

Chapitre 6

La couche liaison

(Partie 2)

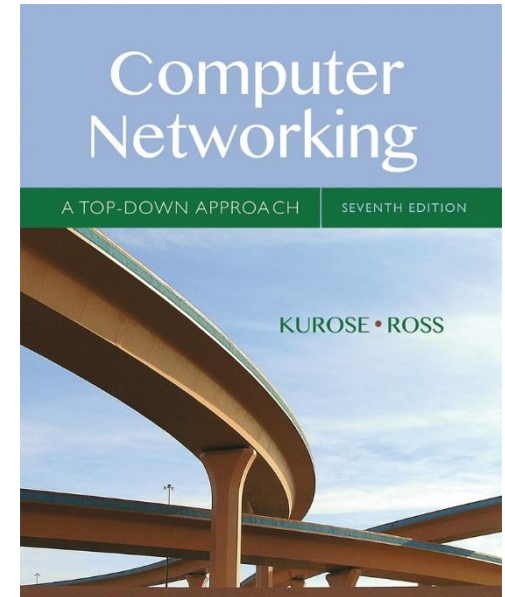
A note on the use of these Powerpoint slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a lot of work on our part. In return for use, we only ask the following:

- *If you use these slides (e.g., in a class) that you mention their source (after all, we'd like people to use our book!)*
- *If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.*

Thanks and enjoy! JFK/KWR

© All material copyright 1996-2016
J.F Kurose and K.W. Ross, All Rights Reserved



**Computer
Networking: A Top
Down Approach**
7ème édition
Jim Kurose, Keith Ross
Addison-Wesley
2017

Couche liaison: plan

5.1 introduction, services

5.2 détection d'erreurs,
correction

5.3 protocoles d'accès
multiple

5.4 LANs

- adressage, ARP
- Ethernet
- commutateurs
- VLANs

Adresse MAC

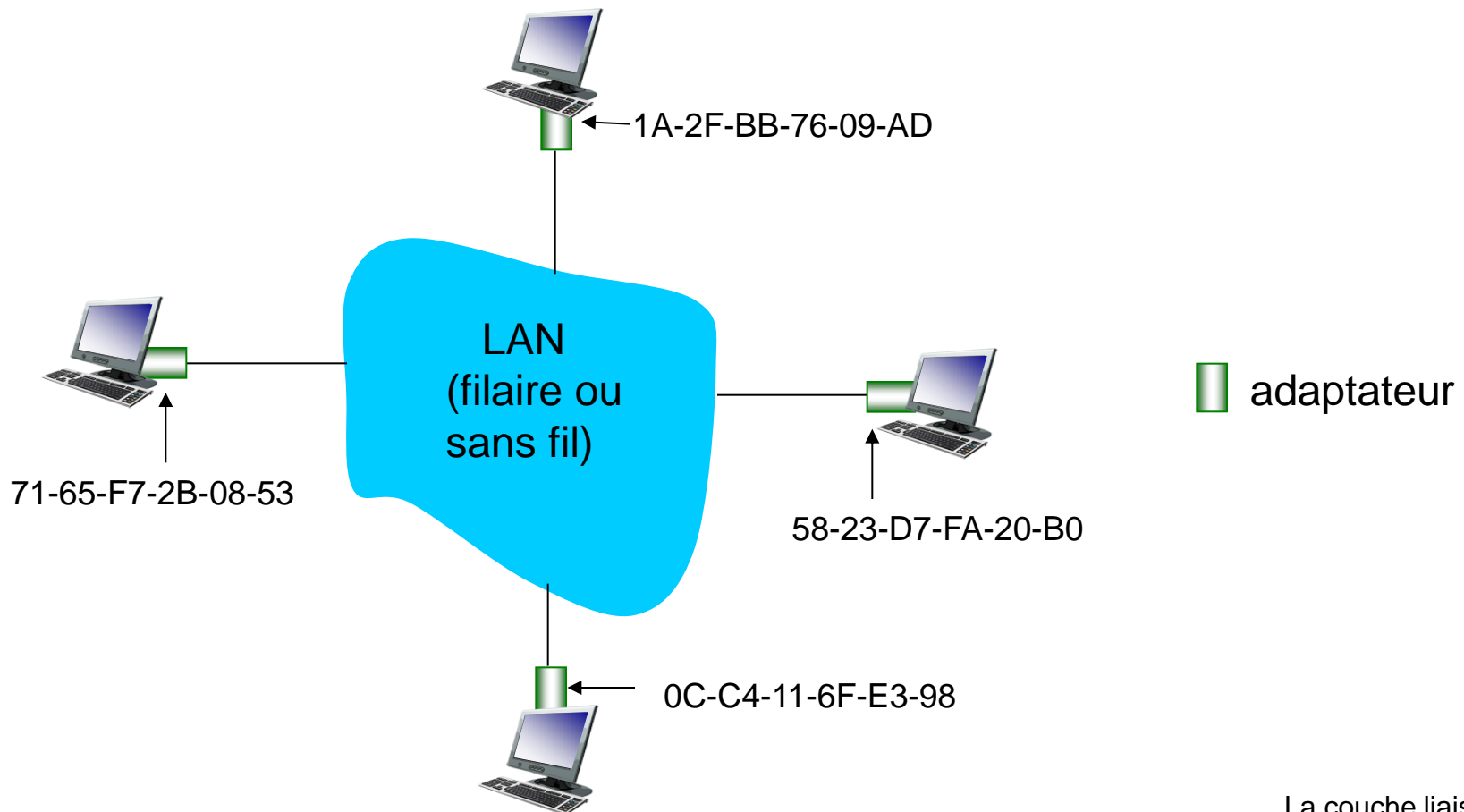
- ❖ adresse MAC (ou LAN ou physique ou Ethernet):
 - fonction: *utilisée ‘localement’ pour transmettre une trame d’une interface vers une autre interface connectée directement par un lien physique*
 - taille de 48 bits écrite dans la ROM de la NIC
 - ex.: 1A-2F-BB-76-09-AD

/

hexadécimal (base 16)
(chaque “2 numéros” représente un octet)

Adresses MAC

chaque adaptateur dans un LAN possède une adresse **MAC** unique



Adresses MAC

- ❖ l'allocation des adresses MAC est gérée par IEEE
- ❖ pour assurer l'unicité, les fabricants achètent des plages
- ❖ analogie:
 - adresse MAC : Numéro d'assurance social
 - adresse IP : adresse postale
- ❖ adresse MAC → portabilité
- ❖ adresse IP est hiérarchique *non* portable
 - dépend du sous-réseau auquel on est attaché

ARP: *address resolution protocol*

Question: comment connaître l'@MAC si on connaît l'@IP?

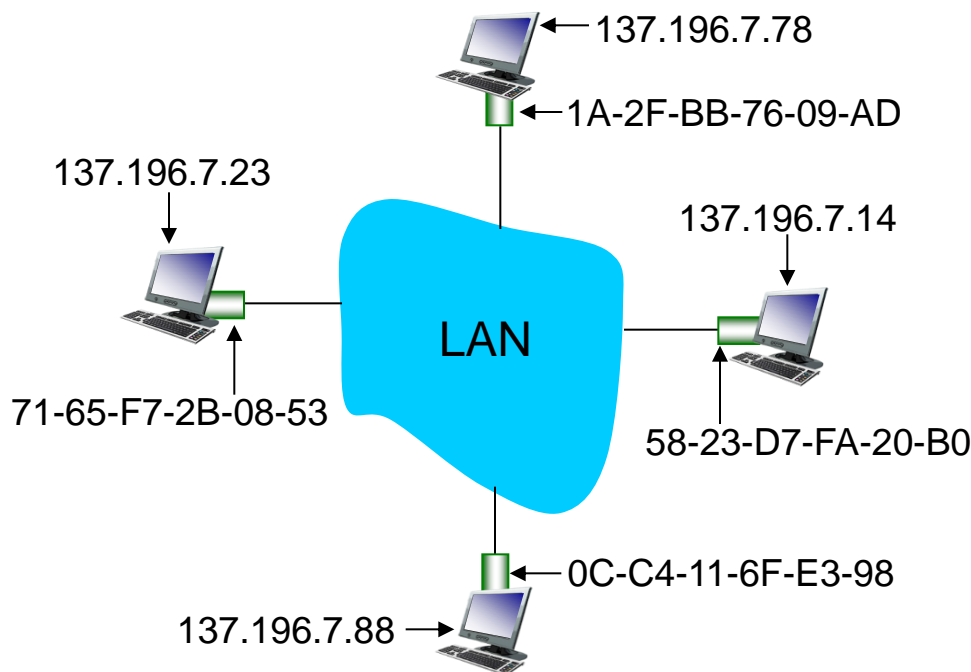


table ARP : chaque nœud possède une table qui contient

- les correspondances @IP/@MAC pour d'autres nœuds:
< adresse IP; adresse MAC; TTL >
- TTL (Time To Live): temps après lequel la correspondance est oubliée (typiquement 20 min)

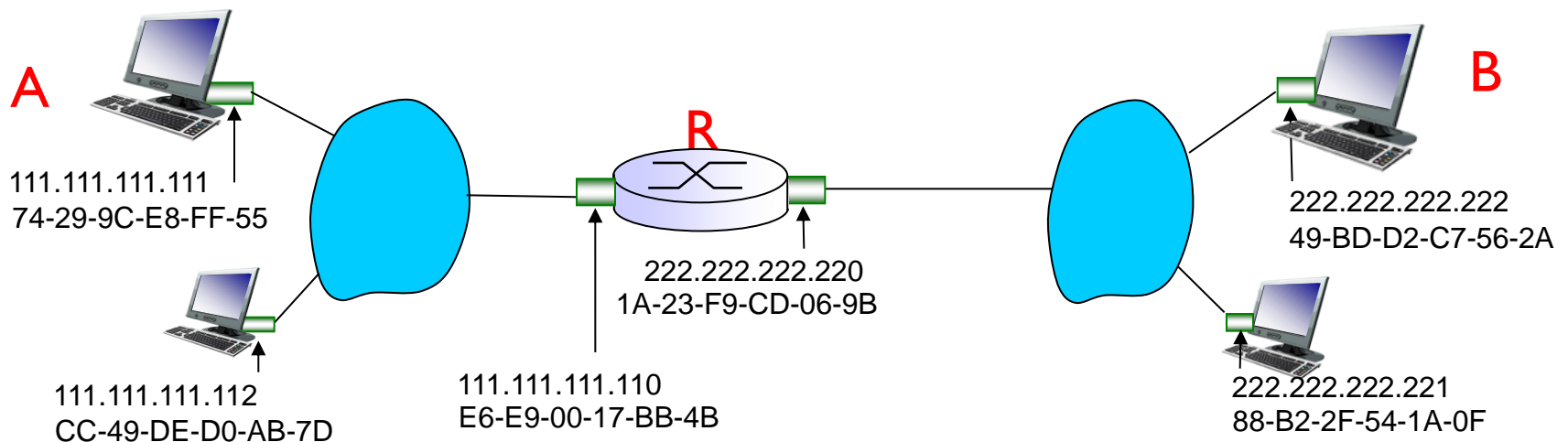
Protocole ARP : même LAN

- ❖ A veut envoyer un datagramme à B
 - @MAC de B n'est pas présente dans la table de A.
- ❖ A **diffuse** une requête ARP, contenant @IP de B
 - @MAC dest = FF-FF-FF-FF-FF-FF
 - tous les nœuds dans le LAN la reçoivent
- ❖ B reçoit la requête, répond à A en fournissant son @MAC (de B)
 - cette trame est envoyé vers A uniquement (unicast)
- ❖ A met la nouvelle correspondance @IP/@MAC dans sa table
- ❖ ARP est “plug-and-play”:
 - sans l'intervention d'un administrateur

Adressage: envoyer vers un autre LAN

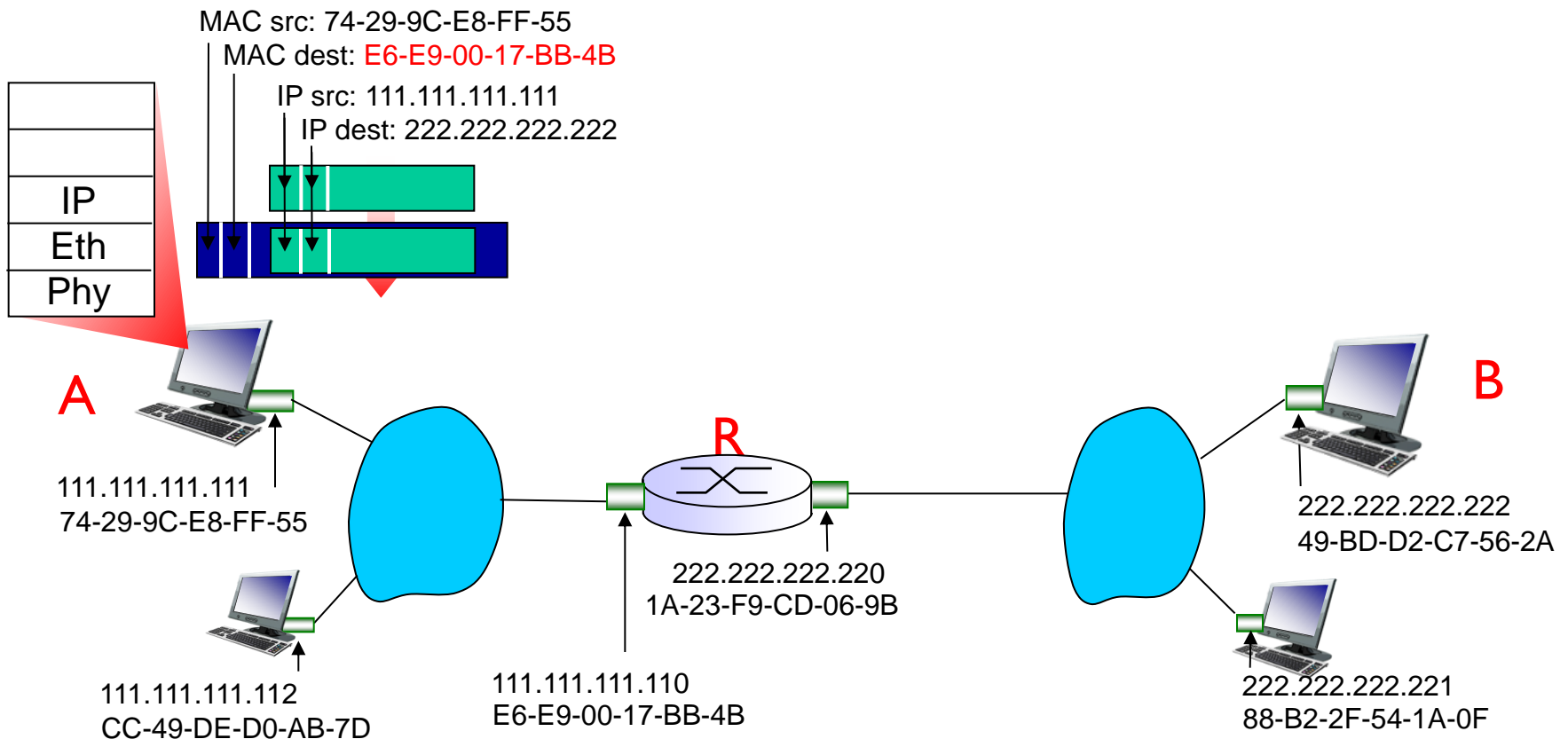
envoi d'un datagramme de A à B via R

- On s'intéresse à l'adressage des paquets
- supposons que A connaît @IP de B (Comment?)
- supposons que A connaît @IP du premier routeur R (Comment?)
- supposons que A connaît @MAC de R (Comment?)



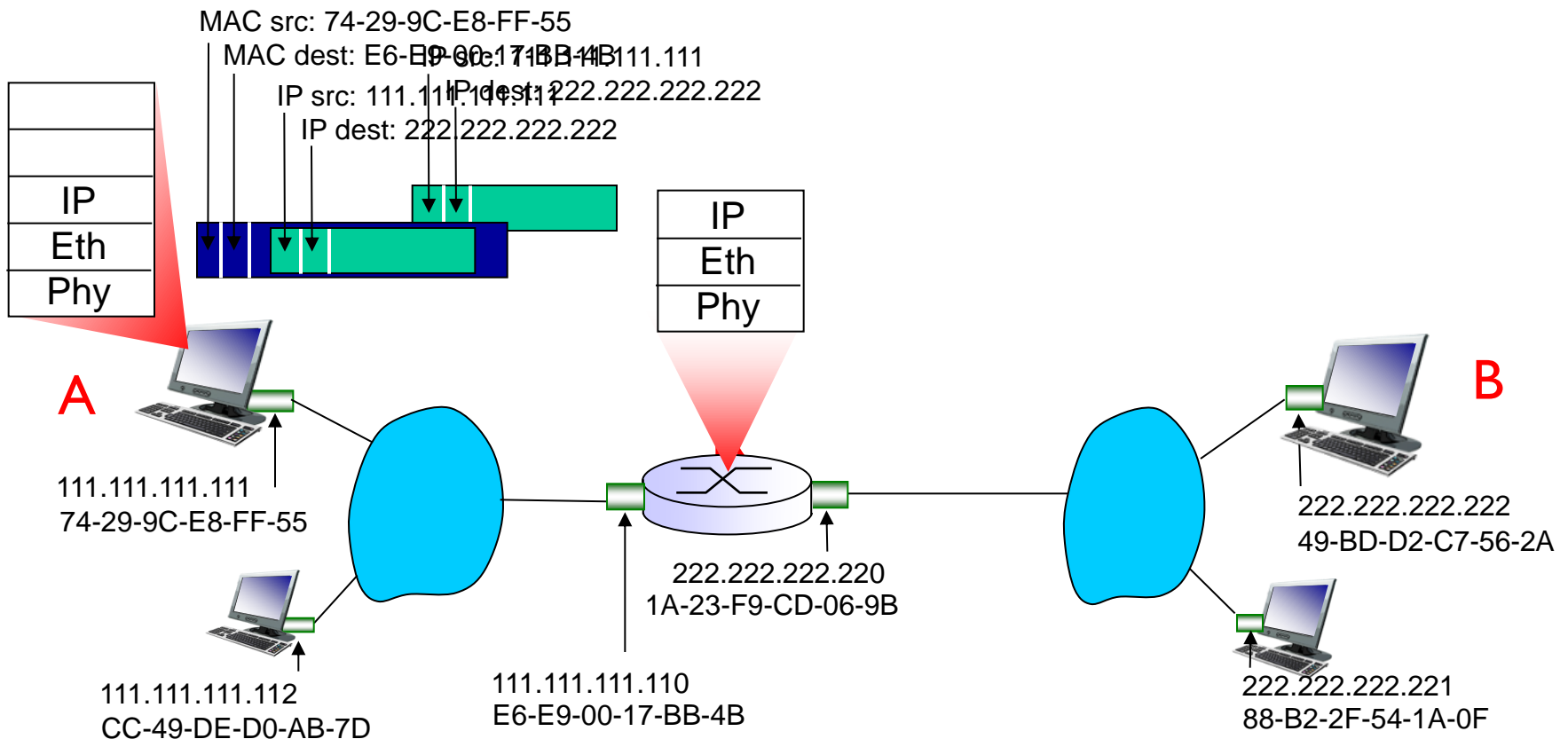
Adressage: envoyer vers un autre LAN

- ❖ A crée le datagramme IP avec @IP source de A, @IP destination de B
- ❖ A crée une trame avec @MAC de R comme dest, la trame encapsule le datagramme créé à l'étape précédente



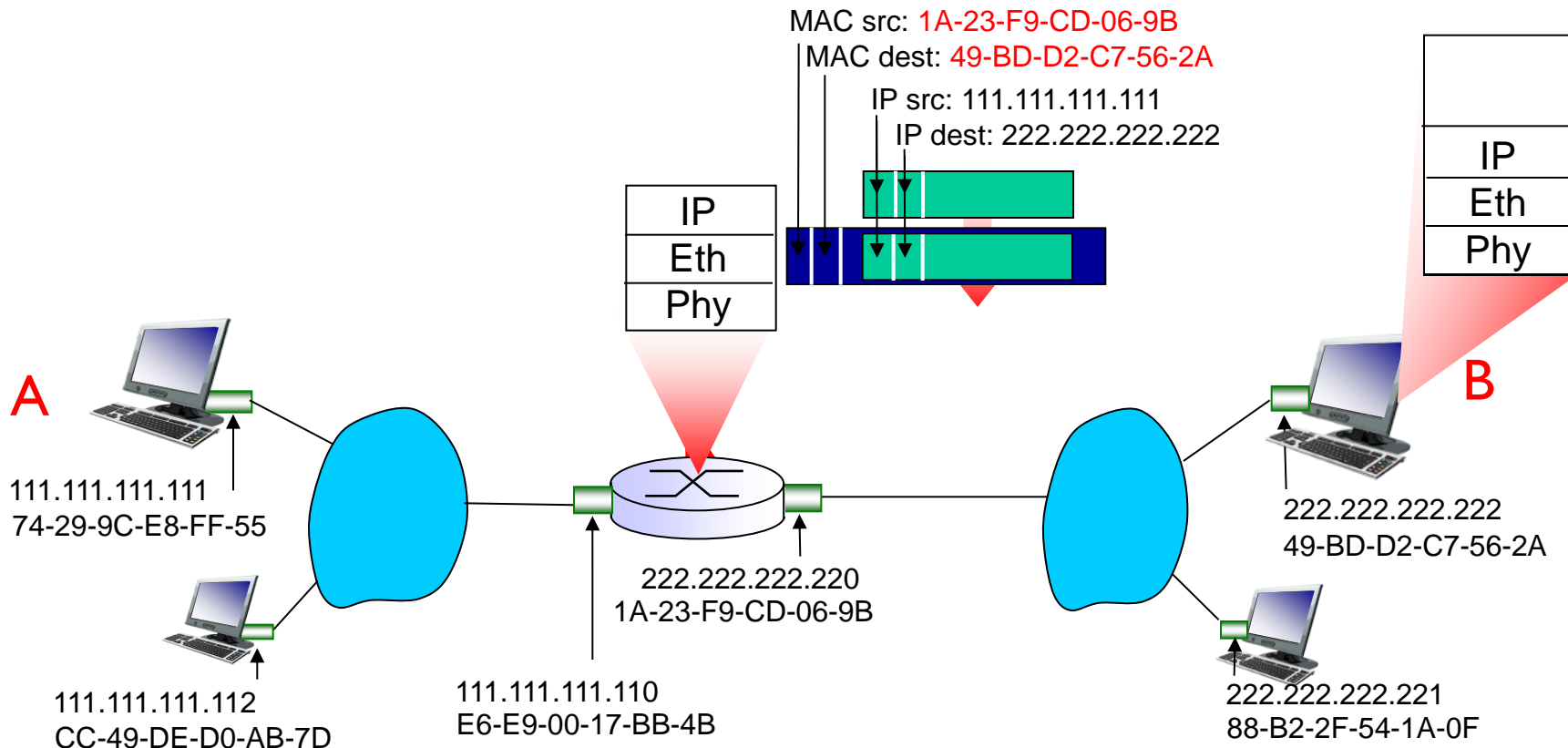
Adressage: envoyer vers un autre LAN

- ❖ trame envoyée de A à R
- ❖ trame reçue par R, passe le datagramme à la couche réseau (IP)



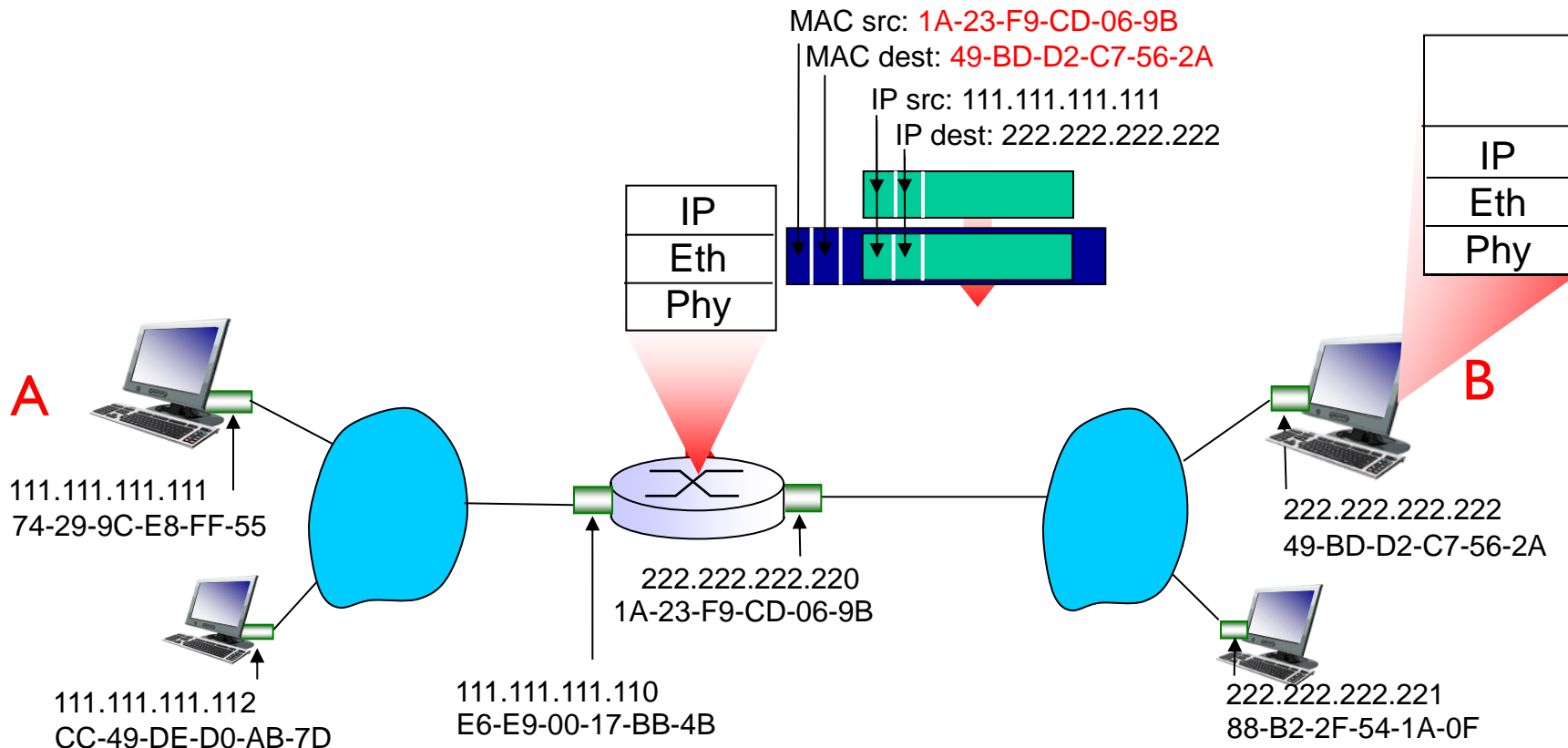
Adressage: envoyer vers un autre LAN

- ❖ R transfère le datagramme avec @IP source A, et @IP destination B
- ❖ R crée une trame avec @MAC de B comme dest



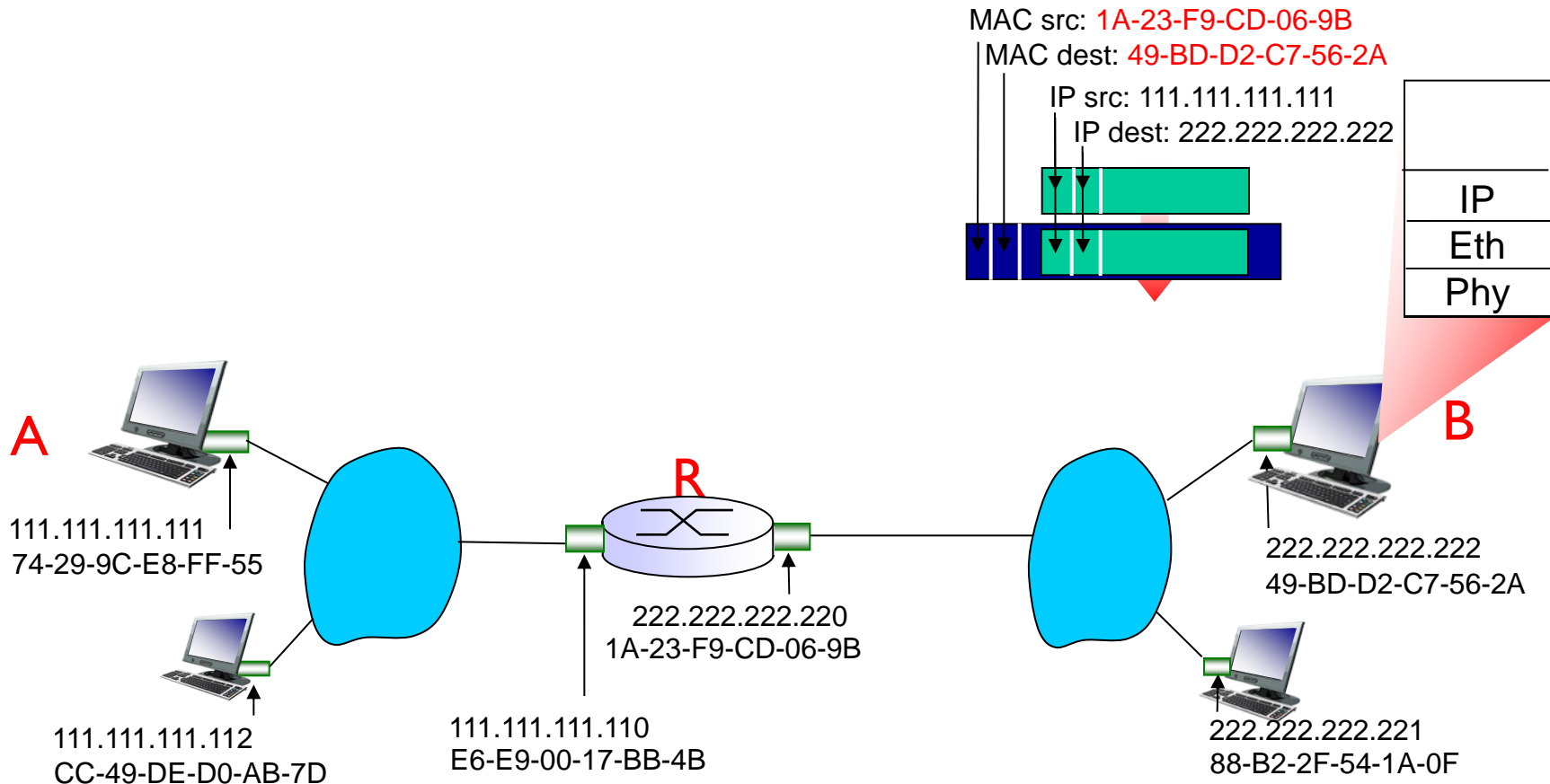
Adressage: envoyer vers un autre LAN

- ❖ R transfère le datagramme avec @IP source A, et @IP destination B
- ❖ R crée une trame avec @MAC de B comme dest



Adressage: envoyer vers un autre LAN

- ❖ R transfère le datagramme avec @IP source A, et @IP destination B
- ❖ R crée une trame avec @MAC de B comme dest



Couche liaison: plan

5.1 introduction, services

5.2 détection d'erreurs,
correction

5.3 protocoles d'accès
multiple

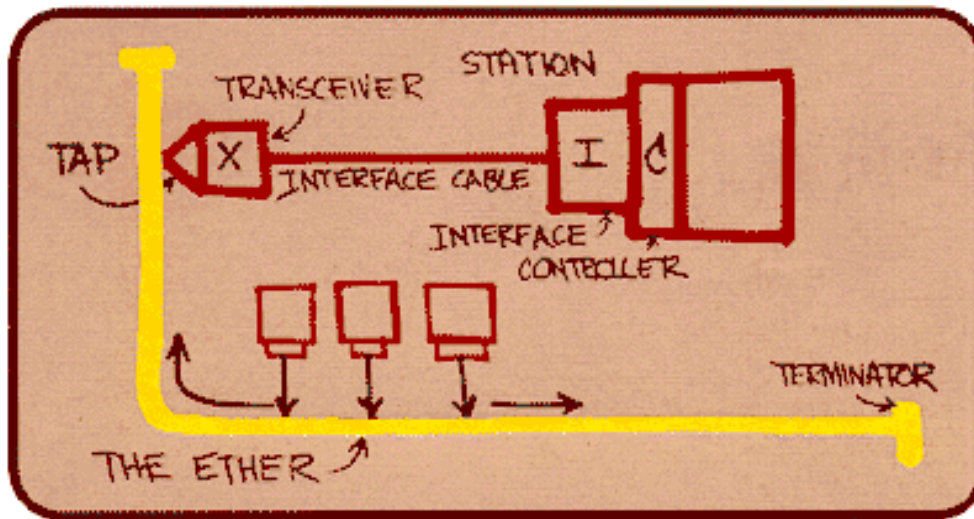
5.4 LANs

- adressage, ARP
- **Ethernet**
- commutateurs
- VLANs

Ethernet

la technologie “dominante” pour les LAN filaires:

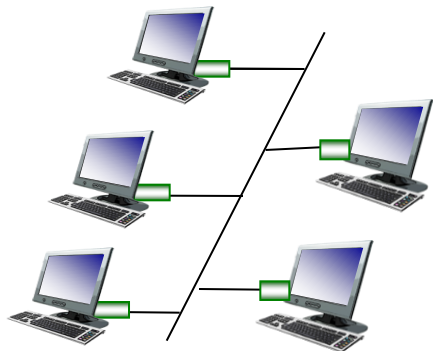
- ❖ les NIC ne sont pas chers
- ❖ la technologie la plus ancienne
- ❖ plus simple, moins coûteuse que les LAN à token et ATM
- ❖ des débits de plus en plus rapides: 10 Mbps – 10 Gbps



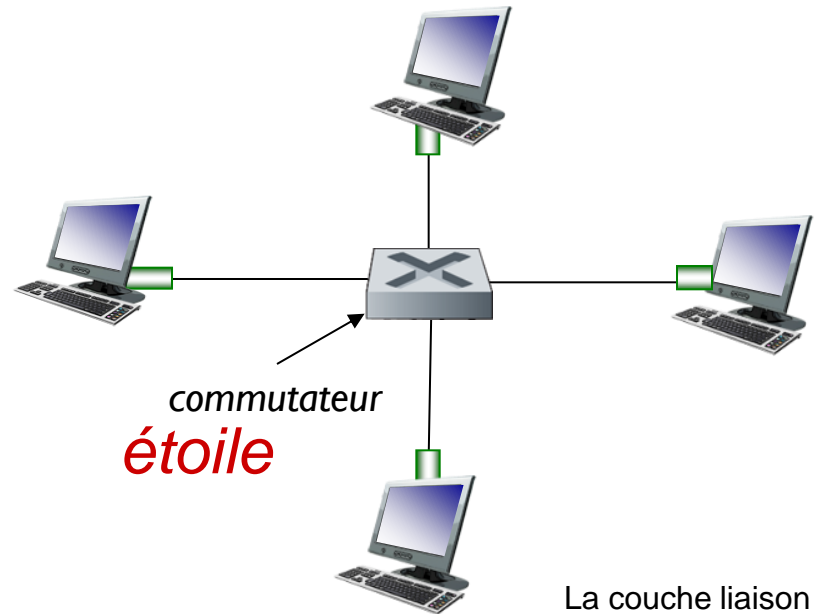
croquis d'Ethernet de Metcalfe

Ethernet: topologie physique

- ❖ **bus**: populaire dans les 90s
 - tous les nœuds dans le même domaine de collision
- ❖ **étoile**: la plus utilisée de nos jours
 - un commutateur (**switch**) au centre
 - chaque nœud exécute un protocole Ethernet d'une manière séparée (pas de collision)



bus: coaxial cable



Structure d'une trame Ethernet

un datagramme IP est encapsulé dans une **trame Ethernet**



préambule:

- ❖ 7 octets avec le pattern 10101010 suivi d'un octet 10101011
- ❖ synchroniser l'horloge de l'émetteur avec celle du récepteur

Structure d'une trame Ethernet

- ❖ *adresses*: adresses MAC source et destination
- ❖ *type*: indique le protocole de couche supérieure
- ❖ *CRC*: contrôle d'erreur chez le récepteur
 - si erreur détecté: trame supprimé

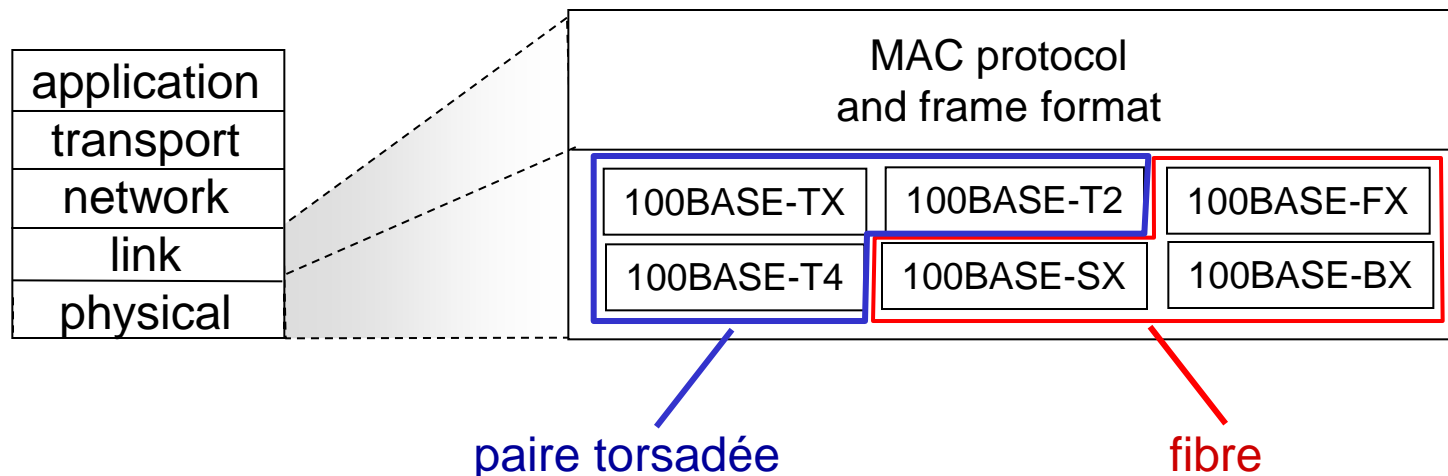


Ethernet: non fiable, sans connexion

- ❖ *sans connexion*: pas de poignée de main entre émetteur et récepteur
- ❖ *non fiable*: pas de ACKs ni NACKs
 - Les données dans les trames éliminées seront recouvert seulement si l'émetteur initial utilise une fiabilité à une couche supérieure (par ex. TCP)
- ❖ protocole MAC de Ethernet : *unslotted CSMA/CD avec backoff binaire*

Standards Ethernet 802.3 : couches phy et liaison

- ❖ *plusieurs* standards Ethernet différents
 - débits différents: 2 Mbps, 10 Mbps, 100 Mbps, 1 Gbps, 10G bps
 - des médias physiques différents: fibre, câble



Couche liaison: plan

5.1 introduction, services

5.2 détection d'erreurs,
correction

5.3 protocoles d'accès
multiple

5.4 LANs

- adressage, ARP
- Ethernet
- commutateurs
- VLANs

Commutateur Ethernet

❖ équipement de couche liaison:

- enregistre, transfère les trames Ethernet
- examine les adresses MAC des trames entrant et les transfère vers la ou les bonnes sorties
- utilise CSMA/CD pour accéder au lien

❖ *transparent*

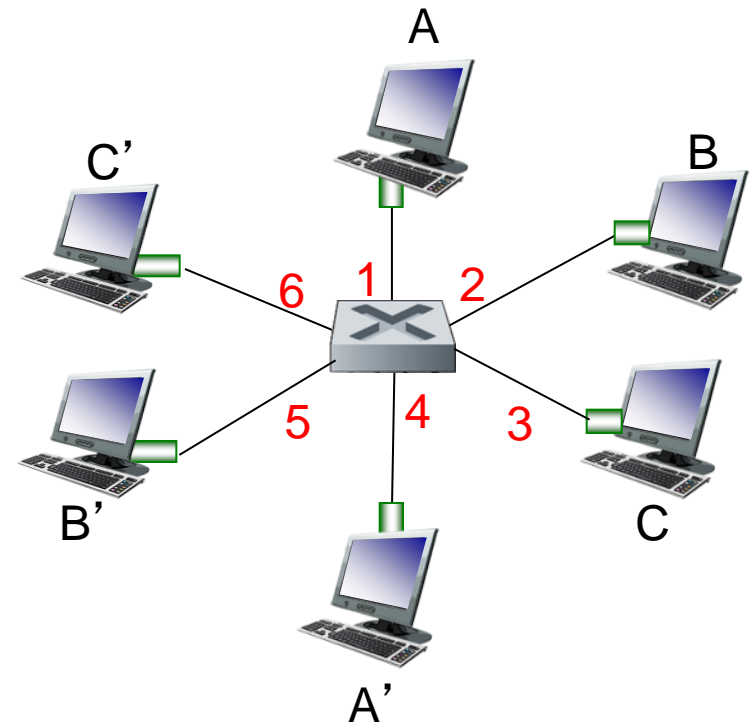
- les hôtes ne sentent pas la présence des commutateurs

❖ *plug-and-play*

- ne nécessite pas une configuration manuelle

Commutateur: plusieurs transmissions simultanées

- ❖ chaque hôte possède un lien dédié le connectant au commutateur
- ❖ commutateur met les paquets dans un buffer
- ❖ chaque lien utilise le protocole Ethernet mais sans collisions; full duplex
 - chaque lien se trouve dans son propre domaine de collision
- ❖ **switching (commutation): les transmissions A-à-A' et B-à-B' peuvent avoir lieu simultanément**

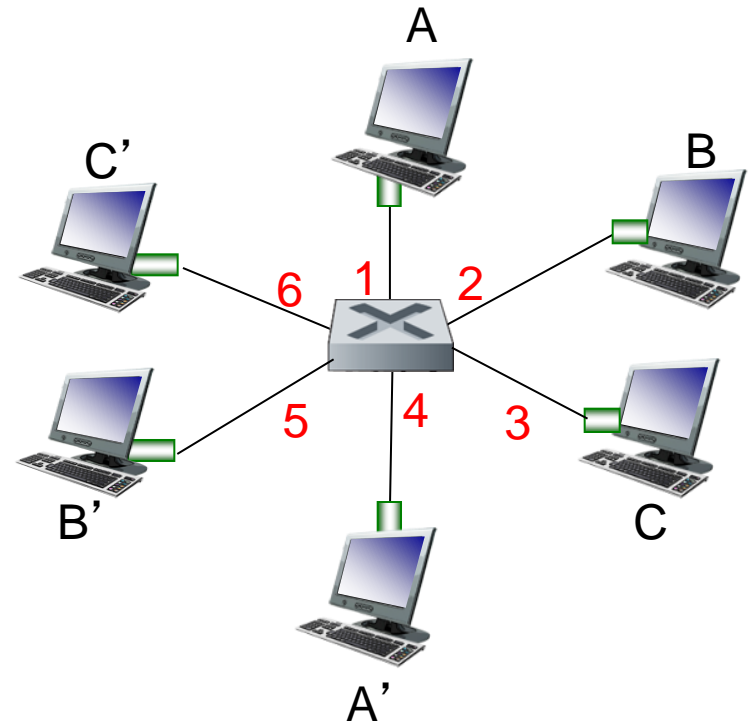


*commutateur avec six interfaces
(1,2,3,4,5,6)*

Commutateur: table de transfert

Q: comment le commutateur sait qu'il peut atteindre A' via l'interface 4 et B' via l'interface 5? A: chaque commutateur possède une **table** qui contient:

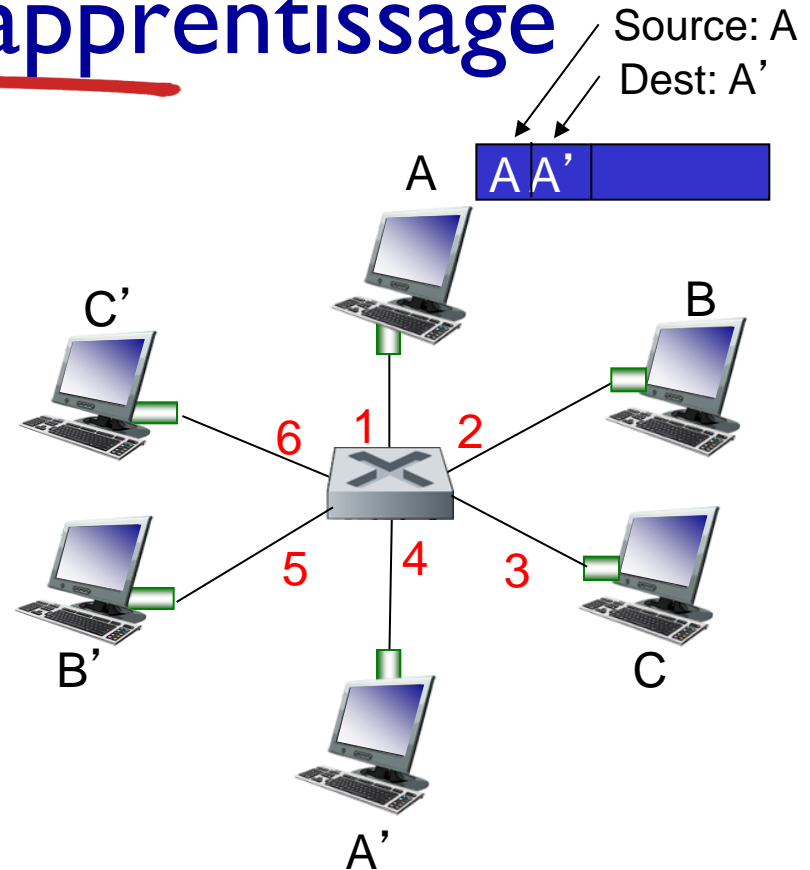
- (@MAC du hôte, interface pour atteindre l'hôte, timestamp)
- ressemble à la table du routeur!



commutateur avec six interfaces
(1,2,3,4,5,6)

Commutateur: auto-apprentissage

- ❖ commutateur *apprend* les interfaces qui mènent aux différents hôtes
 - lorsqu'il reçoit une trame, le commutateur "apprend" la localisation de l'émetteur
 - remplit sa table



@MAC	interface	TTL
A	1	60

*table du commutateur
(vide au début)*

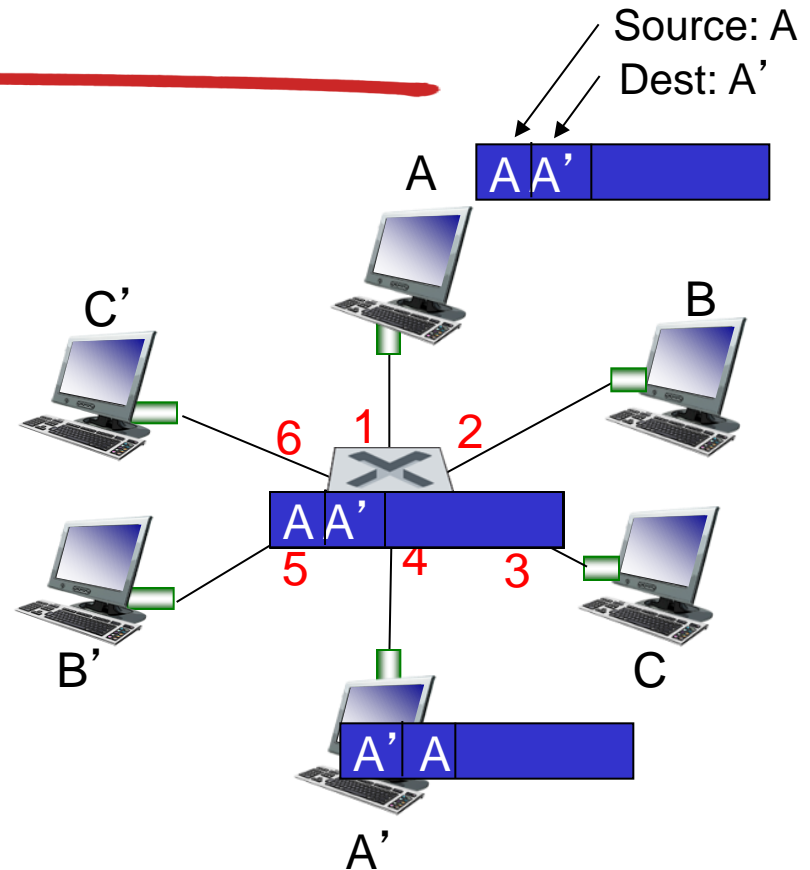
Commutateur: filtrage/transfert des trames

lorsque la trame est reçue au commutateur:

1. ajoute l'interface et @MAC dans sa table
2. cherche dans sa table l'entrée qui correspond à @MAC destination
3. si l'entrée est trouvée
 alors {
 si l'interface par laquelle vient la trame est la même que celle trouvée dans la table
 alors éliminer la trame
 sinon transférer la trame sur l'interface trouvée
 }
 sinon diffuser sur toutes les interfaces

Exemple

- ❖ destination A', inconnue pour le commutateur:
diffuser
- ❖ destination A connue:
envoyer sur le lien correspondant

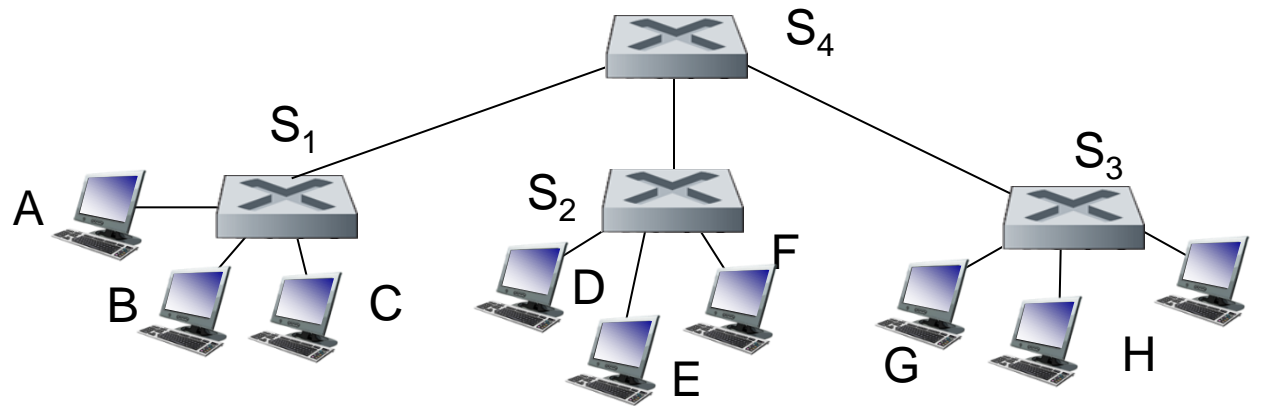


adr. MAC	interface	TTL
A	1	60
A'	4	60

table du commutateur

Interconnecter les commutateurs

- ❖ commutateurs peuvent être interconnectés

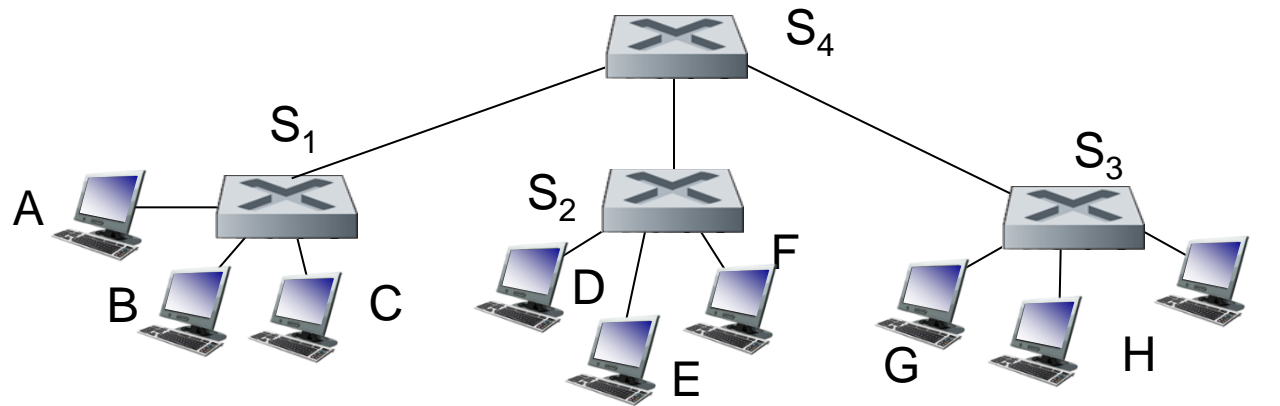


Q: un envoi de A à G - comment S_1 sait que la trame vers G doit passer par S_4 et S_3 ?

- ❖ **A:** par auto-apprentissage (comme précédemment)

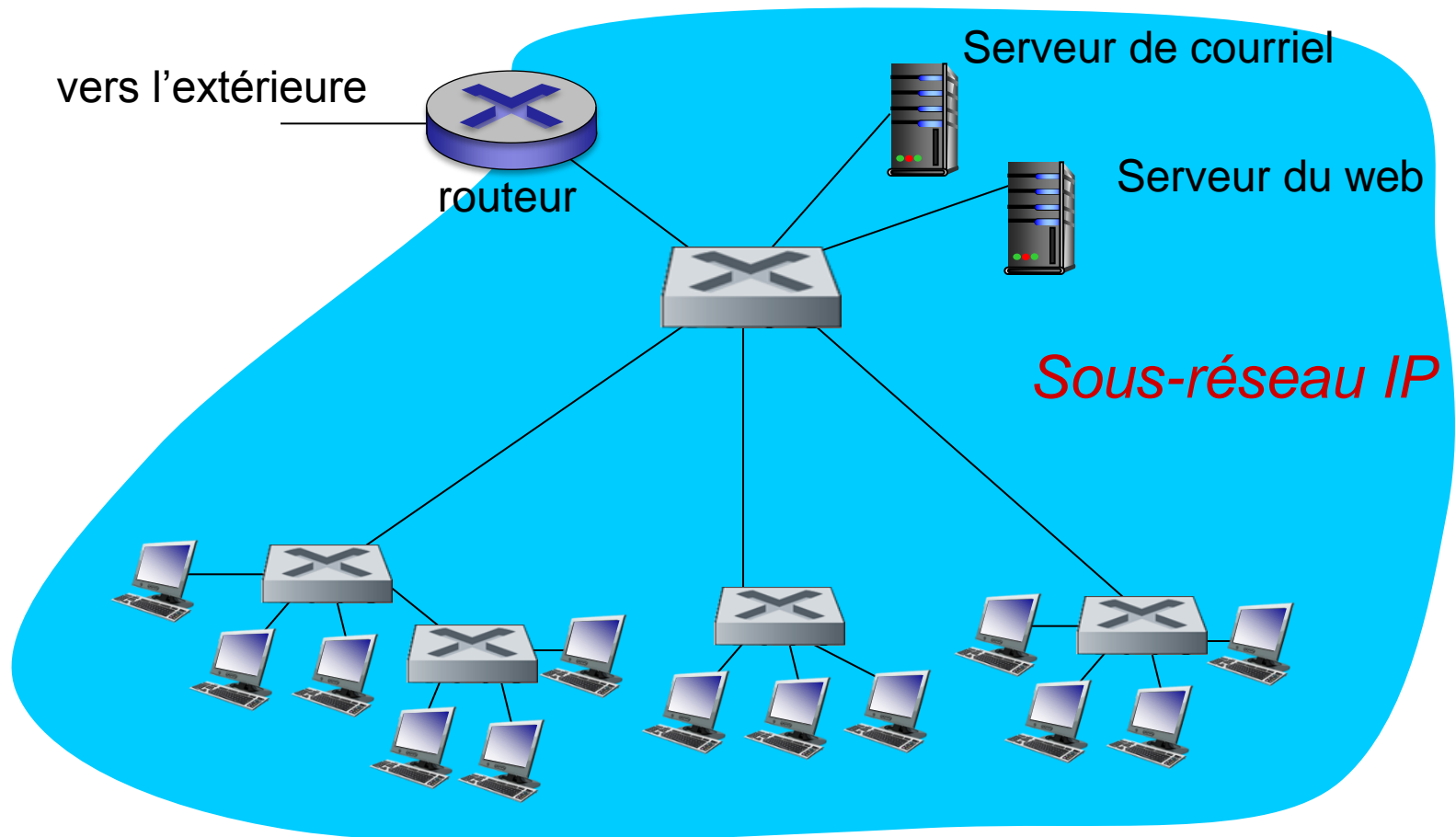
Auto-apprentissage (exemple)

Supposons que C envoie une trame à I puis I répond à C



❖ Q: quelles sont les tables de S_1, S_2, S_3, S_4 ?

Réseau institutionnel



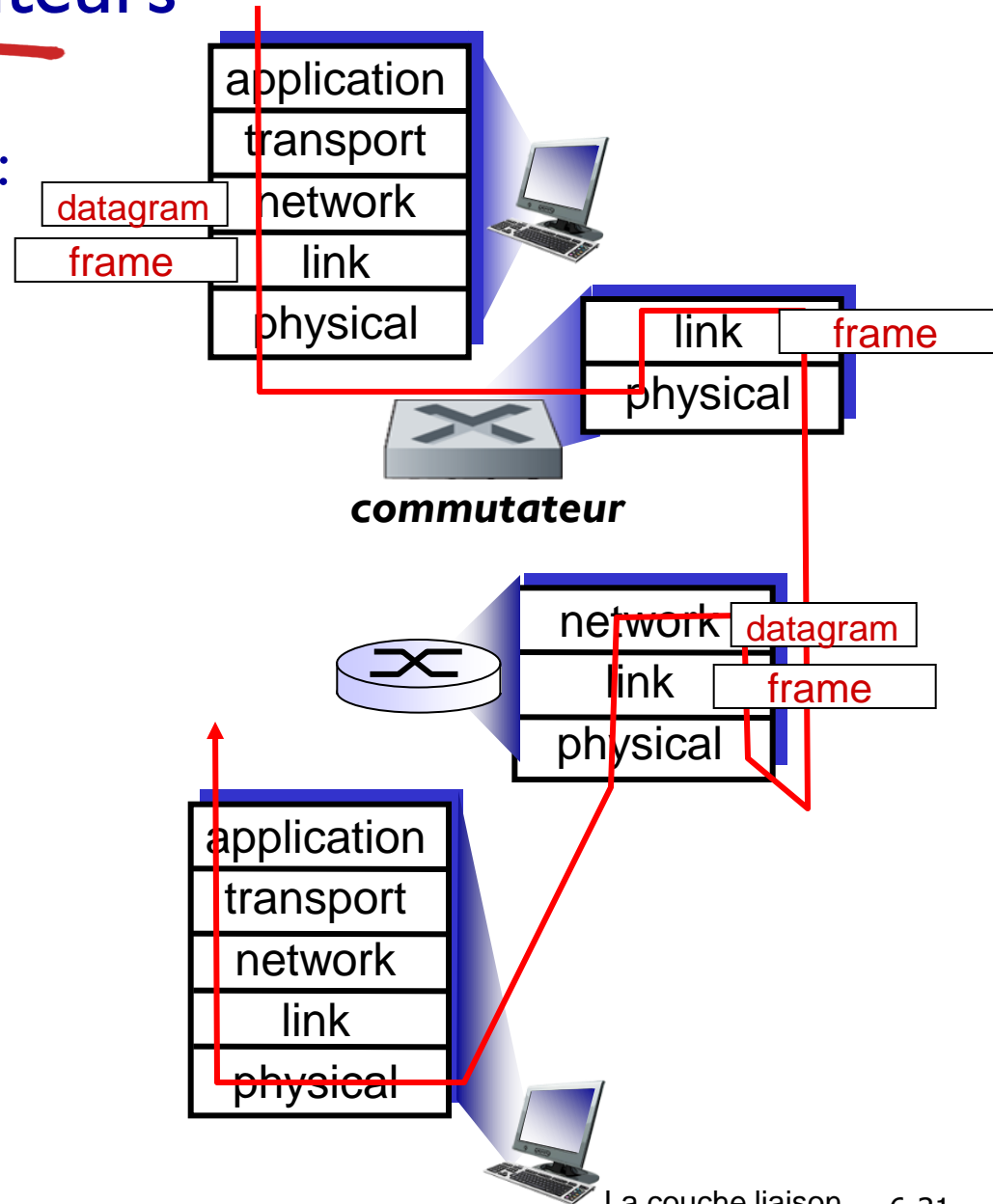
Commutateurs vs. routeurs

ils utilisent store-and-forward:

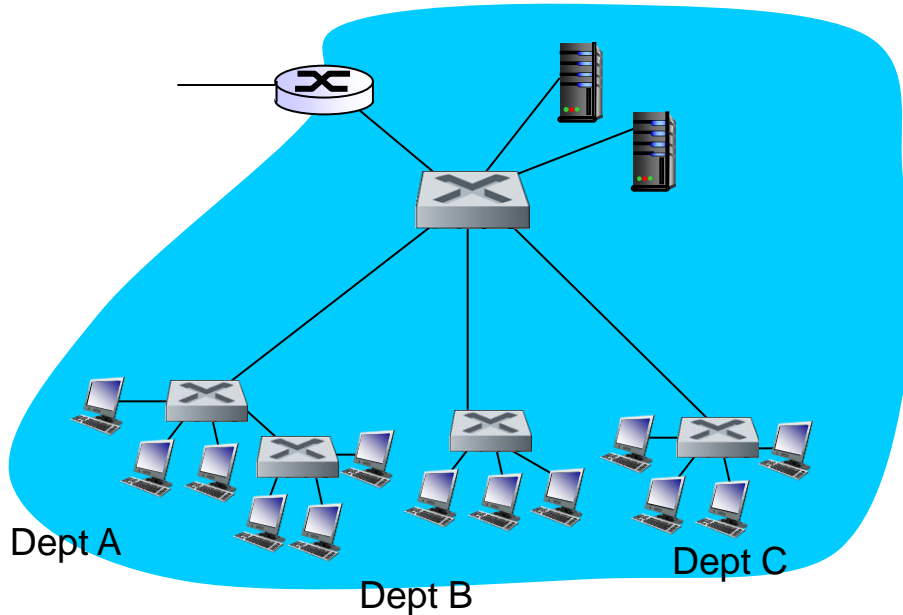
- **routeurs**: examinent les en-têtes de la couche réseau
- **commutateurs**: examinent les en-têtes de la couche liaison

ils utilisent des tables de transfert:

- **routeurs**: utilisent les algos de routage et les @IP
- **commutateurs**: auto-apprentissage et @MAC



VLANs: motivation



soit:

- ❖ un utilisateur du DeptA a changé de bureau au DeptB mais veut se connecter au commutateur A?
- ❖ un seul domaine de diffusion:
 - tout le trafic de couche 2 doit passer par tout le LAN
 - problème de sécurité et d'efficacité

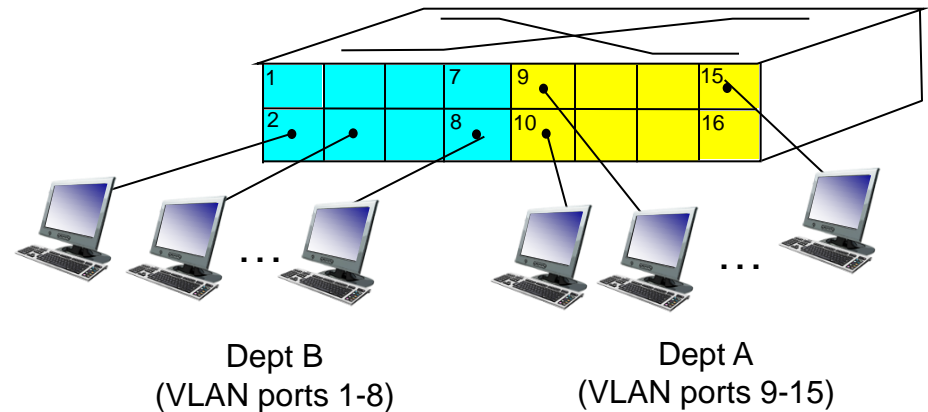
VLANs

Virtual Local Area Network

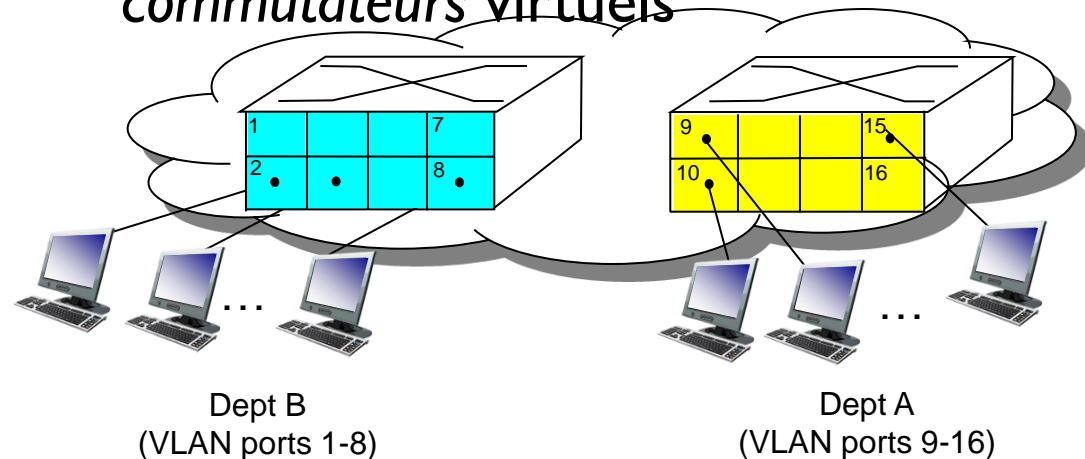
commutateurs
spéciaux peuvent être
configurer pour définir
plusieurs LANs
virtuels dans un
même LAN physique

port-based VLAN: les ports du
commutateur sont groupés pour
qu'un seul commutateur physique

.....

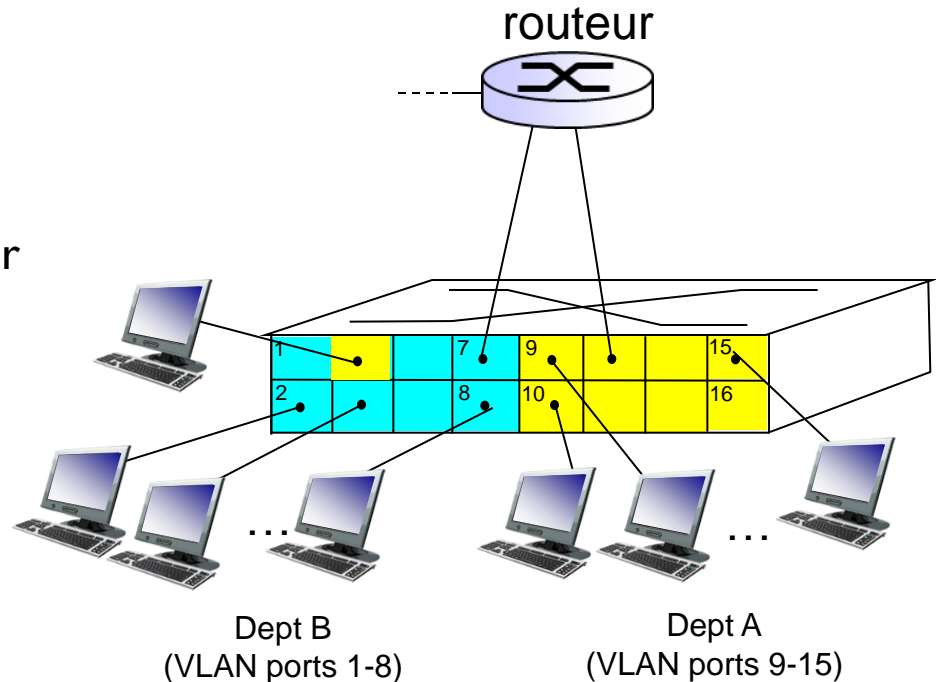


... opère comme plusieurs
commutateurs virtuels

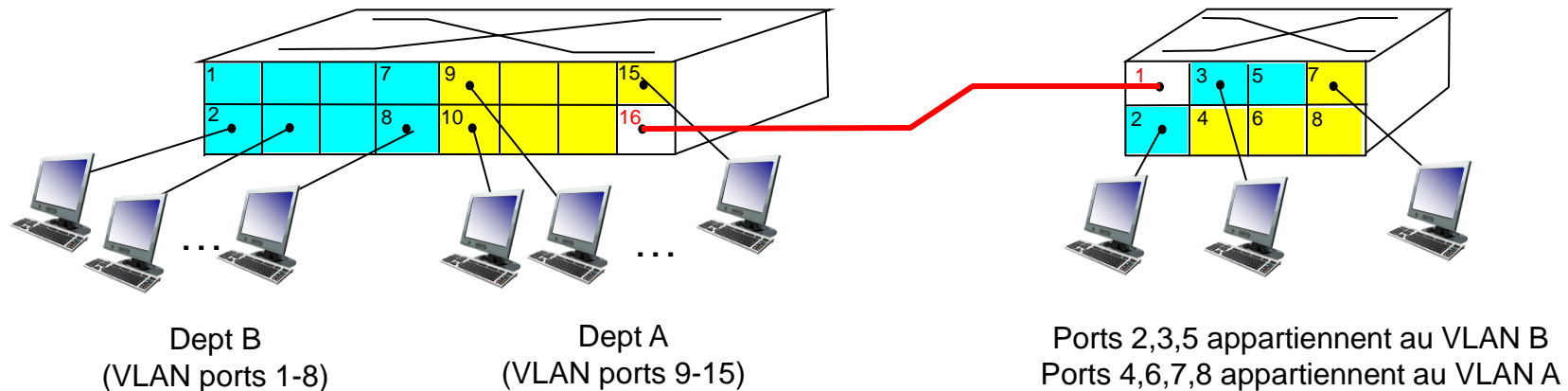


Port-based VLAN

- ❖ *isolation du trafic* : trames vers/de ports 1-8 atteignent seulement les ports 1-8
 - on peut définir des VLAN basés sur les @MAC au lieu des ports des commutateurs
- ❖ *appartenance dynamique* : assignation dynamique des ports au VLANs
- ❖ *transfert entre VLANs* : exactement comme pour des commutateurs séparés

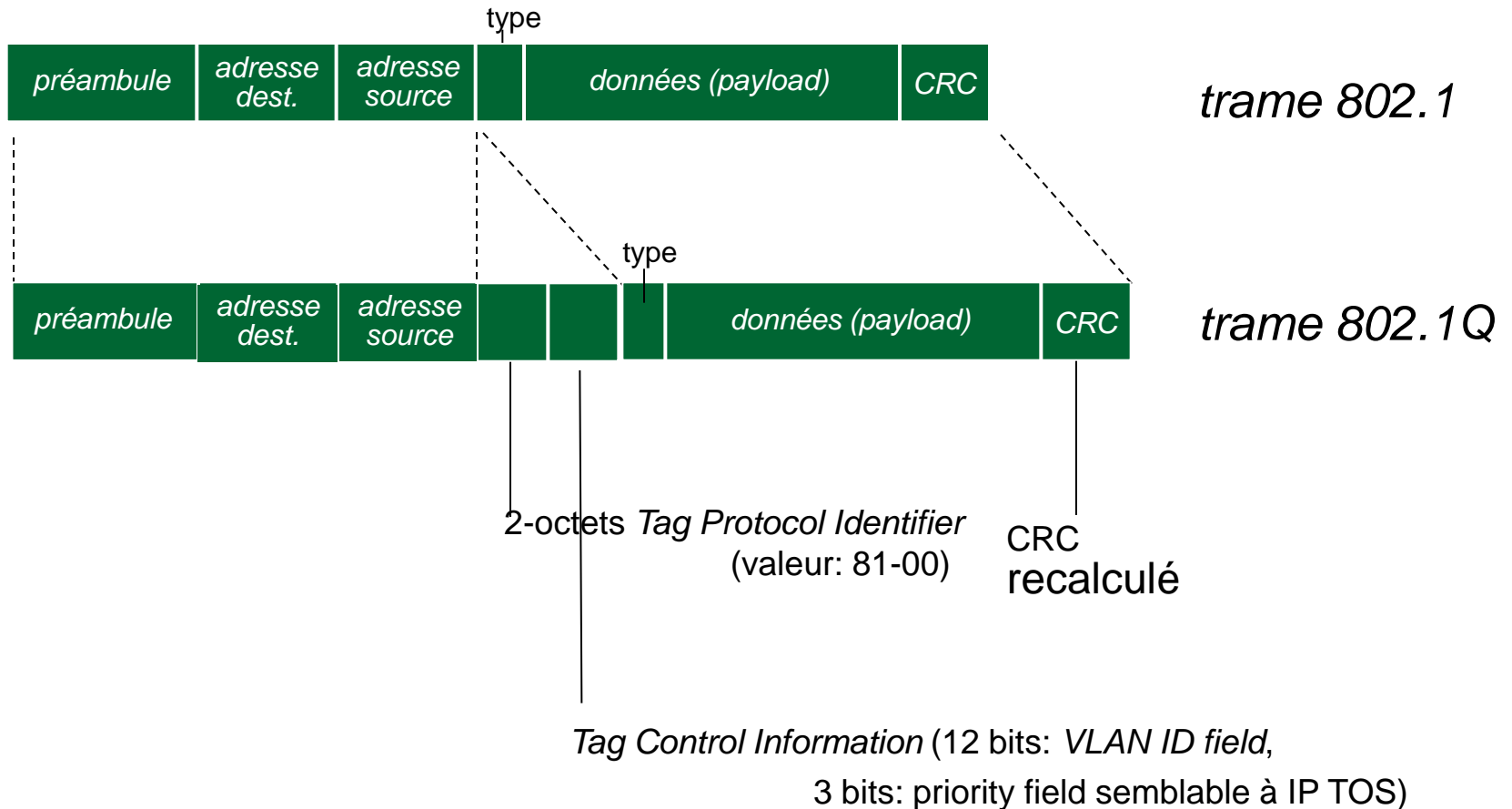


VLANs définis sur plusieurs commutateurs



- ❖ **trunk port:** transporte les trames dans les VLANs définis sur plusieurs commutateurs physiques
 - les trames transférés dans un même VLAN entre deux commutateurs ne peuvent pas être des trames vanilla 802.1 (doivent contenir VLAN ID)
 - le protocole 802.1q ajoute/supprime les champs additionnels de l'en-tête pour les trames transférées entre les trunk ports

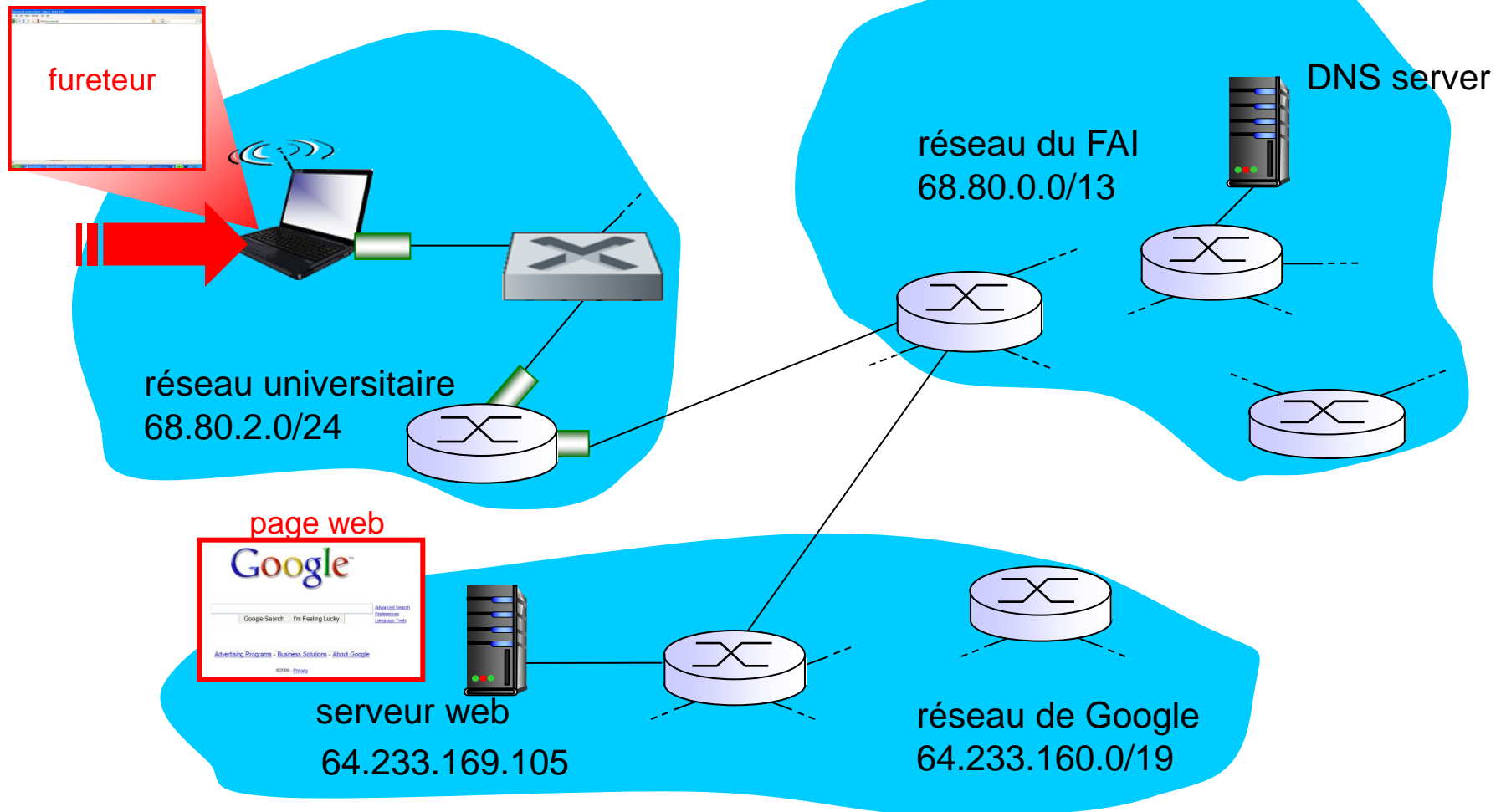
802.1Q VLAN: format de la trame



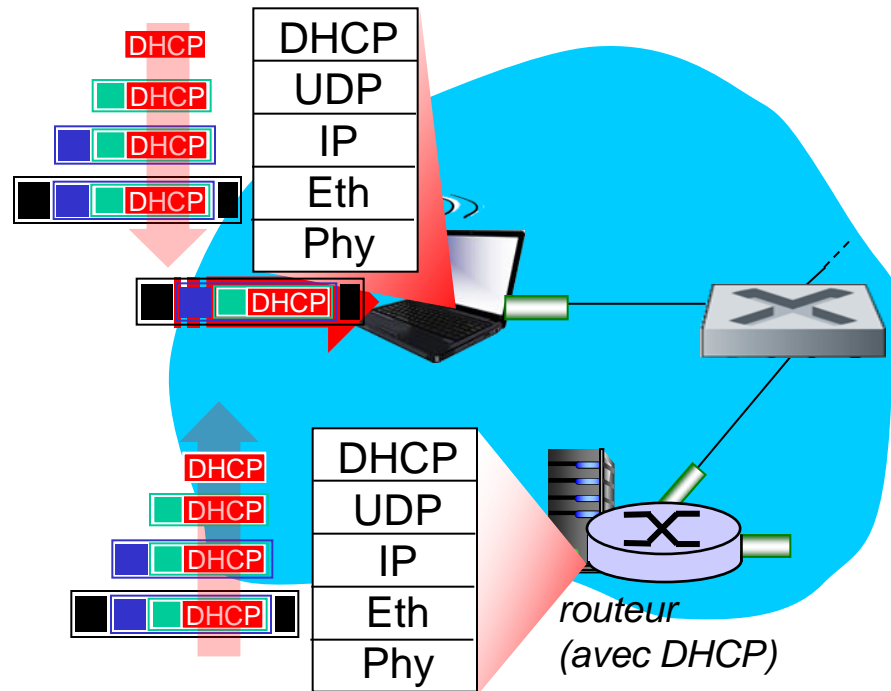
Scénario récapitulatif: requête d'une page web

- ❖ Notre cours s'achève
 - application, transport, réseau, liaison
- ❖ un scénario récapitulatif
 - *objectif*: identifier et réviser les protocoles (couches) qui interagissent lors d'une requête d'une page web

Scénario récapitulatif

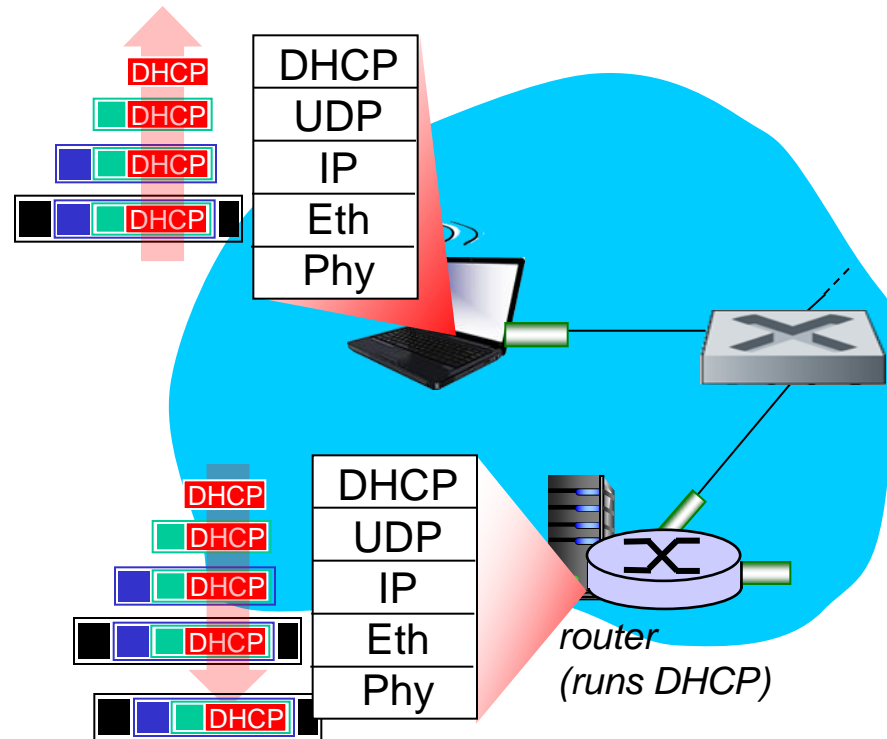


Scénario récapitulatif ... vers Internet



- ❖ pour se connecter, le terminal a besoin d'une @ IP, @ de la passerelle et @ du serveur DNS : utilise **DHCP**
- requête DHCP **encapsulée** en **UDP**, encapsulée en **IP**, encapsulée en **802.3** (Ethernet)
- Trame Ethernet de **diffusion** (dest: FFFFFFFFFFFFFFFF), reçu au niveau du routeur qui implémente un serveur **DHCP**
- Ethernet **demux** à IP qui demux à UDP qui demux à DHCP

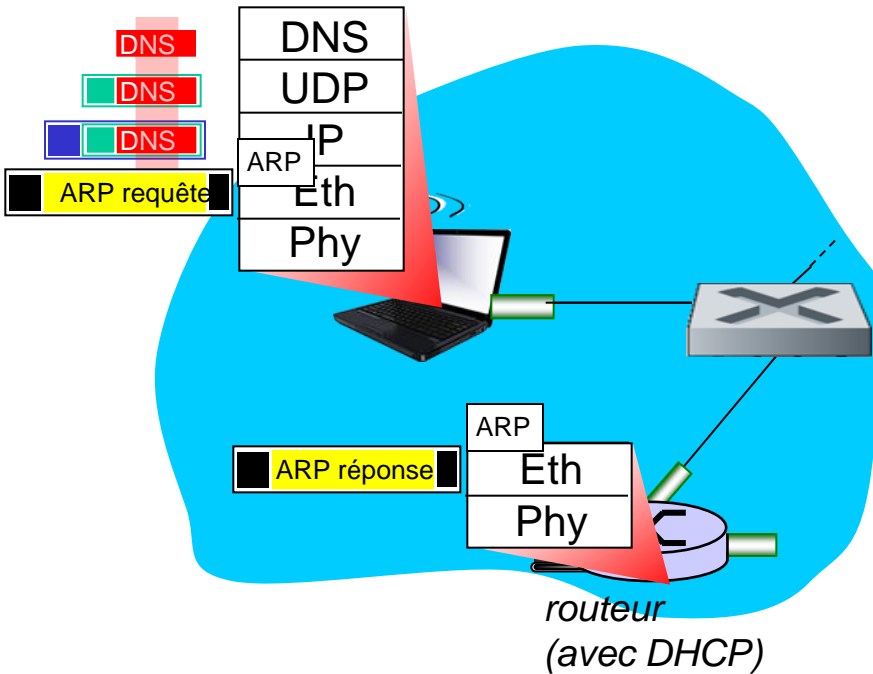
Scénario récapitulatif ... vers Internet



- ❖ serveur DHCP crée un paquet **DHCP ACK** contenant les différentes adresses IP
- le paquet DHCP est encapsulé dans une trame permettant au commutateur d'apprendre une nouvelle entrée

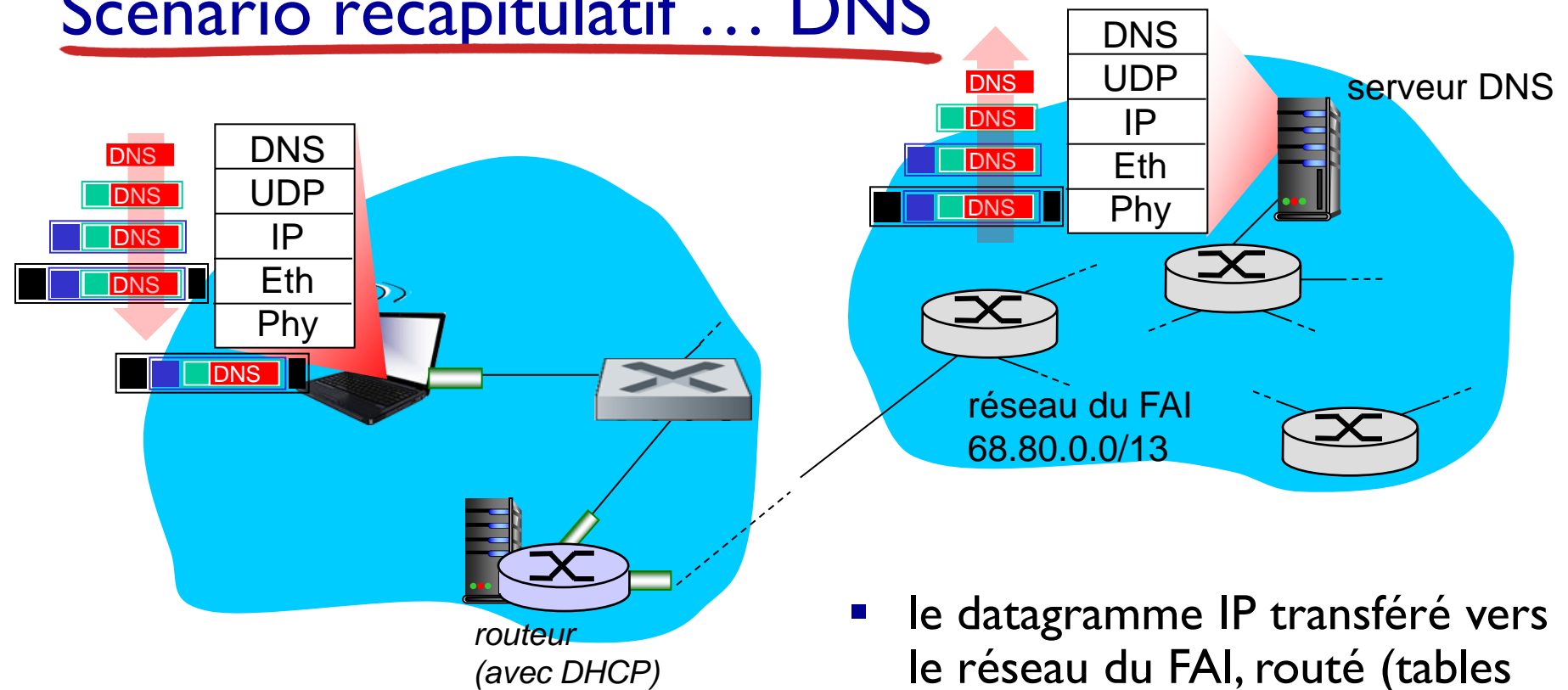
Client dispose d'une @ IP, connaît le nom et l'@ du serveur DNS, ainsi que l'@ IP de la passerelle

Scénario récapitulatif... ARP (avant DNS, avant HTTP)



- ❖ avant d'envoyer une requête **HTTP**, besoin de l'@ IP de **www.google.com**: **DNS**
- requête DNS créée, encapsulée en UDP, encapsulée en IP, encapsulée en Eth. Mais, besoin de l'@ MAC du routeur: **ARP**
- **une requête ARP** est diffusée, reçue par le routeur, qui réponds par une **réponse ARP** qui fournit son @ MAC
- client dispose maintenant de l'@ MAC de la passerelle et peut envoyer une trame contenant une requête DNS

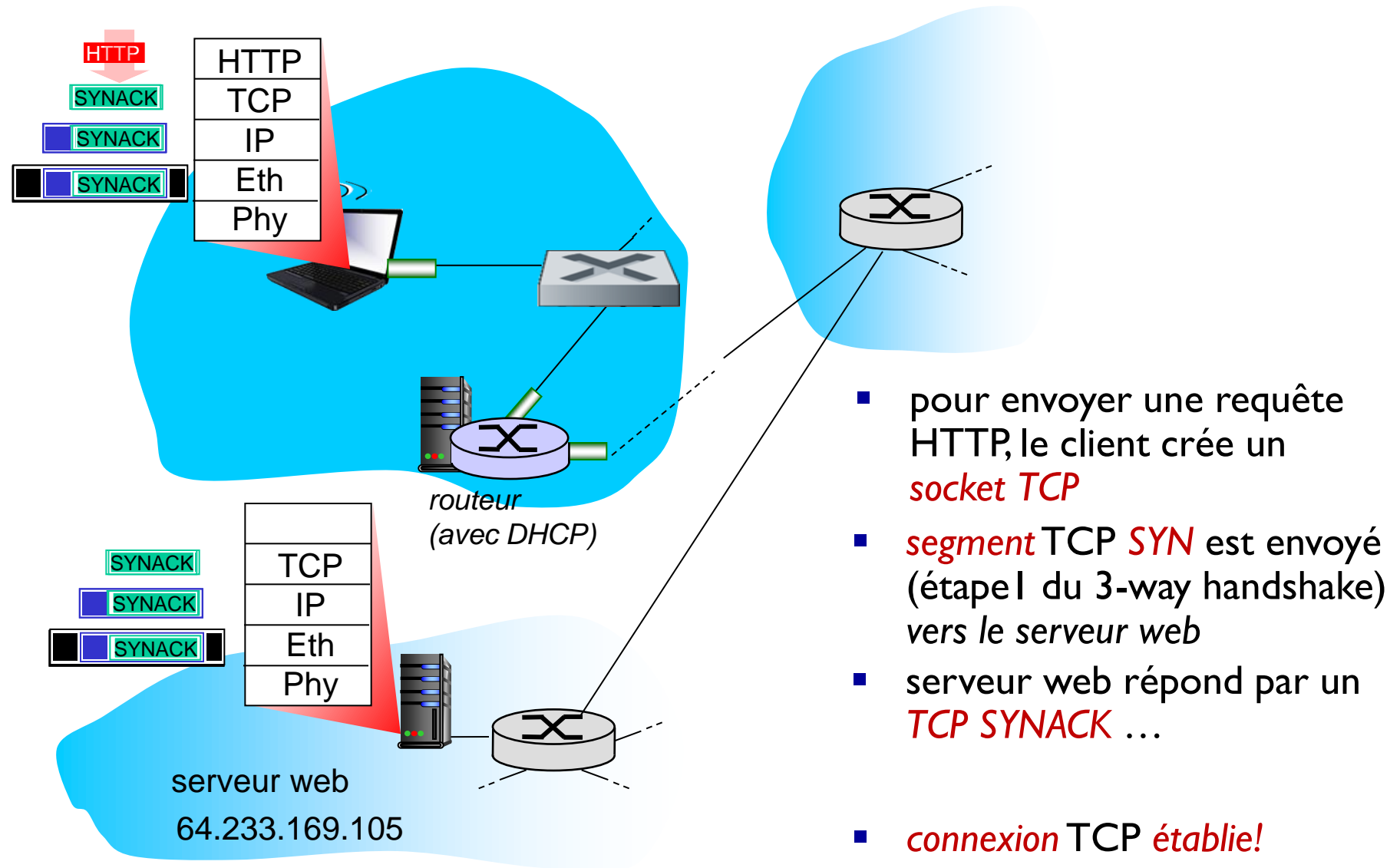
Scénario récapitulatif ... DNS



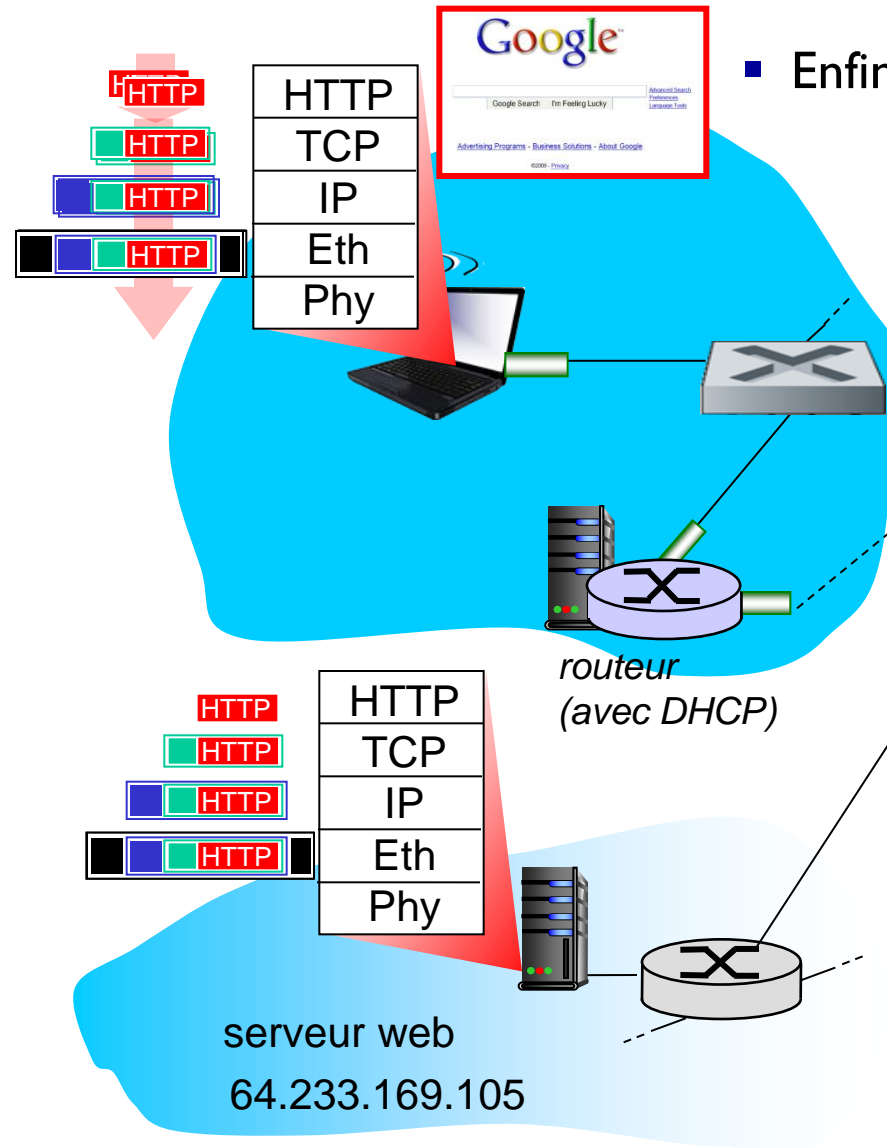
- la requête DNS (un datagramme IP) est transférée par le commutateur du client vers la passerelle

- le datagramme IP transféré vers le réseau du FAI, routé (tables créées par les protocoles *RIP*, *OSPF*, *IS-IS* et/ou *BGP*) vers le serveur DNS
- demux vers le serveur DNS
- serveur DNS réponds au client en fournissant l'@ IP de www.google.com

Scénario récapitulatif ... connection TCP (HTTP)



Scénario récapitulatif ... HTTP requête/réponse



- Enfin, la page web est affichée

- **requête HTTP** envoyée au socket TCP
- un datagramme IP contenant la requête HTTP route vers www.google.com
- le serveur web répond avec **réponse HTTP** (contenant une page web)
- datagramme IP contenant la réponse HTTP route vers le client