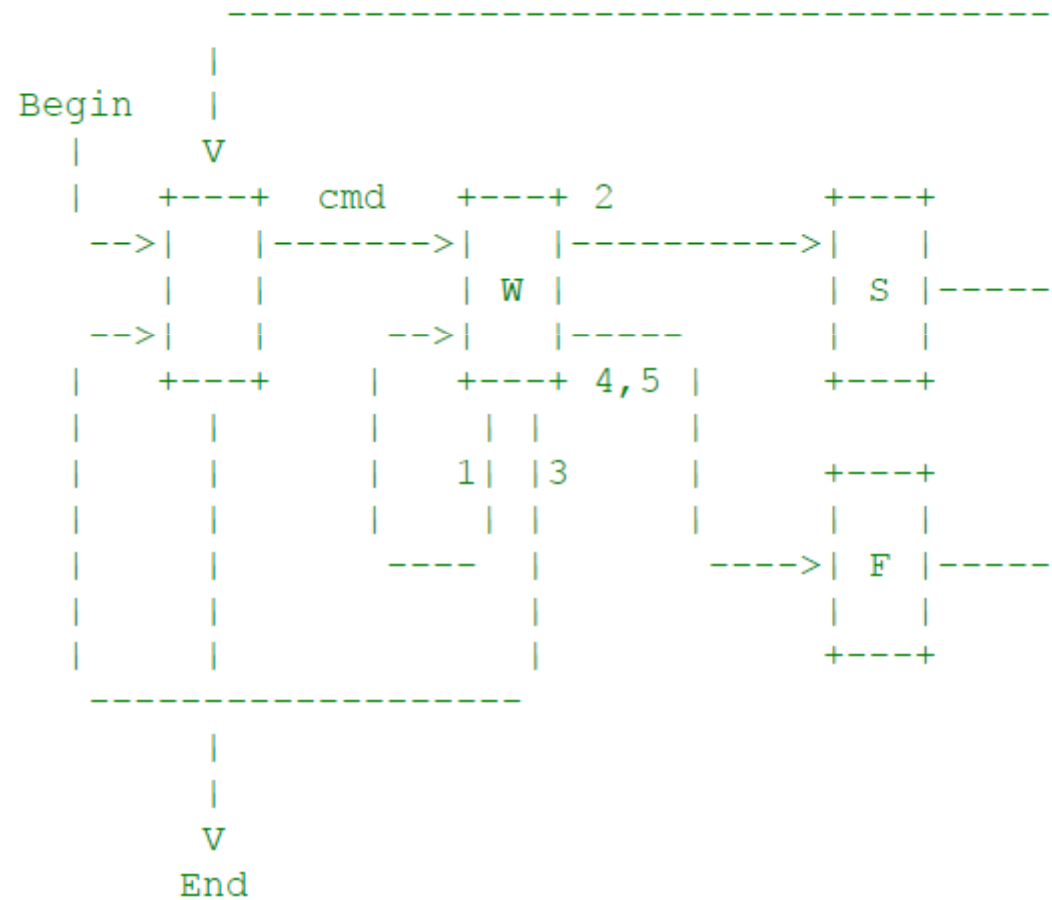
6 – Le format des données

Descripteur	Compte
code=16	= 6

Marqueur	Marqueur	Marqueur
8 bits	8 bits	8 bits

Spécification et RFC

7 – Les échanges types – Diagrammes d'états



Spécification et RFC

En résumé

Un RFC est un document de référence sur lequel s'appuient tous les développeurs d'applications réseaux (client-serveur).

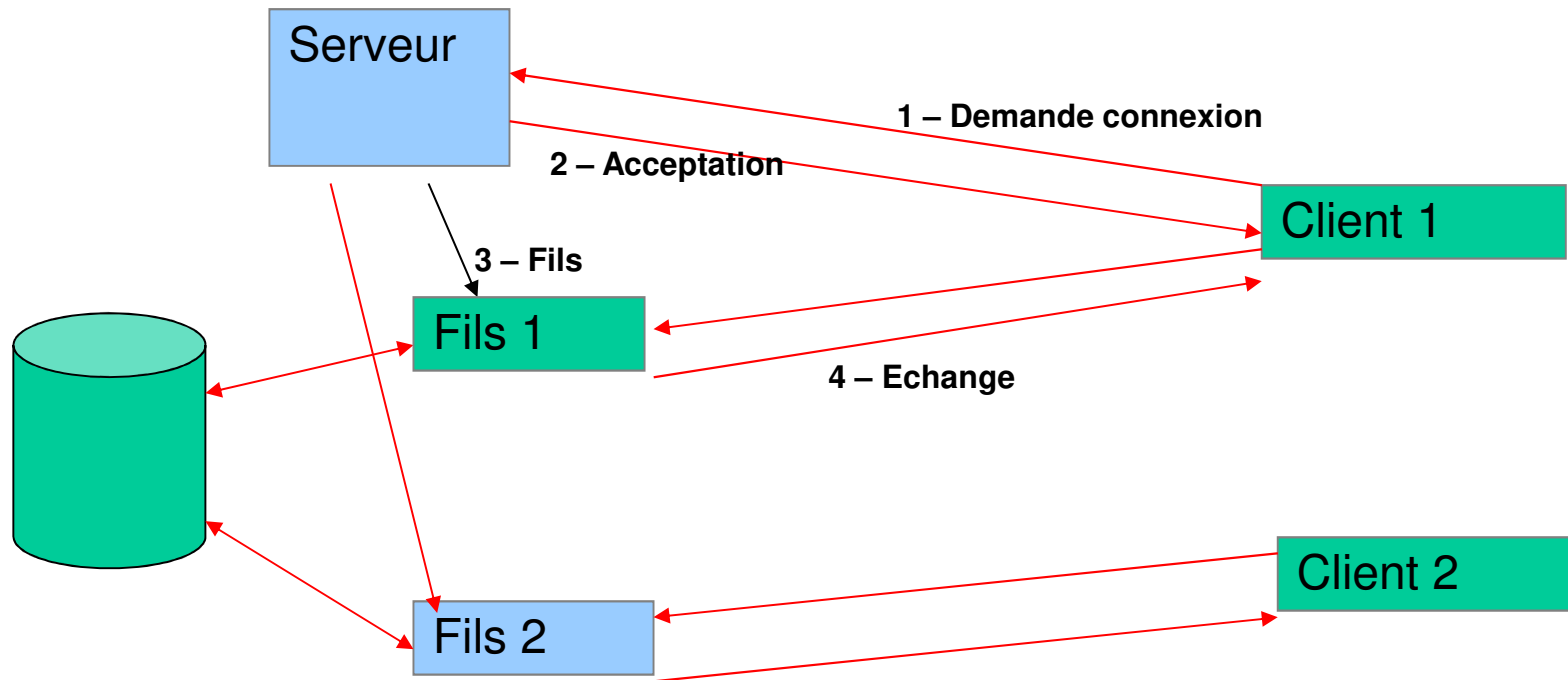
Il donne toutes les spécifications pour pouvoir développer, n'importe où dans le monde, des applications compatibles.

En cas de non respect des spécifications, des erreurs peuvent survenir.

Ces erreurs peuvent devenir des failles ...

Gestion des processus

Rappels



Comment créer des applications client/serveur ?

Gestion des processus

LA CREATION DES PROCESSUS

Les processus sont créés de différentes façons :

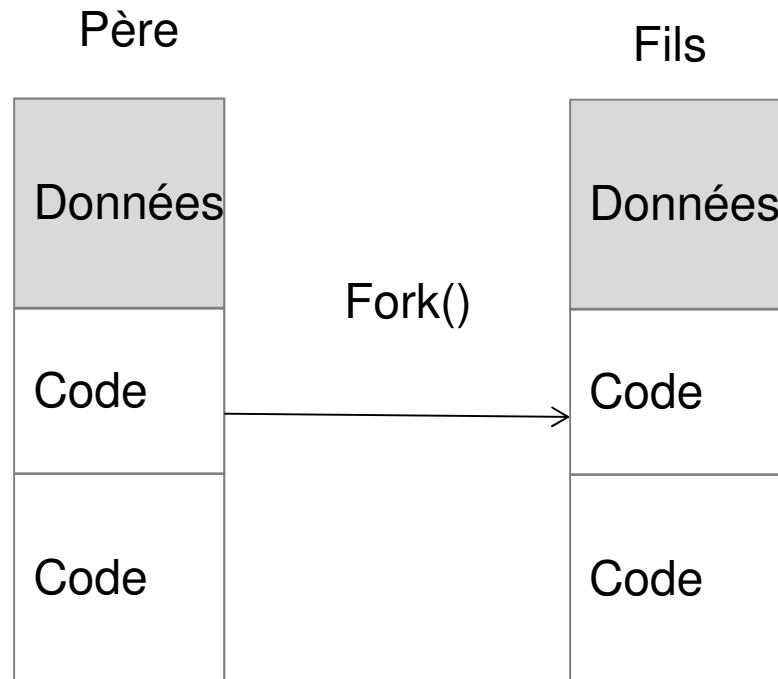
- Clic sur un icône (mode graphique)
- Saisie du nom du programme au clavier (mode texte)
- Exécution via un programme par des fonctions particulières (exemple : `fork()`)

Gestion des processus

LA CREATION DES PROCESSUS – FORK()

La primitive `fork()` permet la création dynamique d'un nouveau processus.

Le processus qui exécute le `fork()` est nommé PÈRE, celui qui est créé FILS. Le processus FILS est une copie exacte du PÈRE (code et données).



Gestion des processus

LA CREATION DES PROCESSUS – FORK()

Il sera nécessaire de dupliquer autant de fois que nécessaire, les instructions chargées du dialogue avec le client, juste après la connexion.

```
while(1)
{ socket2=accept(sock, (struct sockaddr *)&distant, &lg);
  {
    ....
    // instructions chargées du dialogue avec le client
    ....
    close(socket);
  }
}
```

Instructions à
dupliquer



Gestion des processus

LA CREATION DES PROCESSUS – FORK()

Deux techniques sont possibles pour cela :

- La duplication du processus via une fonction : fork(),
- La création d'un Thread, contenant les instructions du dialogue.

Exemple :

```
while(1)
{ socket2=accept(sock, (struct sockaddr *)&distant, &lg);
  if (fork()==0)
  {....
    // instructions chargées du dialogue avec le client
    ....
    close(socket);
  }
}
```

Gestion des processus

LES PROCESSUS ET LIMITES

Chaque processus créé occupe une place dans la table des processus.

La table des processus est limitée en taille.

Limite : si trop de processus , le système refuse l'exécution des nouveaux.

Dans un contexte normal (hors réseau) cela n'arrive quasiment jamais.

Gestion des processus

LES PROCESSUS ET LIMITES

Dans un contexte réseau, si on utilise des applications « SERVEUR », la table peut se remplir très vite.

Raisons :

- Chaque client qui se connecte déclenche un nouveau processus (fork())
- L'application serveur est conçue pour répondre à tous les clients, sans se soucier de la taille de la table des processus.