

*Reti di calcolatori e Internet:  
Un approccio top-down*

7<sup>a</sup> edizione  
Jim Kurose, Keith Ross

Pearson Paravia Bruno Mondadori Spa

# Capitolo 6: Livello di collegamento e reti locali

6.1 Livello di collegamento: introduzione e servizi

6.2 Tecniche di rilevazione e correzione degli errori

6.3 Protocolli di accesso multiplo

6.4 Indirizzi a livello di collegamento

6.5 Ethernet

6.6 Switch a livello di collegamento

# Indirizzi MAC

## □ Indirizzo IP a 32 bit:

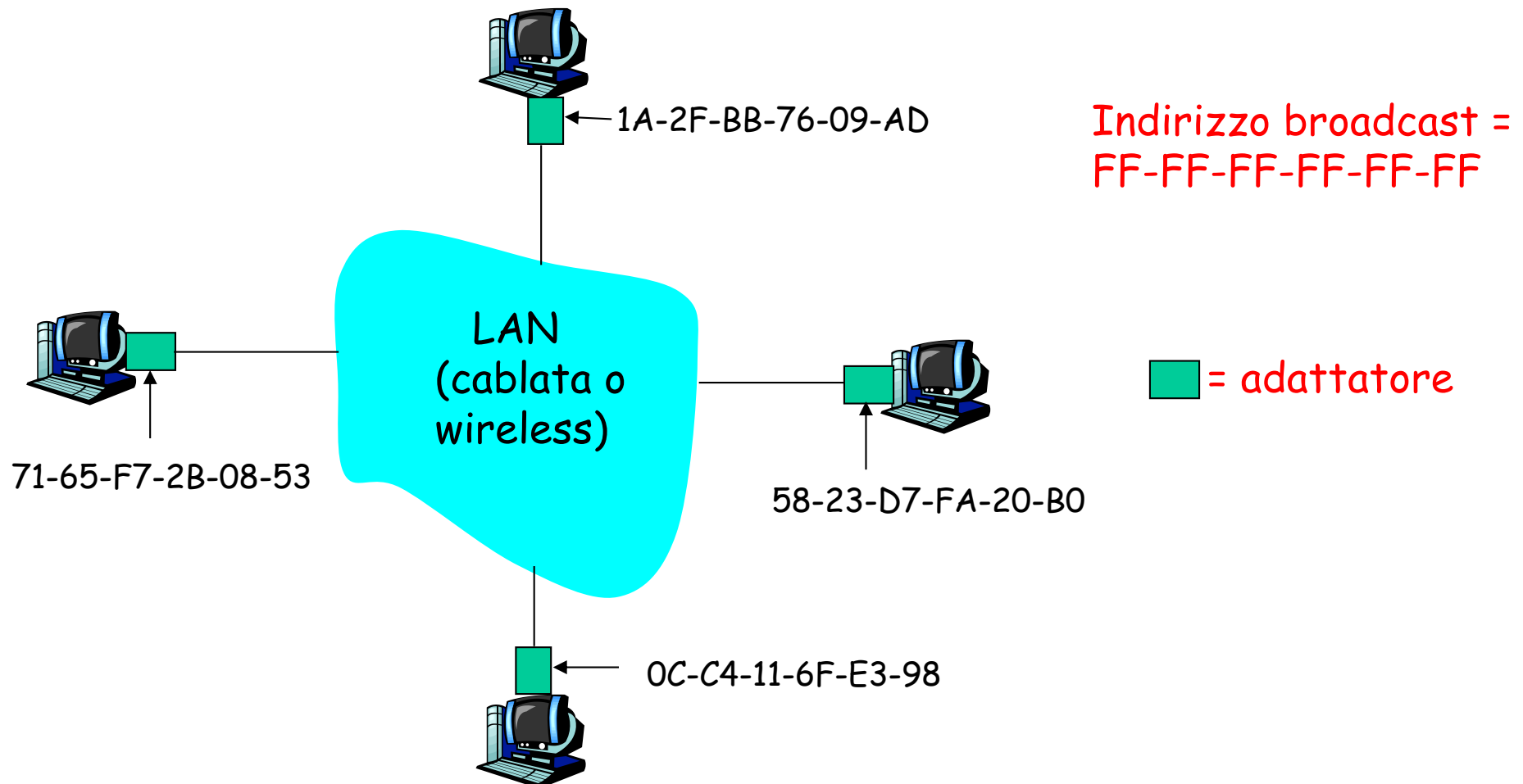
- Indirizzo a *livello di rete*.
- Analogo all'indirizzo postale di una persona: hanno una struttura gerarchica e devono esser aggiornati quando una persona cambia residenza.

## □ Indirizzo MAC (o LAN o fisico o Ethernet):

- Analogo al numero di codice fiscale di una persona: ha una struttura orizzontale e non varia a seconda del luogo in cui la persona si trasferisce.
- Indirizzo a 48 bit (per la maggior parte delle LAN) .

# Indirizzi MAC

Ciascun adattatore di una LAN ha un indirizzo MAC univoco .

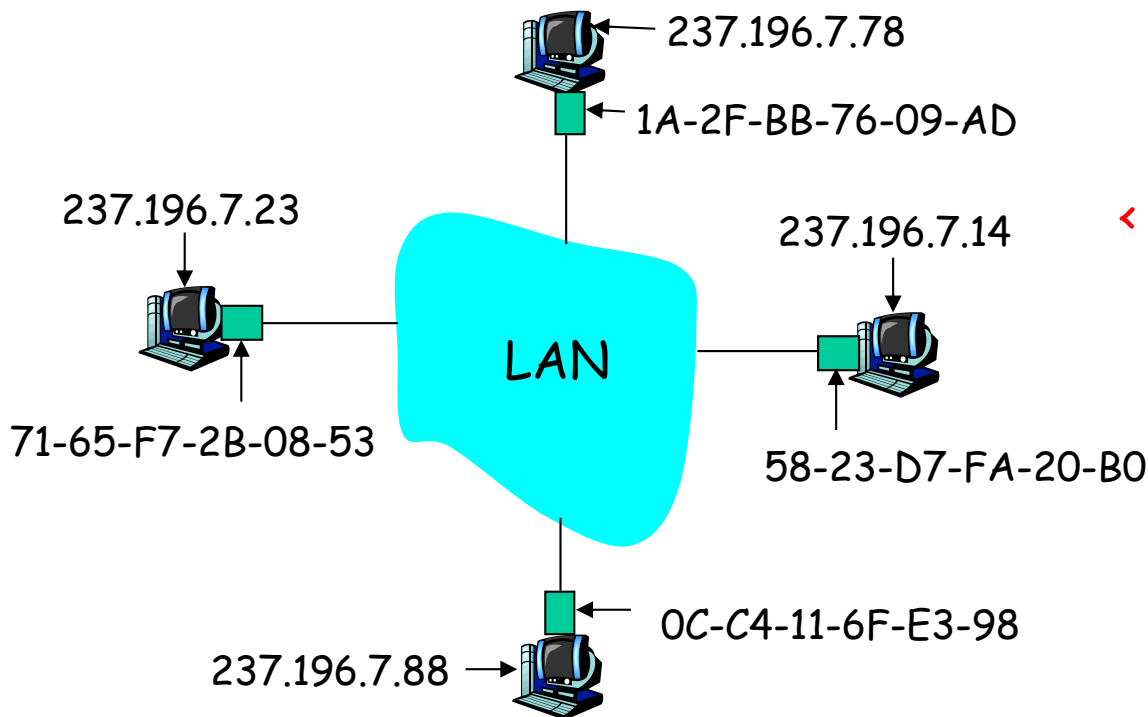


# Indirizzi MAC

- ❑ La IEEE sovrintende alla gestione degli indirizzi MAC.
- ❑ Quando una società vuole costruire adattatori, compra un blocco di spazio di indirizzi (unicità degli indirizzi).
- ❑ Analogia:
  - ❑ (a) Indirizzo MAC: come il codice fiscale di una persona.
  - ❑ (b) Indirizzo IP: come l'indirizzo postale di una persona.
- ❑ Indirizzo orizzontale MAC → portabilità
  - È possibile spostare una scheda LAN da una LAN a un'altra.
- ❑ Gli indirizzi IP hanno una struttura gerarchica e devono essere aggiornati se spostati.
  - dipendono dalla sottorete IP cui il nodo è collegato.

# Protocollo per la risoluzione degli indirizzi (ARP)

Domanda: come si determina l'indirizzo MAC di B se si conosce solo l'indirizzo IP di B?



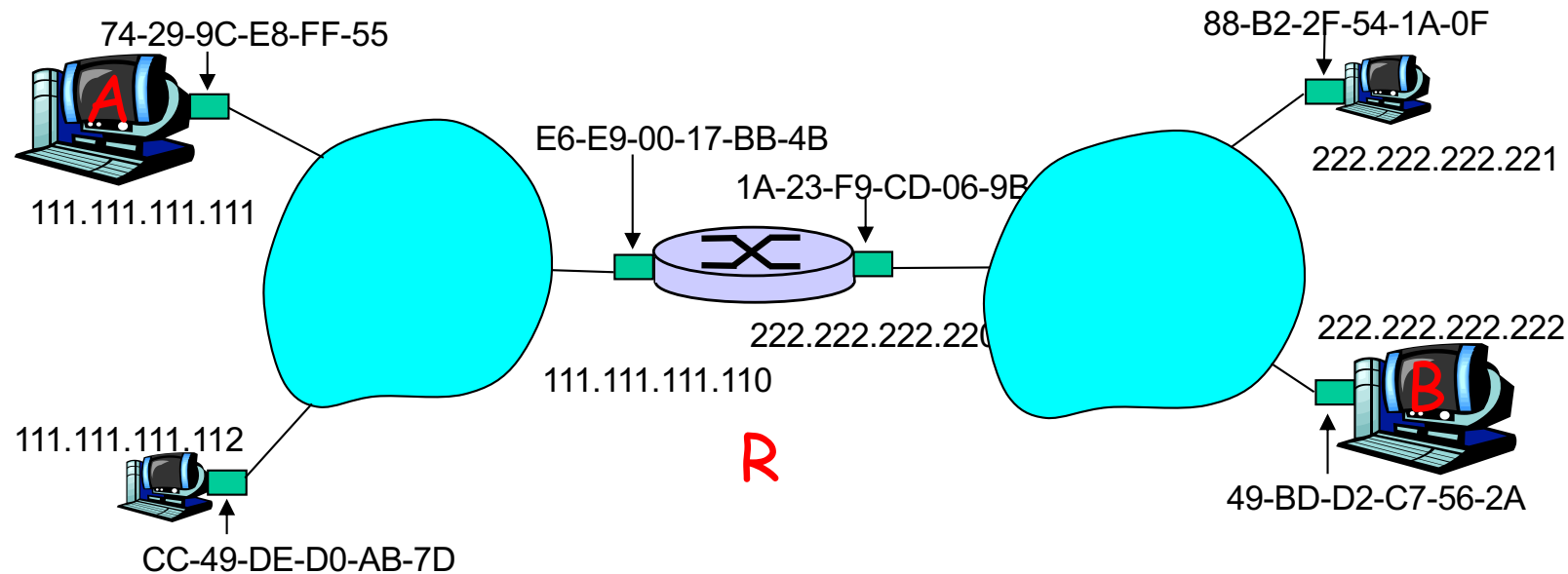
- r Ogni nodo IP (host, router) nella LAN ha una **tabella ARP**.
- r Tabella ARP: contiene la corrispondenza tra indirizzi IP e MAC.
  - < Indirizzo IP; Indirizzo MAC; TTL >
  - m TTL (tempo di vita): valore che indica quando bisognerà eliminare una data voce nella tabella (il tempo di vita tipico è di 20 min).

# Protocollo ARP nella stessa sottorete

- *A* vuole inviare un datagramma a *B*, e l'indirizzo MAC di *B* non è nella tabella ARP di *A*.
- *A* trasmette in un pacchetto **broadcast** il messaggio di richiesta ARP, contenente l'indirizzo IP di *B*.
  - Indirizzo MAC del destinatario  
= FF-FF-FF-FF-FF-FF
  - Tutte le macchine della LAN ricevono una richiesta ARP.
- *B* riceve il pacchetto ARP, e risponde ad *A* comunicandogli il proprio indirizzo MAC.
  - il frame viene inviato all'indirizzo MAC di *A*.
- Il messaggio di richiesta ARP è inviato in un pacchetto broadcast mentre il messaggio di risposta ARP è inviato in un pacchetto standard.
- ARP è "plug-and-play":
  - La tabella ARP di un nodo si costituisce automaticamente e non deve essere configurata dall'amministratore del sistema.

# Invio verso un nodo esterno alla sottorete

Invio di un datagramma da A a B attraverso R, ipotizzando che A conosca l'indirizzo IP di B.

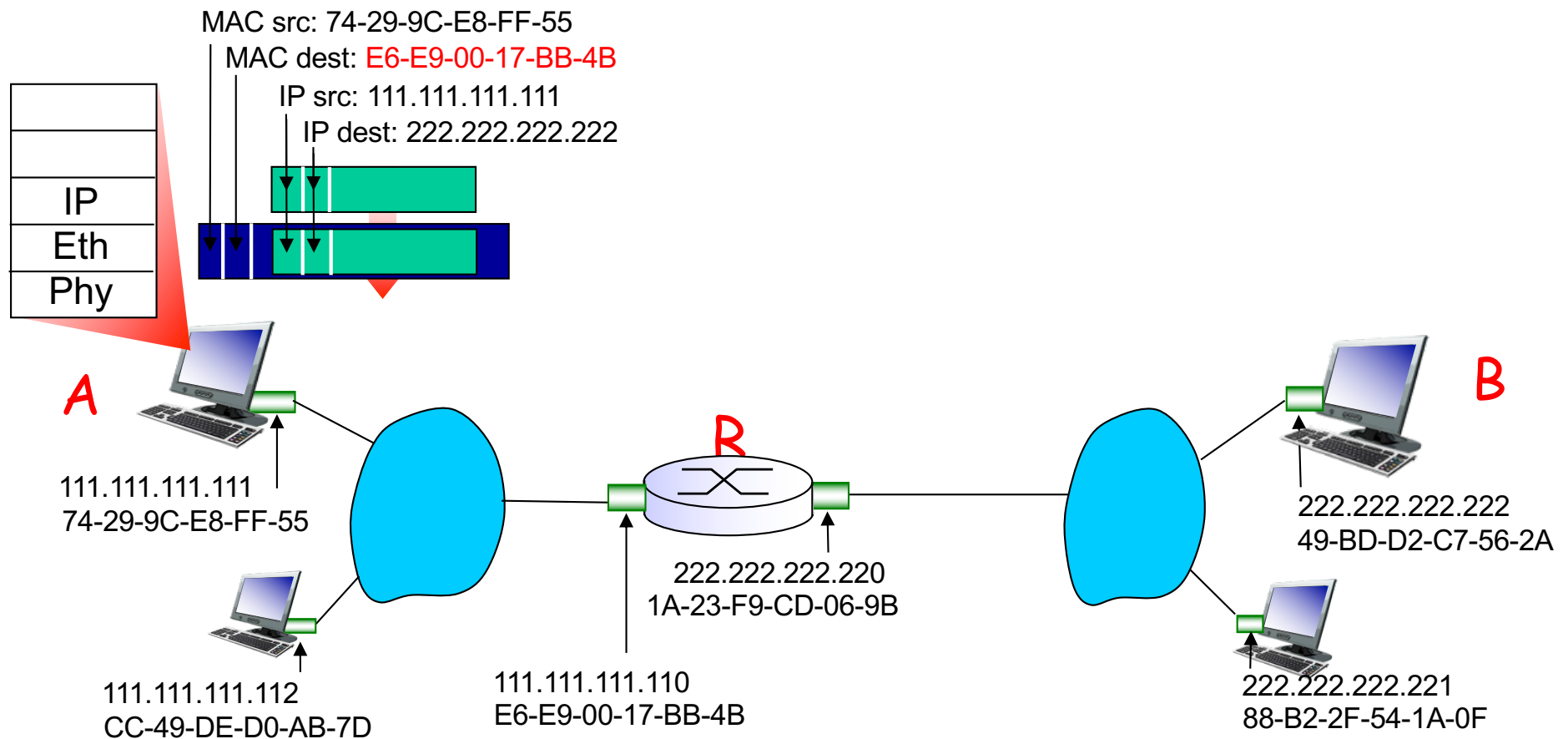


- Due tabelle ARP nel router R, una per ciascuna rete IP (LAN).



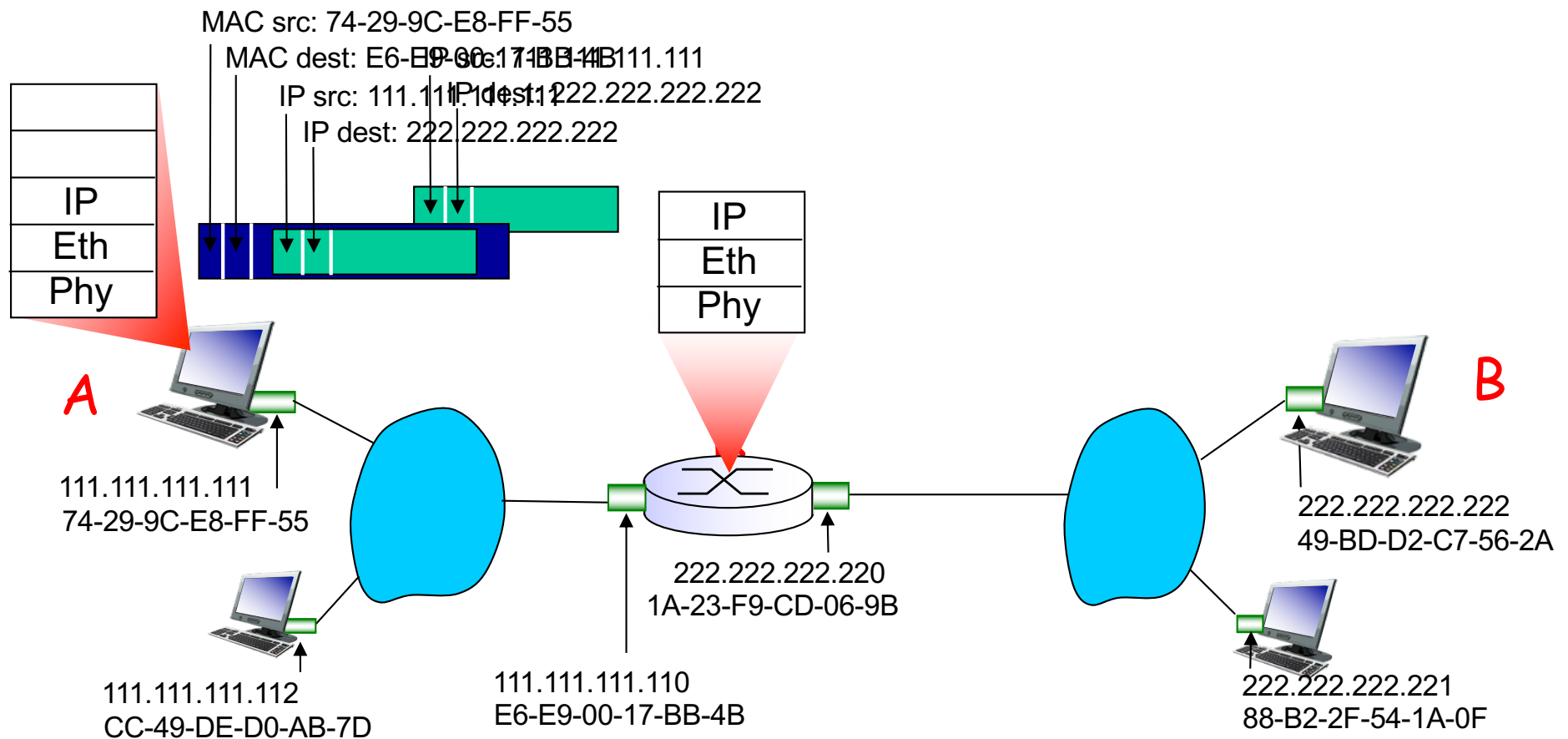
# Invio verso un nodo esterno alla sottorete

- ❖ A crea un datagramma con origine A, e destinazione B.
- ❖ A crea un frame con l'indirizzo MAC di R come destinazione (conosciuto con ARP), contenente il datagramma IP A-to-B



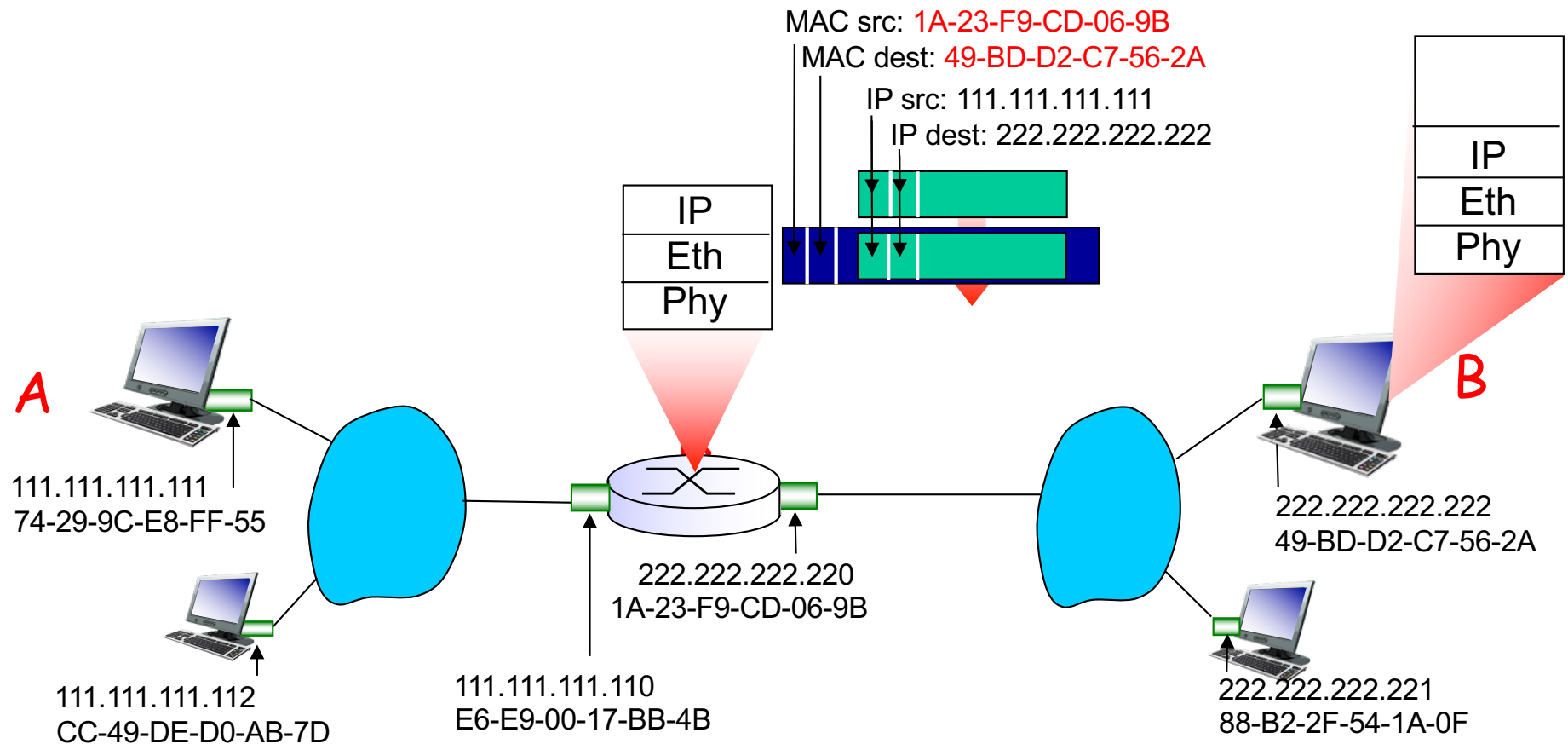
# Invio verso un nodo esterno alla sottorete

- ❖ A invia il frame ad R
- ❖ R riceve il frame, estrae il datagramma e lo passa al livello IP



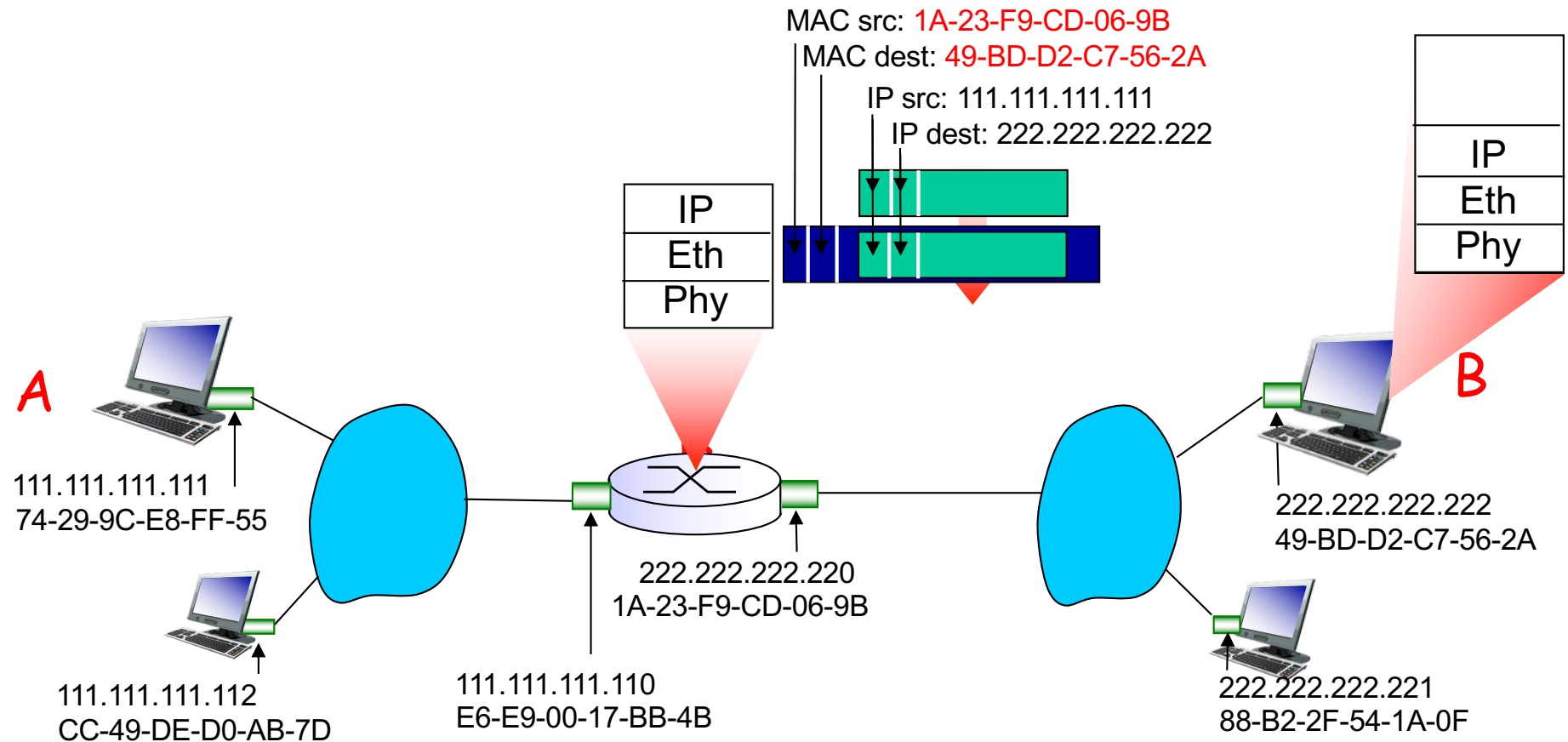
# Invio verso un nodo esterno alla sottorete

- ❖ R inoltra il datagramma con IP sorgente di A e IP destinazione di B
- ❖ R crea un frame con l'indirizzo MAC di B come destinazione (conosciuto con ARP), contenente il datagramma IP A-to-B



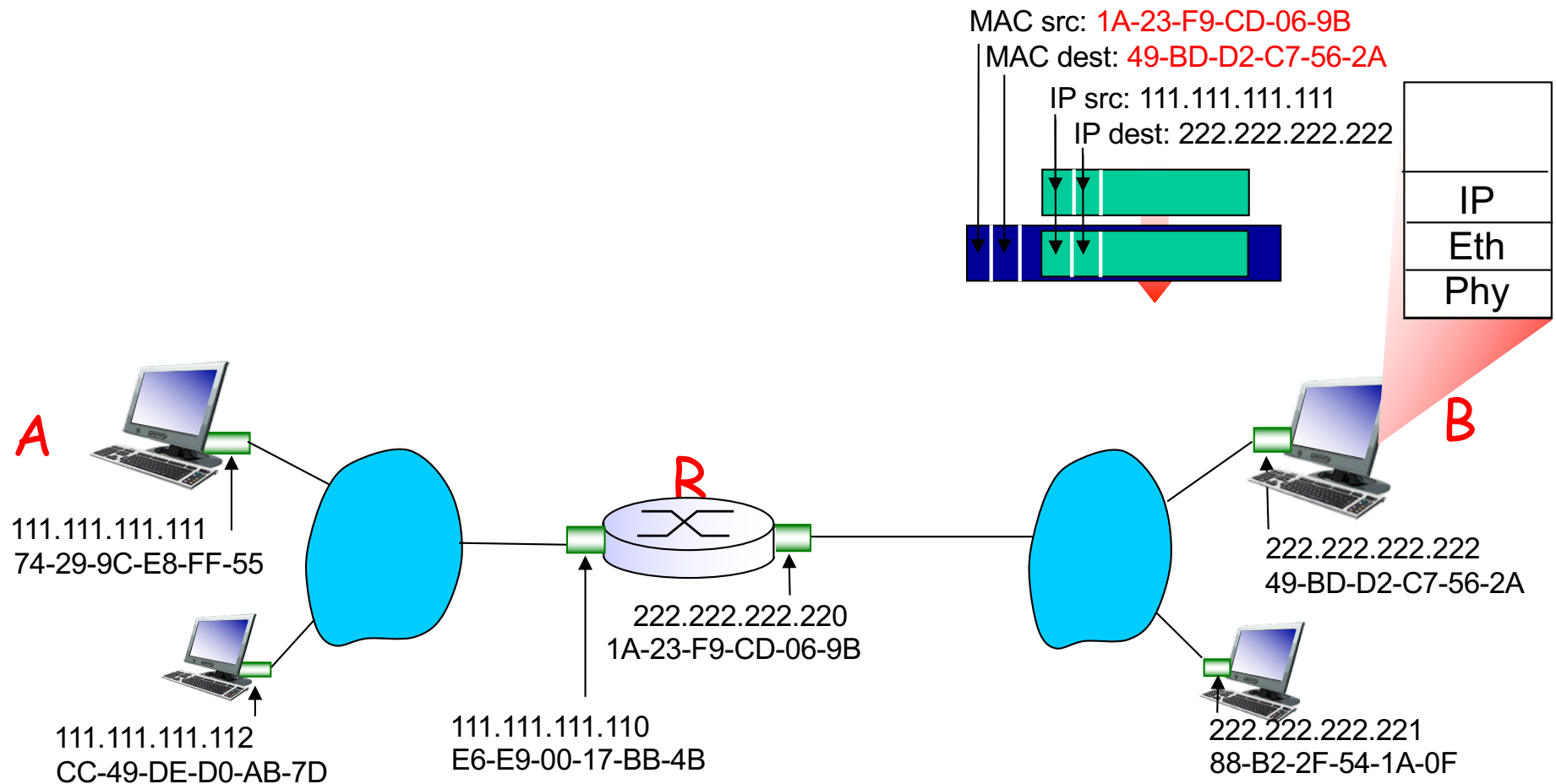
# Invio verso un nodo esterno alla sottorete

- ❖ R inoltra il datagramma con IP sorgente di A e IP destinazione di B
- ❖ R crea un frame con l'indirizzo MAC di B come destinazione (conosciuto con ARP), contenente il datagramma IP A-to-B



# Invio verso un nodo esterno alla sottorete

- ❖ R inoltra il datagramma con IP sorgente di A e IP destinazione di B
- ❖ R crea un frame con l'indirizzo MAC di B come destinazione (conosciuto con ARP), contenente il datagramma IP A-to-B



# Capitolo 6: Livello di collegamento e reti locali

6.1 Livello di collegamento: introduzione e servizi

6.2 Tecniche di rilevazione e correzione degli errori

6.3 Protocolli di accesso multiplo

6.4 Indirizzi a livello di collegamento

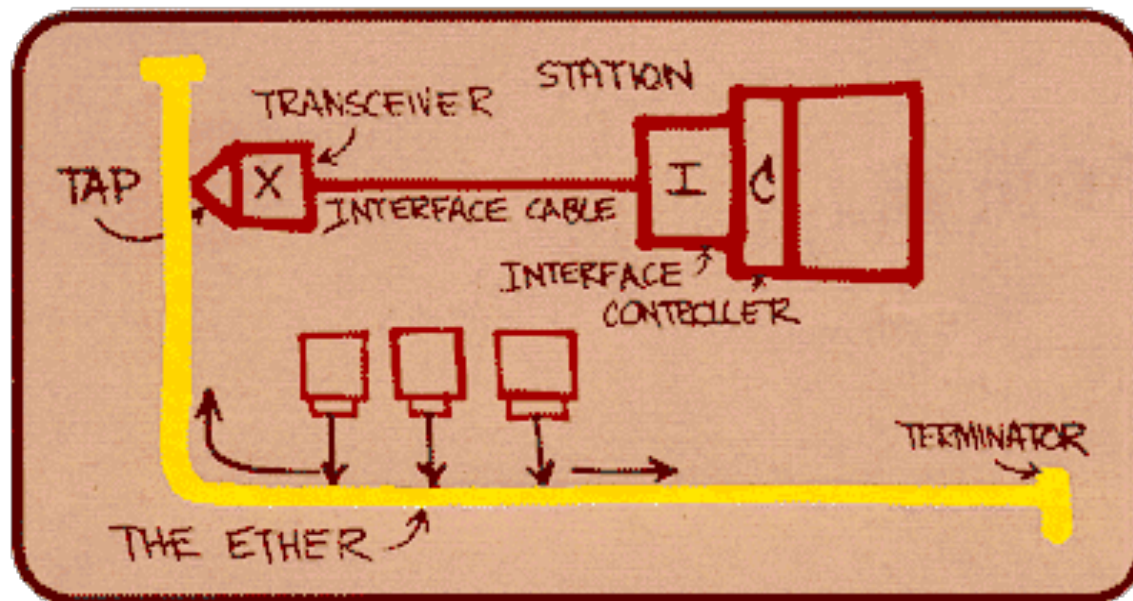
6.5 Ethernet

6.6 Switch a livello di collegamento

# Ethernet

Detiene una posizione dominante nel mercato delle LAN cablate.

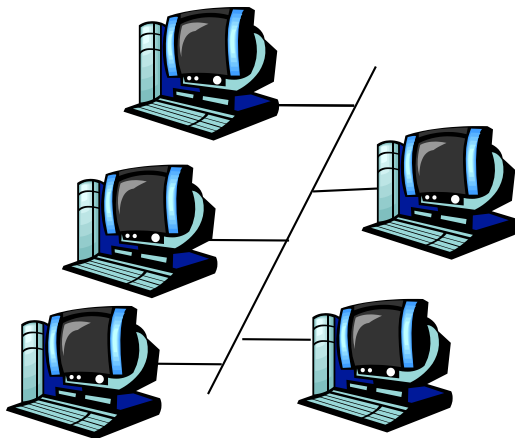
- ❑ È stata la prima LAN ad alta velocità con vasta diffusione.
- ❑ Più semplice e meno costosa di token ring, FDDI e ATM.
- ❑ Sempre al passo dei tempi con il tasso trasmissivo: 10 Mbps - 10 Gbps



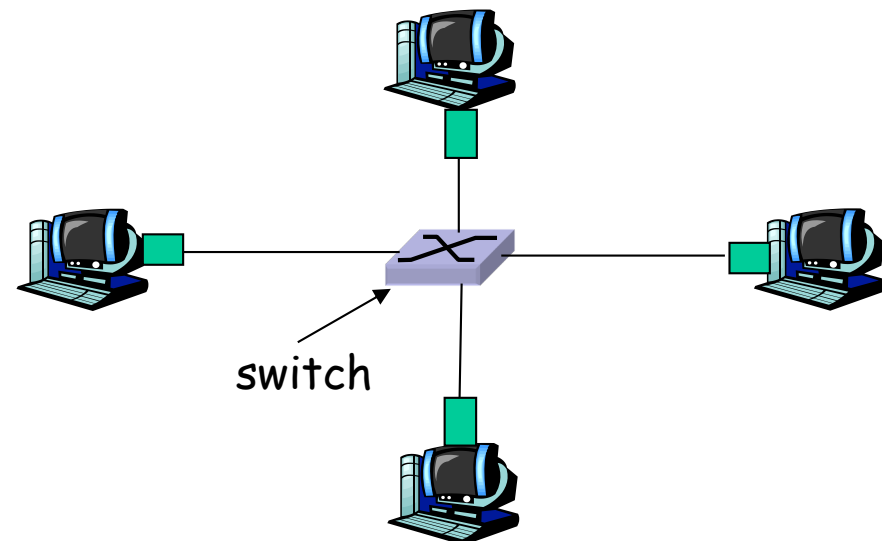
Il progetto originale di Bob Metcalfe che portò allo standard Ethernet.

# Topologia a stella

- ❑ La topologia a bus era diffusa fino alla metà degli anni 90.
- ❑ Quasi tutte le odierne reti Ethernet sono progettate con topologia a stella.
- ❑ Al centro della stella è collocato un hub o commutatore (*switch*).
- ❑ Ciascun nodo è separato e non entra in collisione con gli altri



Bus: cavo coassiale



Topologia a stella



# Struttura dei frame Ethernet

L'adattatore trasmittente incapsula i datagrammi IP in un **frame Ethernet**.



## Preambolo:

- ❑ I frame Ethernet iniziano con un campo di otto byte: sette hanno i bit 10101010 e l'ultimo è 10101011.
- ❑ Servono per "attivare" gli adattatori dei riceventi e sincronizzare i loro orologi con quello del trasmittente.

# Struttura dei frame Ethernet

## ❑ **Indirizzi:** 6 byte

- Quando un adattatore riceve un frame contenente l'indirizzo di destinazione o con l'indirizzo broadcast (es.: un pacchetto ARP), trasferisce il contenuto del campo dati del pacchetto al livello di rete.
- I frame con altri indirizzi MAC vengono ignorati.

## ❑ **Campo tipo:** consente a Ethernet di supportare vari protocolli di rete ("multiplexing" dei protocolli).

## ❑ **Controllo CRC:** consente all'adattatore ricevente di rilevare la presenza di un errore nei bit del frame .

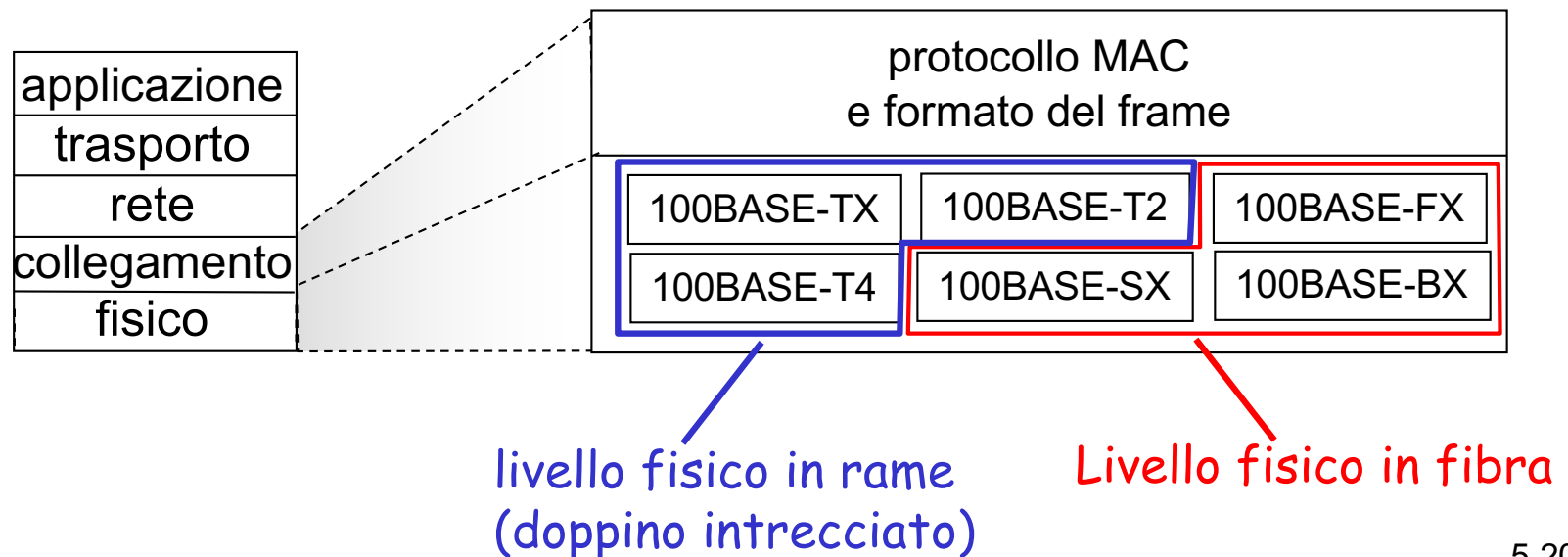


# Servizio senza connessione non affidabile

- ❑ **Senza connessione:** non è prevista nessuna forma di handshake preventiva con il destinatario prima di inviare un pacchetto.
- ❑ **Non affidabile:** l'adattatore ricevente non invia un riscontro né se un pacchetto supera il controllo CRC né in caso contrario.
  - Il flusso dei datagrammi che attraversano il livello di rete può presentare delle lacune.
  - L'applicazione può rilevare le lacune se viene impiegato TCP.
  - Altrimenti, potrebbe accusare problemi a causa dell'incompletezza dei dati.

# Ethernet 802.3 : livelli di collegamento e fisico

- *molte* differenti standard Ethernet
  - protocollo MAC e formato del frame standard
  - differenti velocità: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10G bps
  - differenti mezzi trasmissivi del livello fisico: fibra, cavo



# Capitolo 6: Livello di collegamento e reti locali

6.1 Livello di collegamento: introduzione e servizi

6.2 Tecniche di rilevazione e correzione degli errori

6.3 Protocolli di accesso multiplo

6.4 Indirizzi a livello di collegamento

6.5 Ethernet

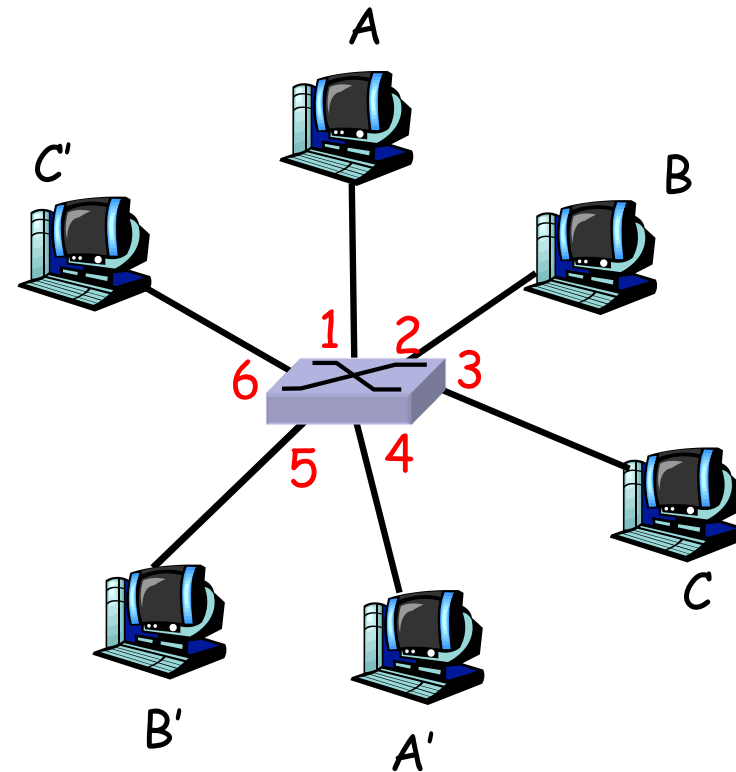
6.6 Switch a livello di collegamento

# Switch

- ❑ Dispositivo del livello di link: più intelligente di un hub, svolge un ruolo *attivo*
  - Filtra e inoltra i frame Ethernet.
  - Esamina l'indirizzo di destinazione e lo invia all'interfaccia corrispondente alla sua destinazione.
  - Quando un pacchetto è stato inoltrato nel segmento, usa CSMA/CD per accedere al segmento.
- ❑ Trasparente
  - Gli host sono inconsapevoli della presenza di switch.
- ❑ Plug-and-play, autoapprendimento
  - Gli switch non hanno bisogno di essere configurati.

# Switch: consente più trasmissioni simultanee

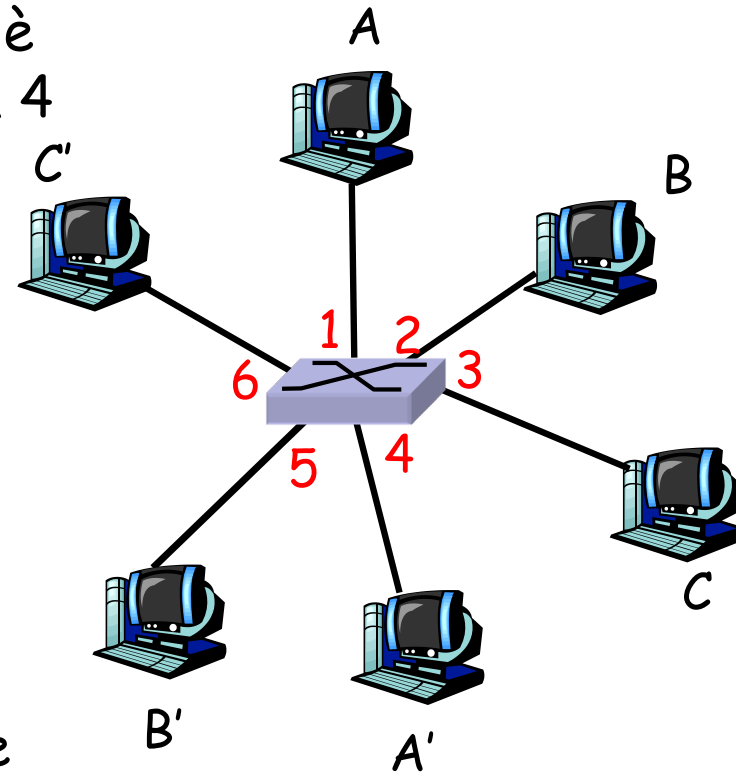
- ❑ Gli host hanno collegamenti dedicati e diretti con lo switch
- ❑ Gli switch bufferizzano i pacchetti
- ❑ Il protocollo Ethernet è usato su ciascun collegamento in entrata, ma non si verificano collisioni; full duplex
- ❑ *switching*: da A ad A' e da B a B' simultaneamente, senza collisioni
  - Non possibile con gli hub "stupidi"



*switch con sei interfacce  
(1,2,3,4,5,6)*

# Tabella di commutazione

- D: come fa lo switch a sapere che A' è raggiungibile attraverso l'interfaccia 4 e B' attraverso l'interfaccia 5?
- R: ogni switch ha una **tabella di commutazione** (*switch table*), e ciascuna voce contiene:
  - (indirizzo MAC del nodo, interfaccia che conduce al nodo, time stamp)
- Assomiglia a una tabella d'instradamento!
- D: come si creano e si mantengono le voci di una tabella di commutazione?



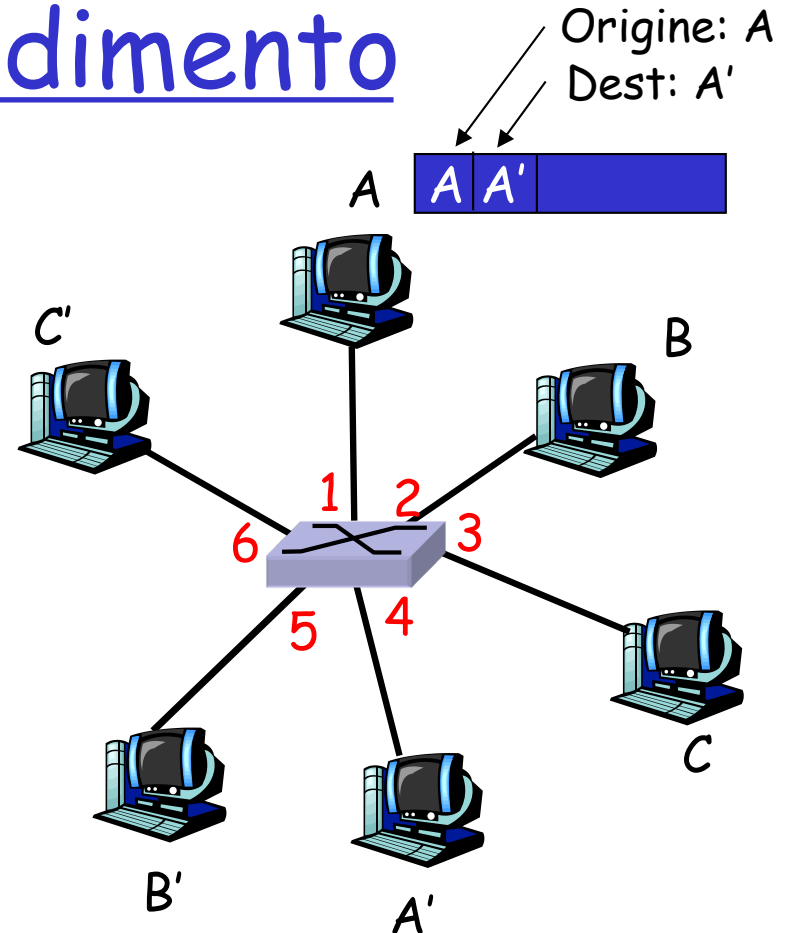
*switch con sei interfacce  
(1,2,3,4,5,6)*



# Switch: autoapprendimento

- Lo switch *apprende* quali nodi possono essere raggiunti attraverso determinate interfacce
  - quando riceve un pacchetto, lo switch "impara" l'indirizzo del mittente
  - registra la coppia mittente/indirizzo nella sua tabella di commutazione

| Indir. MAC | Interfaccia | TTL       |
|------------|-------------|-----------|
| <i>A</i>   | <i>1</i>    | <i>60</i> |
|            |             |           |



*Tabella di commutazione  
(inizialmente vuota)*

# Switch: filtraggio e inoltra

Quando uno switch riceve un frame:

1. Registra il collegamento associato con l'host mittente
2. Indicizza la tabella utilizzando gli indirizzi MAC

**if** entry found for destination  
**then**{

**if** dest on segment from which frame arrived  
**then** drop the frame

**else** forward the frame on interface indicated

}

**else** flood

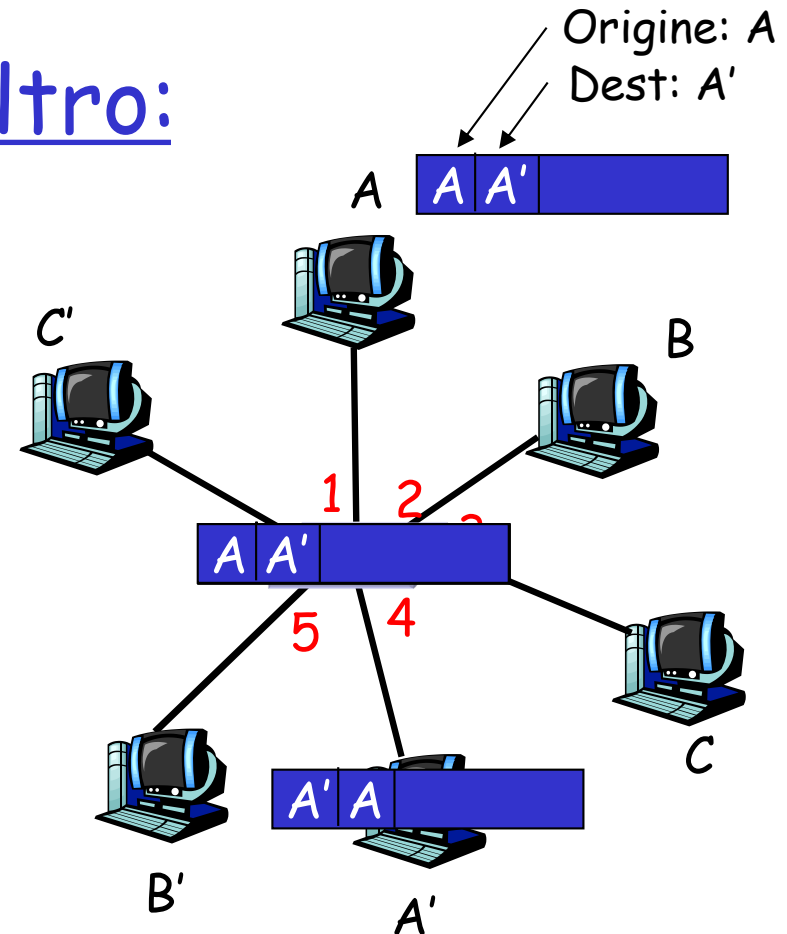


*Lo inoltra a tutti tranne all'interfaccia  
dalla quale è arrivato il pacchetto*

# Autoapprendimento, inoltre: un esempio

- ❑ Destinazione del frame ignota: *flood*
- ❑ destinazione A, location nota: *selective send*

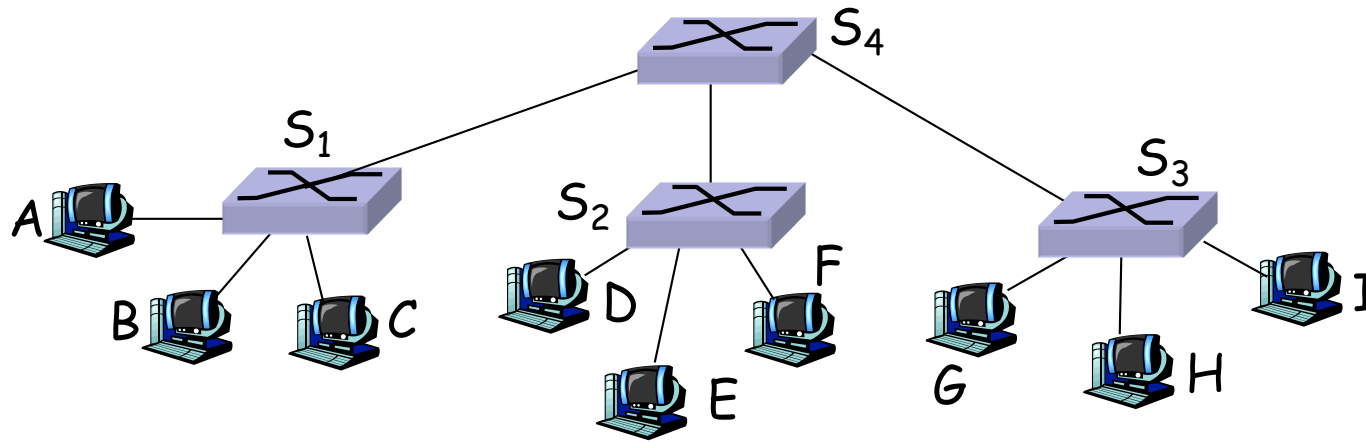
| Indir. MAC | Interfaccia | TTL |
|------------|-------------|-----|
| A          | 1           | 60  |
| A'         | 4           | 60  |



*Tabella di commutazione  
(inizialmente vuota)*

# Collegare gli switch

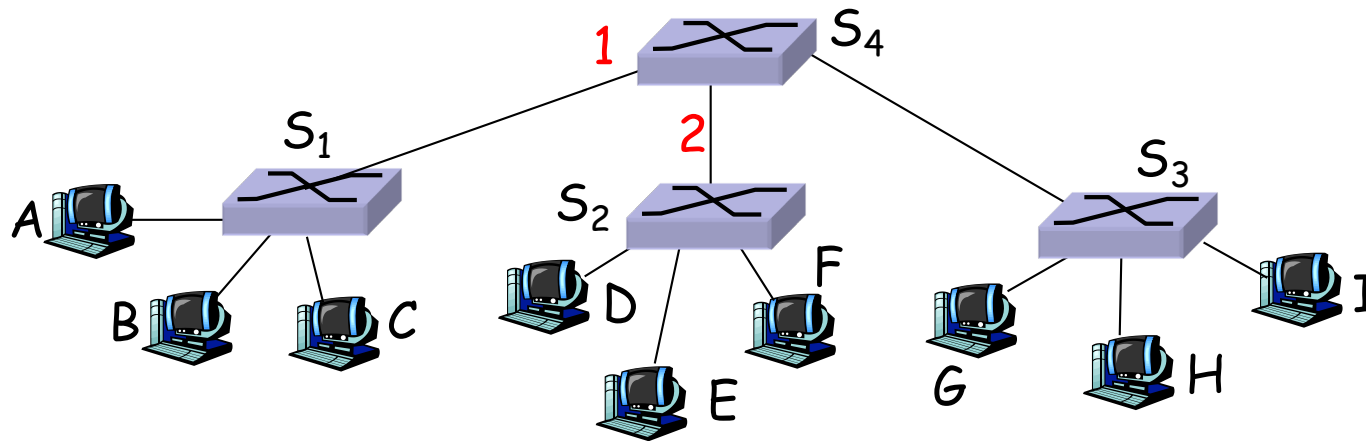
- Gli switch possono essere interconnessi



- D: per inviare da A a G, come fa S<sub>1</sub> a sapere che deve inoltrare il frame attraverso S<sub>4</sub> e S<sub>3</sub>?
- A: autoapprende! (funziona esattamente come nel caso di un singolo switch!)

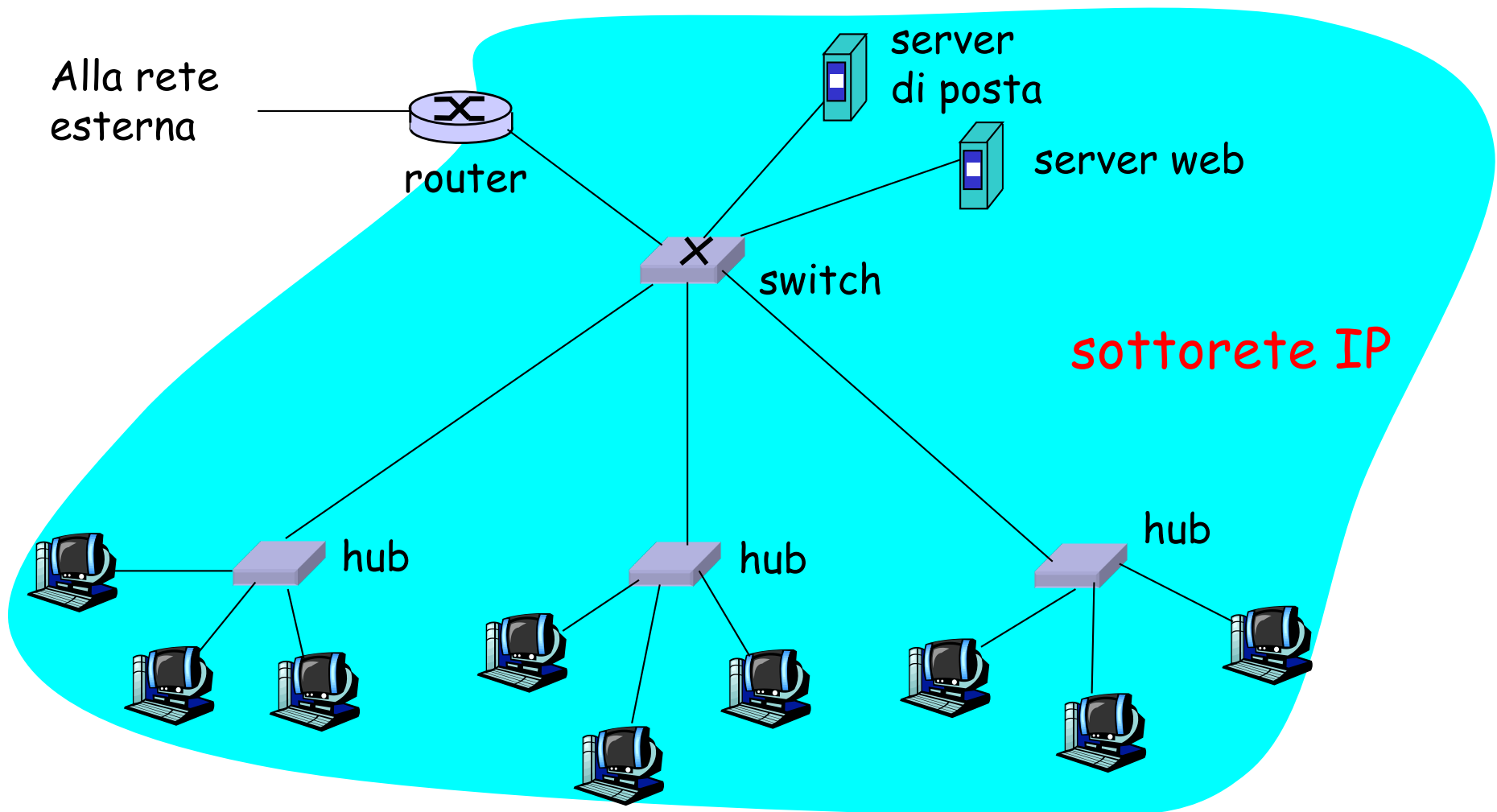
# Esempio di apprendimento multi-switch

Supponiamo che C invii un frame a I, e che I risponda a C



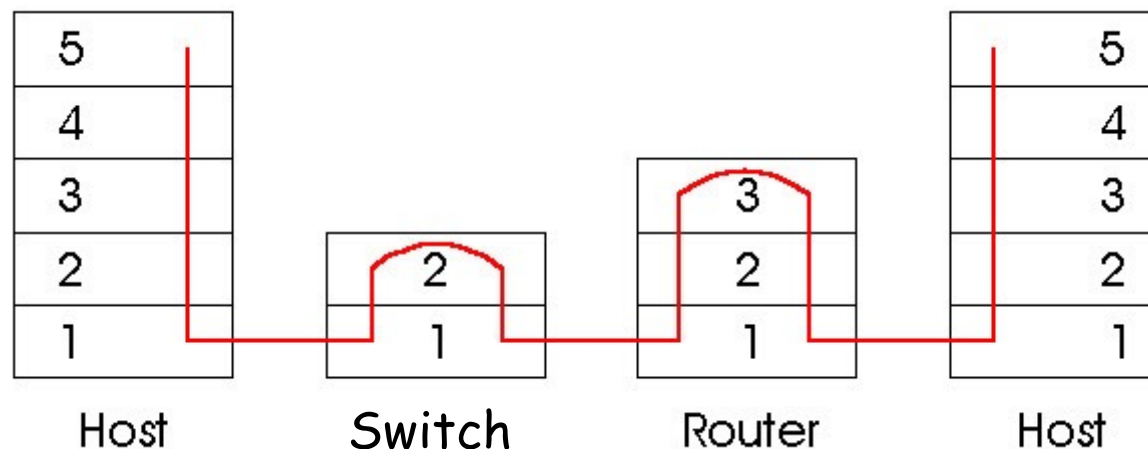
D: illustrate le tabelle di commutazione e l'inoltro dei pacchetti in  $S_1$ ,  $S_2$ ,  $S_3$ ,  $S_4$

# Esempio di rete di un'istituzione



# Switch e router a confronto

- ❑ Entrambi sono dispositivi store-and-forward
  - router: dispositivi a livello di rete (esaminano le intestazioni del livello di rete)
  - switch: dispositivi a livello di collegamento
- ❑ I router mantengono tabelle d'inoltro e implementano algoritmi d'instradamento
- ❑ Gli switch mantengono tabelle di commutazione e implementano il filtraggio e algoritmi di autoapprendimento

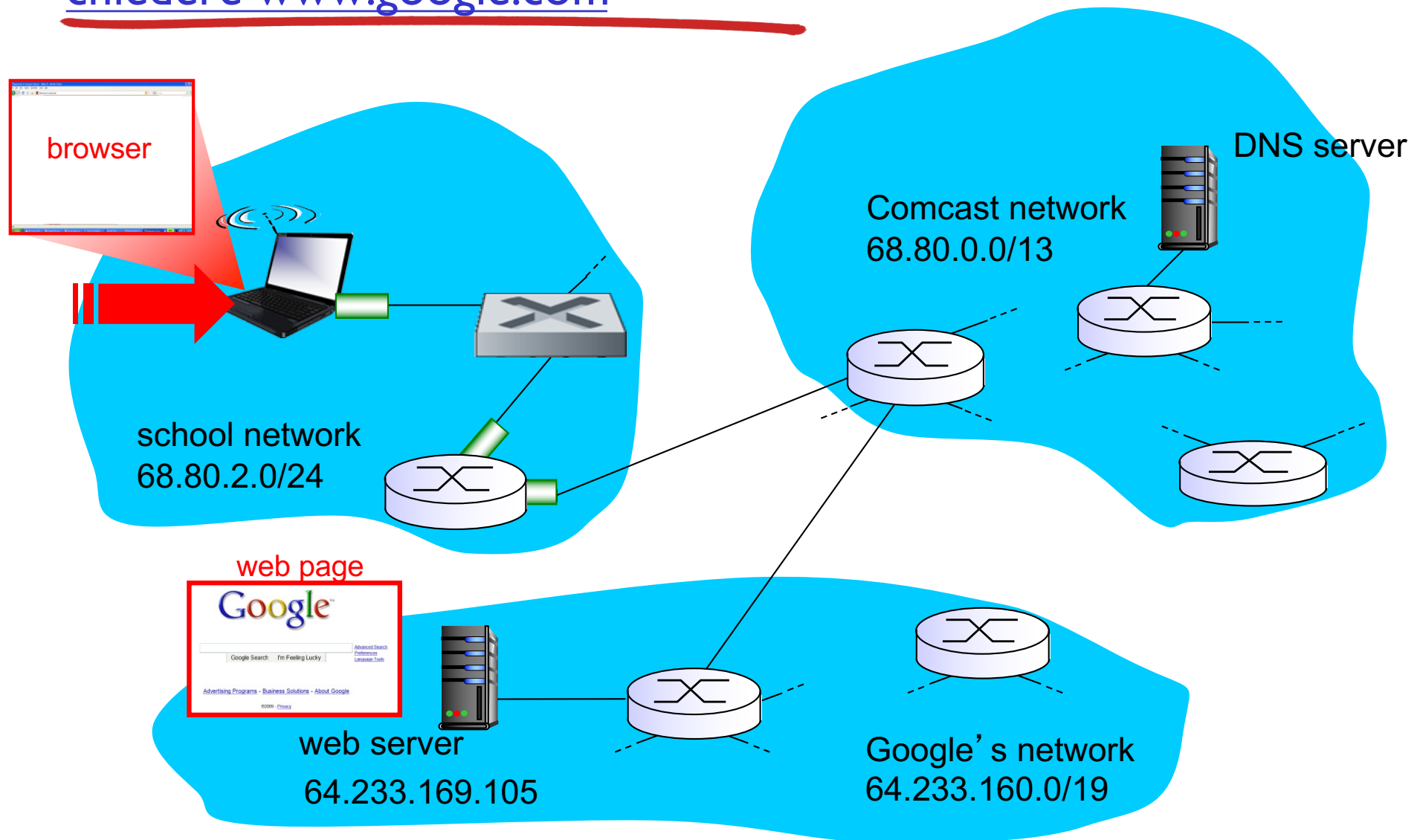


# Riepilogo del viaggio nella pila di protocolli

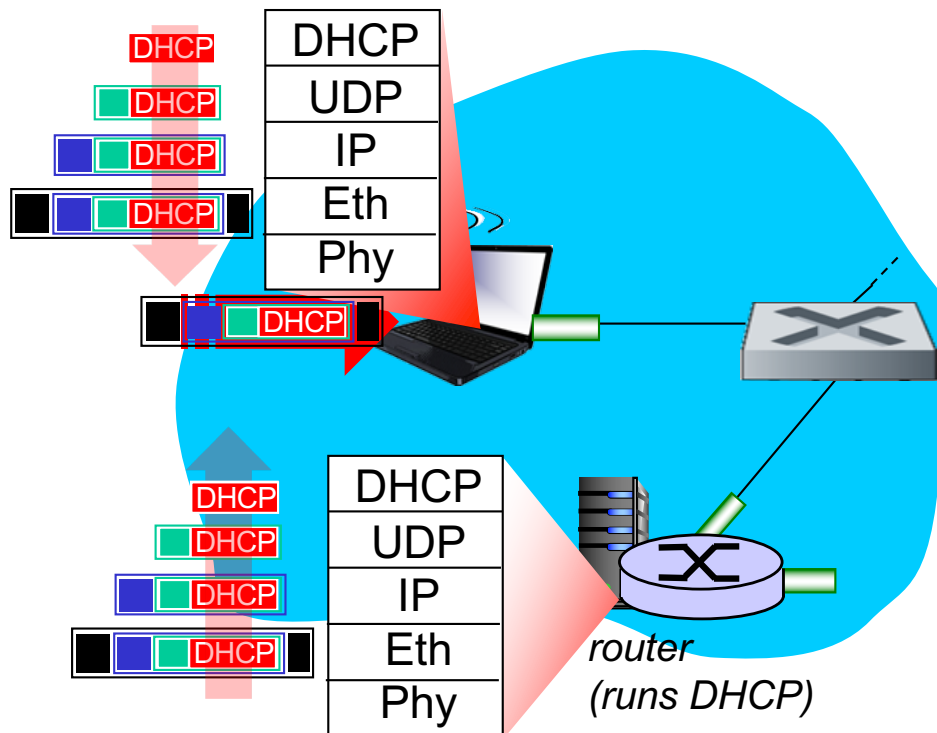
Richiesta di una pagina web



# Scenario: uno studente accende il portatile all'università per chiedere [www.google.com](http://www.google.com)



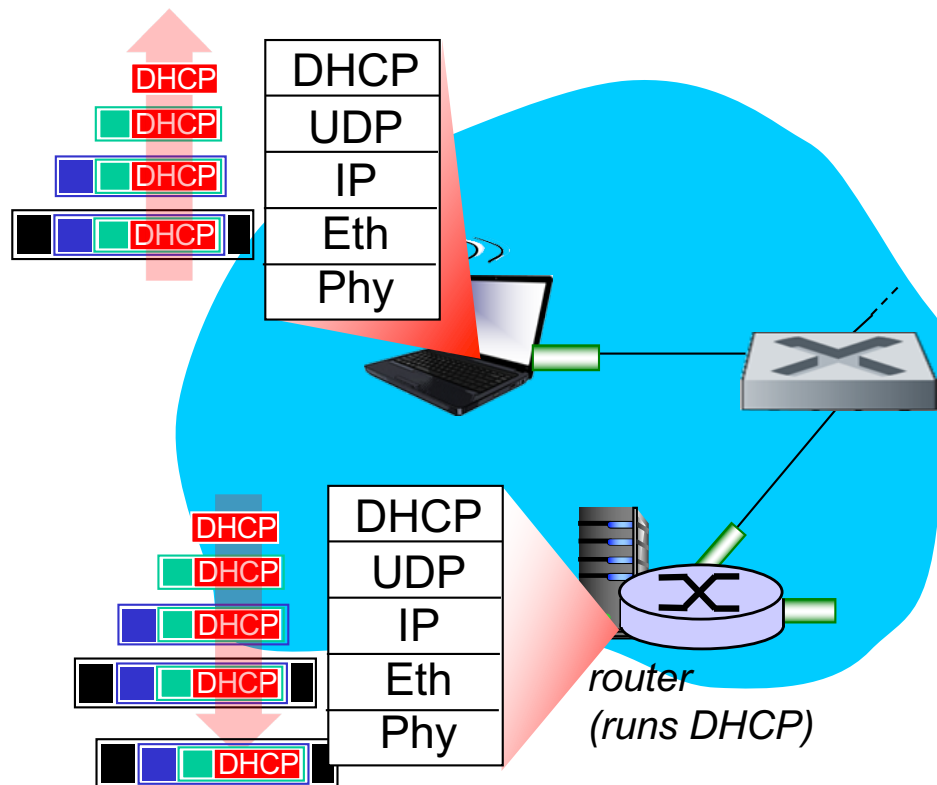
# A day in the life... connecting to the Internet



- r Il portatile ha bisogno di un suo indirizzo IP address, dell'indirizzo del primo router e dell'indirizzo del server DNS => **DHCP**

- ❖ La richiesta DHCP è incapsulata in **UDP**, incapsulato in **IP**, incapsulato in **802.3** Ethernet
- ❖ Il frame Ethernet viene inviato in **broadcast** (dest: FFFFFFFFFFFFFFFF) sulla lan LAN, e ricevuta da tutte le interfacce di rete.
- ❖ Solo il router che esegue il server **DHCP** elabora la richiesta
- ❖ Ethernet fa il demultiplexing verso IP, che fa il demultiplexing verso UDP, che fa il demultiplexing verso DHCP

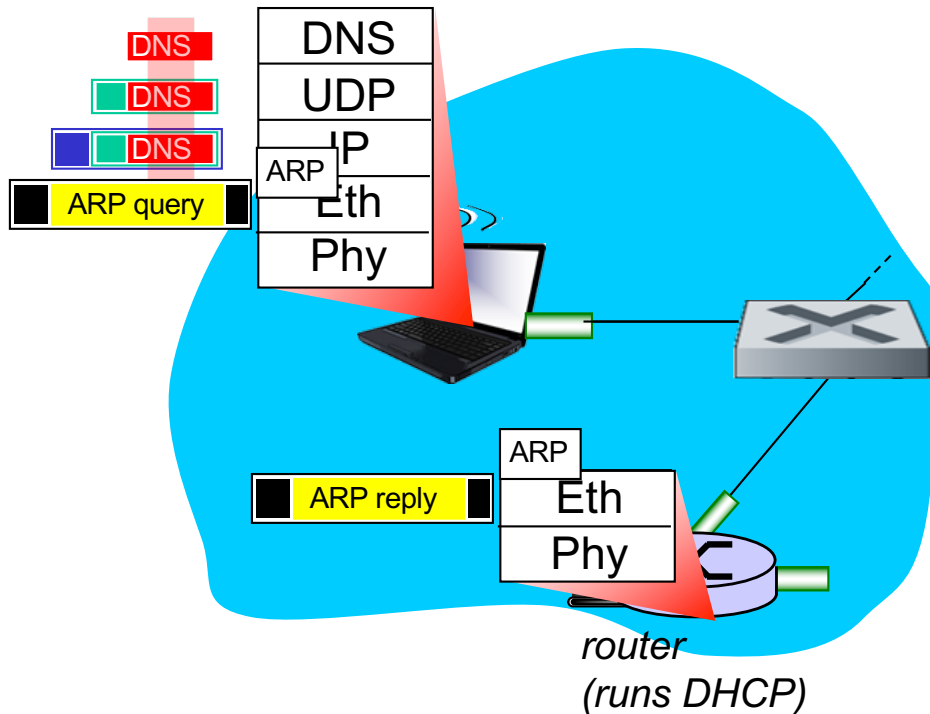
# A day in the life... connecting to the Internet



- r IL server DHCP prepara un **DHCP ACK** contenente l'indirizzo IP dell'host, l'indirizzo IP del primo router, nome & indirizzo IP del server DNS
- ❖ Il server DHCP incapsula la richiesta in UDP, poi IP, poi Ethernet
- ❖ Il frame viene inviato tramite la LAN (**switch learning**) fino all'host
- ❖ IL DHCP client riceve la risposta DHCP ACK

*L'host adesso ha un indirizzo IP, conosce nome e indirizzo del server DNS,  
Conosce l'indirizzo IP del primo router*

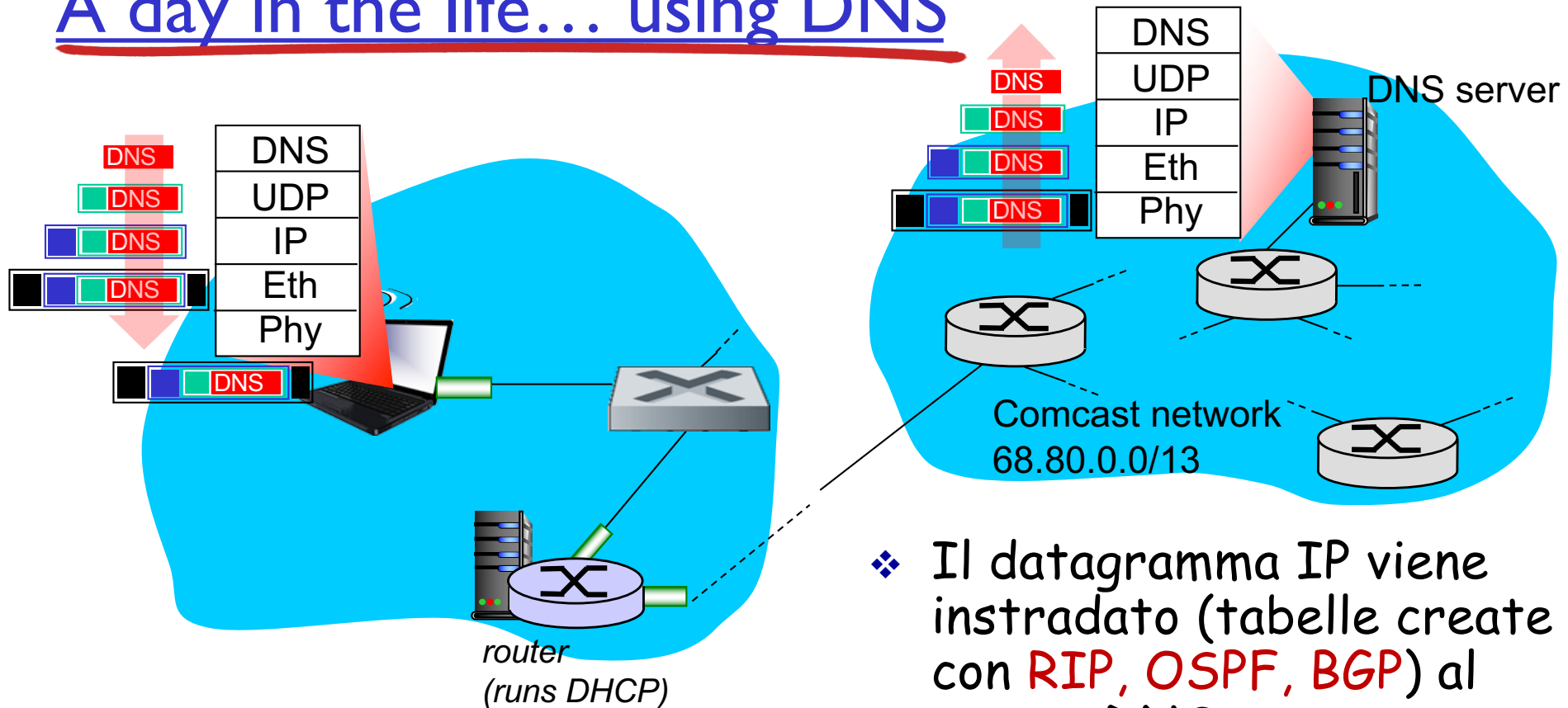
# A day in the life... ARP (before DNS, before HTTP)



- r Prima di inviare la richiesta **HTTP** serve l'indirizzo IP di `www.google.com` => **DNS**

- ❖ DNS query creata, incapsulata in UDP, incapsulata in IP, incapsulata in Eth.
- ❖ Per inviare il frame al router serve l'indirizzo MAC dell'interfaccia del router => **ARP**
- ❖ **ARP query** inviata in broadcast, ricevuta dal router, che risponde con una **ARP reply** dando l'indirizzo MAC della propria interfaccia
- ❖ Adesso l'host può inviare la richiesta DNS

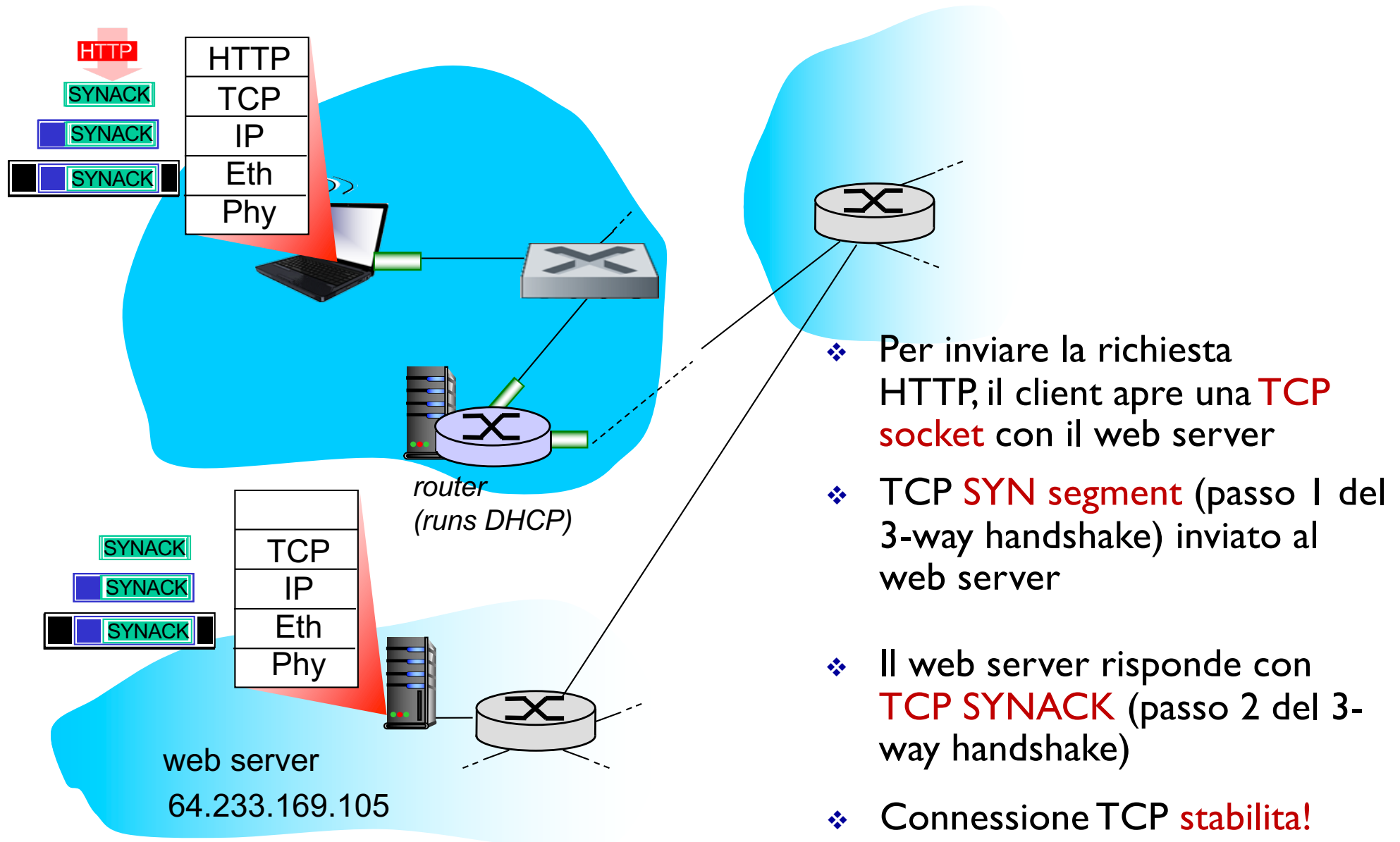
# A day in the life... using DNS



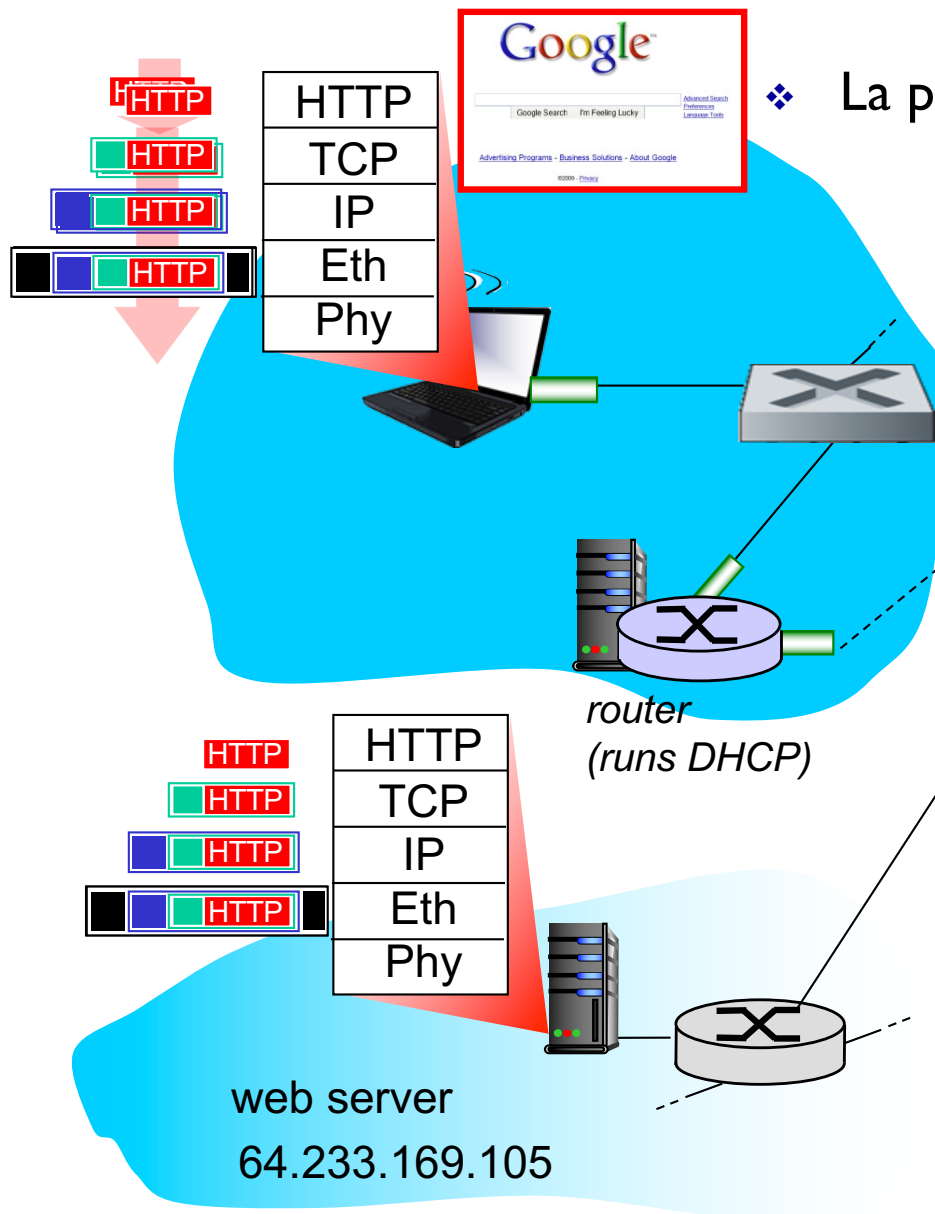
- ❖ Il datagramma IP contenente la richiesta viene inoltrata tramite il LAN switch dall'host al router

- ❖ Il datagramma IP viene instradato (tabelle create con **RIP, OSPF, BGP**) al server DNS
- ❖ Demultiplexing presso il server DNS
- ❖ Il server DNS risponde all'host con l'indirizzo di [www.google.com](http://www.google.com)

# A day in the life...TCP connection carrying HTTP



# A day in the life... HTTP request/reply



❖ La pagina web è **visualizzata (!!!)**

- ❖ **Richiesta HTTP** inviata tramite la TCP socket
- ❖ Il datagramma IP contiene la richiesta HTTP viene instradato fino a [www.google.com](http://www.google.com)
- ❖ Il web server risponde con una risposta **HTTP** (contenete la pagina web)
- ❖ I datagrammi IP che contengono la risposta HTTP sono instradati al client