

# Reti di Elaboratori

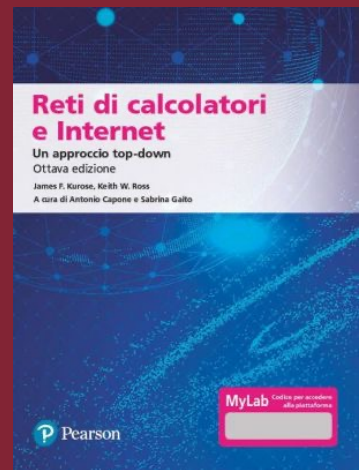
Livello di Collegamento: LAN



SAPIENZA  
UNIVERSITÀ DI ROMA

Alessandro Checco

[alessandro.checco@uniroma1.it](mailto:alessandro.checco@uniroma1.it)



Capitolo 6

# Livello di collegamento e LAN: sommario

- introduzione
- rilevamento e correzione degli errori
- protocolli di accesso multiplo
- LAN
  - indirizzamento, ARP
  - Ethernet
  - switch
  - VLAN
- virtualizzazione dei collegamenti: MPLS
- data center
- un giorno nella vita di una richiesta web

# indirizzi MAC

- Indirizzo IP a 32 bit:
  - indirizzo dell'interfaccia del *livello di rete*
  - utilizzato per il forwarding di livello 3 (livello di rete).
  - es: 128.119.40.136
- Indirizzo MAC (o LAN o fisico o Ethernet):
  - funzione: utilizzato "localmente" per ottenere frame da un'interfaccia a un'altra interfaccia fisicamente connessa (stessa sottorete, nel senso dell'indirizzamento IP)
  - Indirizzo MAC a 48 bit (per la maggior parte delle LAN) hardcoded nella ROM NIC, a volte anche impostabile via software
  - es: 1A-2F-BB-76-09-AD

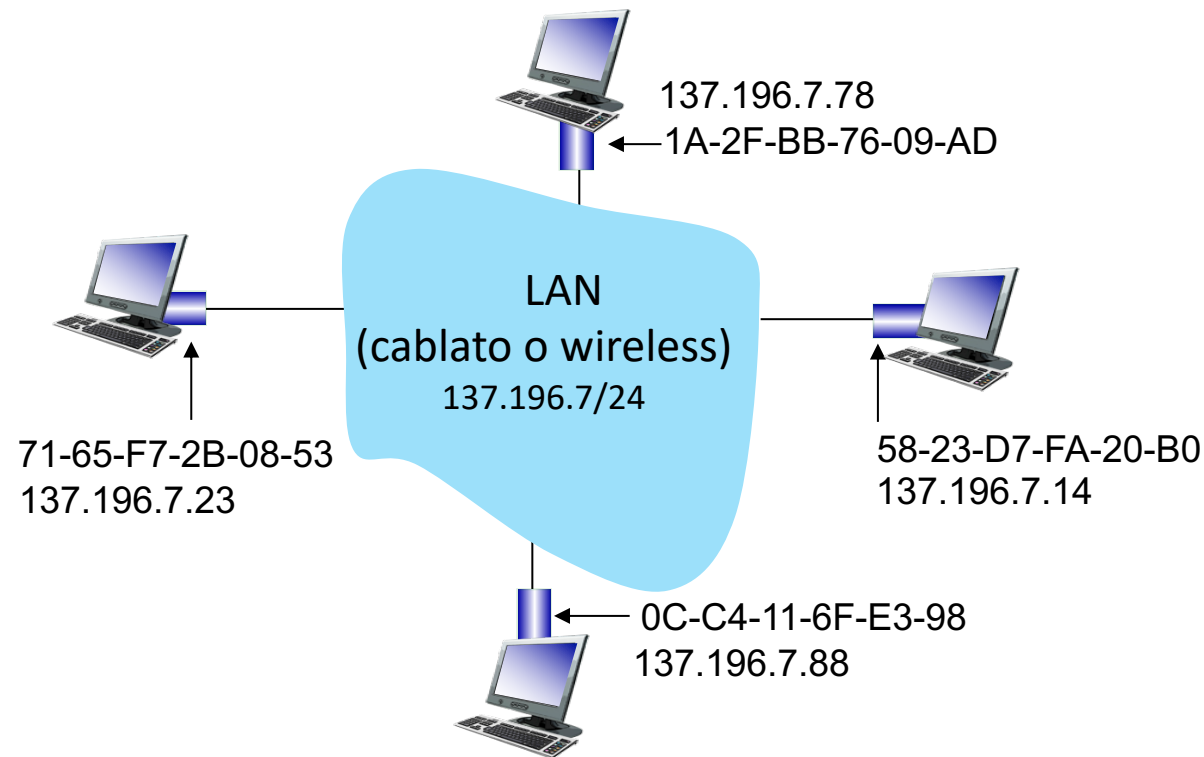
notazione esadecimale (base 16).  
(ogni carattere rappresenta 4 bit)



# indirizzi MAC

ogni interfaccia sulla Local Area Network ha:

- un indirizzo **MAC** univoco a 48 bit
- un indirizzo IP a 32 bit localmente univoco (come abbiamo visto)

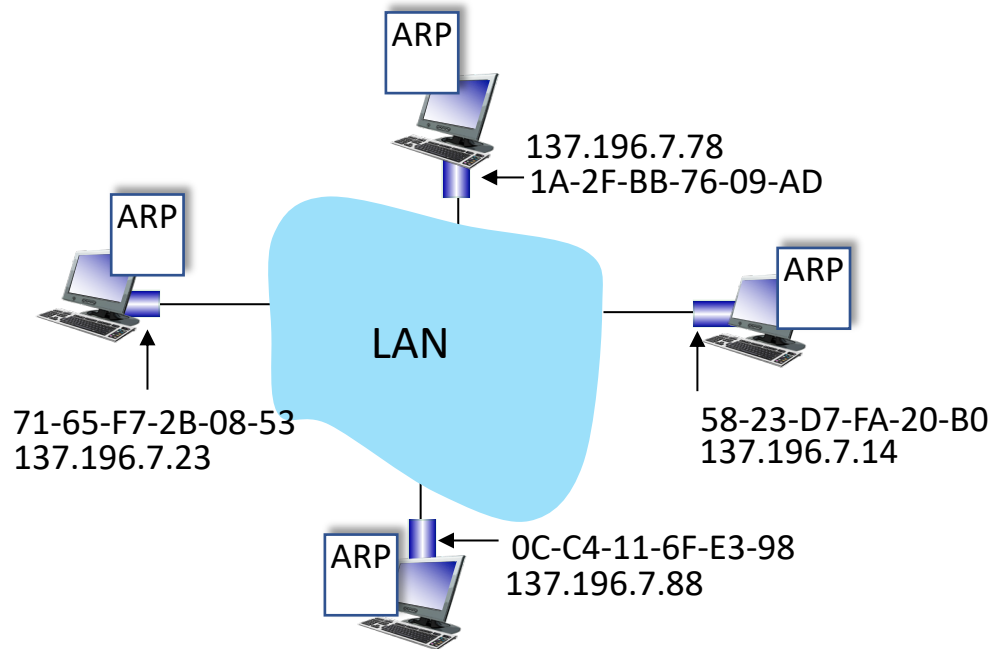


# indirizzi MAC

- Allocazione degli indirizzi MAC gestita da IEEE
- il produttore acquista parte dello spazio degli indirizzi MAC (per garantire l'unicità) OUI (Organizational Unique Identifier, primi 12 bit, aziende grandi può averne molti)
- analogia:
  - Indirizzo MAC: come il codice fiscale (non associato alla posizione)
  - Indirizzo IP: come indirizzo postale
- Indirizzo MAC flat: portabilità
  - si può spostare l'interfaccia da una LAN all'altra
  - ricordiamo che l'indirizzo IP *non* è portabile: dipende dalla sottorete IP a cui è collegato il nodo

# ARP: protocollo di risoluzione degli indirizzi

*Domanda:* come determinare l'indirizzo MAC dell'interfaccia, conoscendone l'indirizzo IP?



**Tabella ARP:** ogni nodo IP (host, router) sulla LAN ha una tabella

- Mappature IP/MAC per nodi LAN:  
< indirizzo IP; Indirizzo MAC; TTL >
- TTL (Time To Live): tempo trascorso il quale la mappatura verrà dimenticata (in genere 20 min)

# Protocollo ARP in azione

esempio: A vuole inviare un datagramma a B

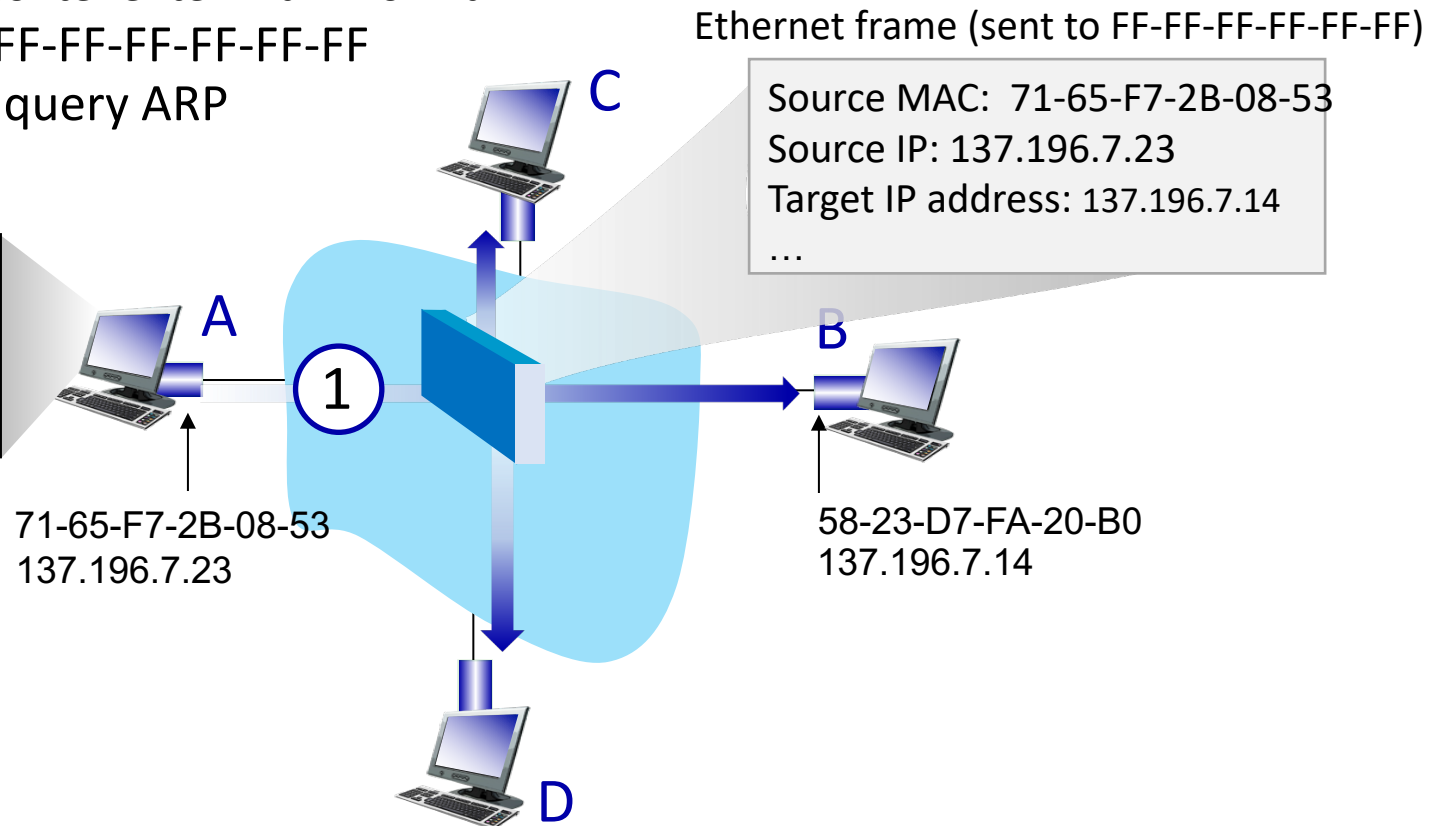
- L'indirizzo MAC di B non è nella tabella ARP di A, quindi A utilizza ARP per trovare l'indirizzo MAC di B

A trasmette in broadcast la query ARP, contenente l'indirizzo IP di B

- ①
- indirizzo MAC di destinazione = FF-FF-FF-FF-FF-FF
  - tutti i nodi sulla LAN ricevono la query ARP

ARP table in A

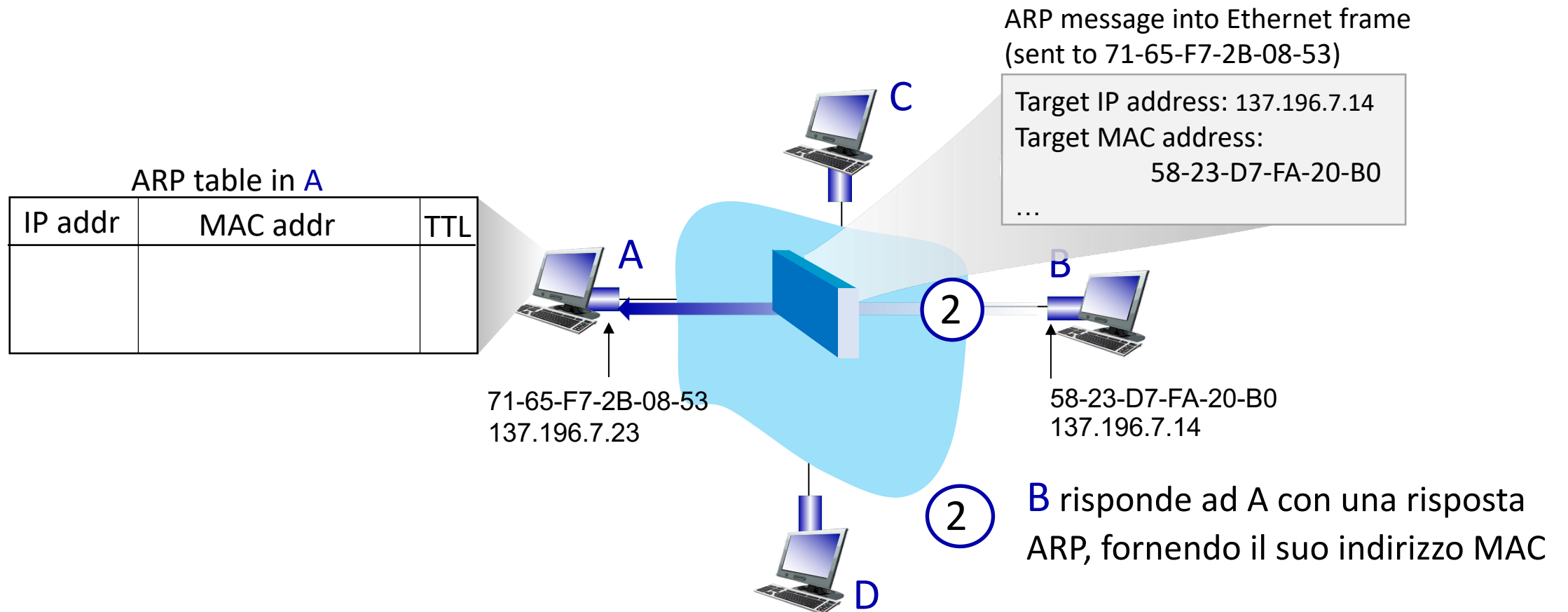
IP addr	MAC addr	TTL



# Protocollo ARP in azione

esempio: A vuole inviare un datagramma a B

- L'indirizzo MAC di B non è nella tabella ARP di A, quindi A utilizza ARP per trovare l'indirizzo MAC di B

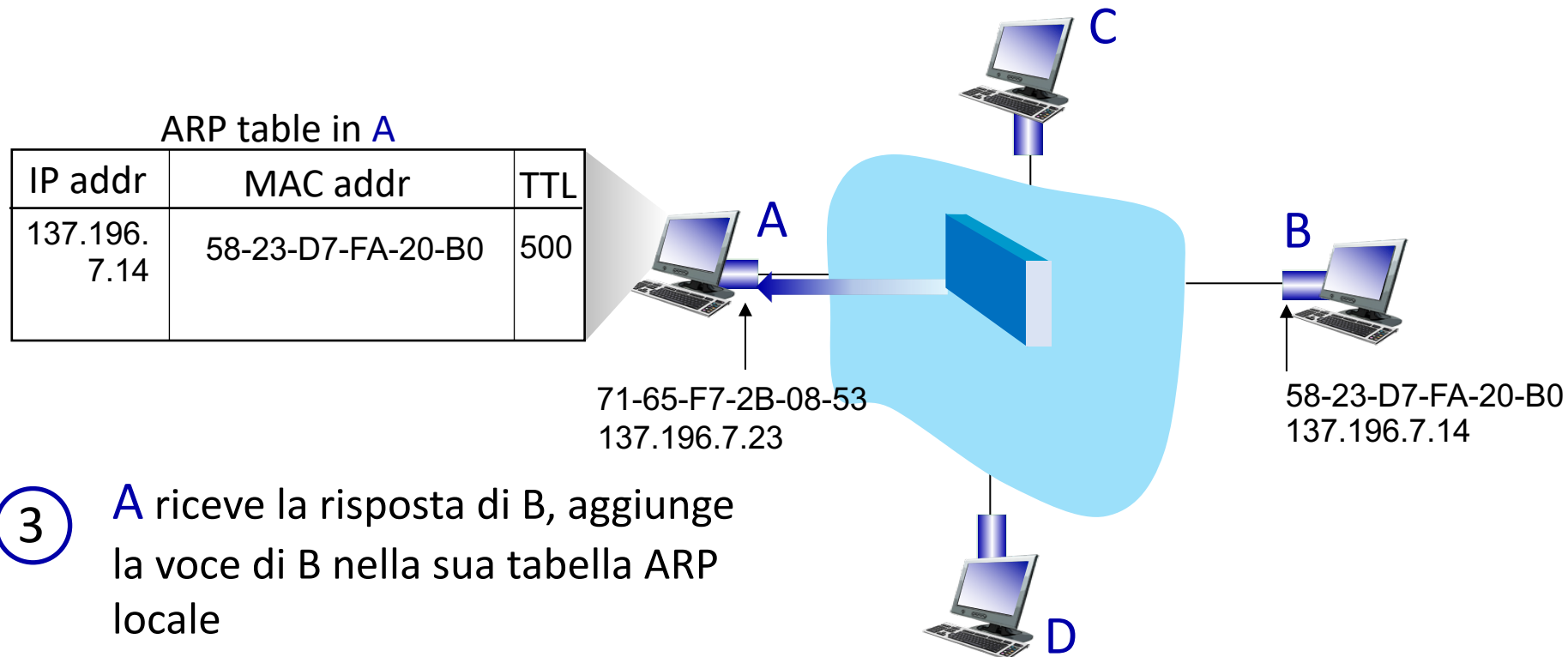




# Protocollo ARP in azione

esempio: A vuole inviare un datagramma a B

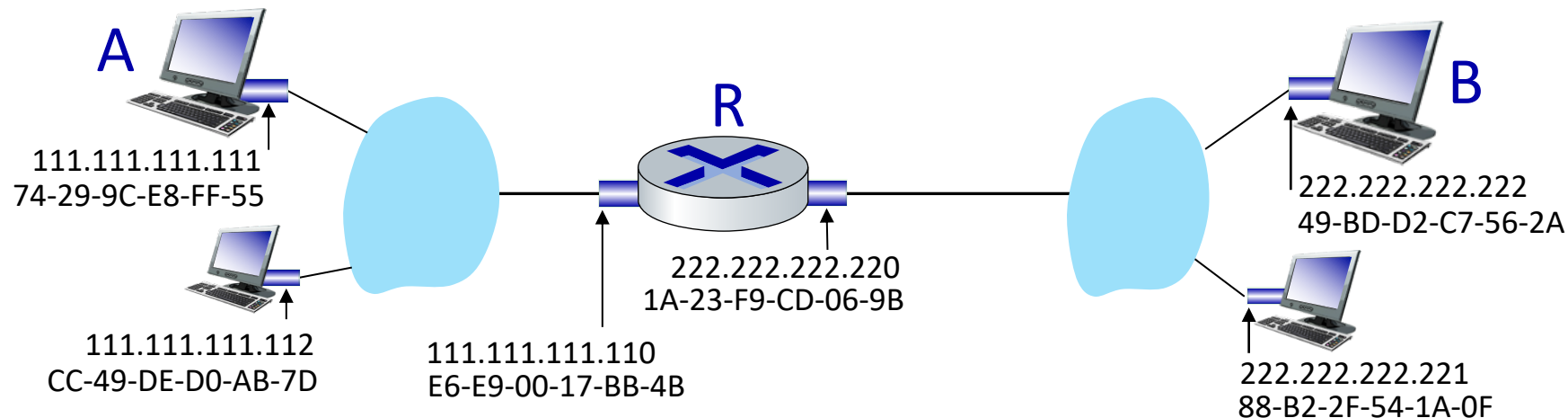
- L'indirizzo MAC di B non è nella tabella ARP di A, quindi A utilizza ARP per trovare l'indirizzo MAC di B



# Instradamento verso un'altra sottorete: indirizzamento

invio di un datagramma da A a B tramite R

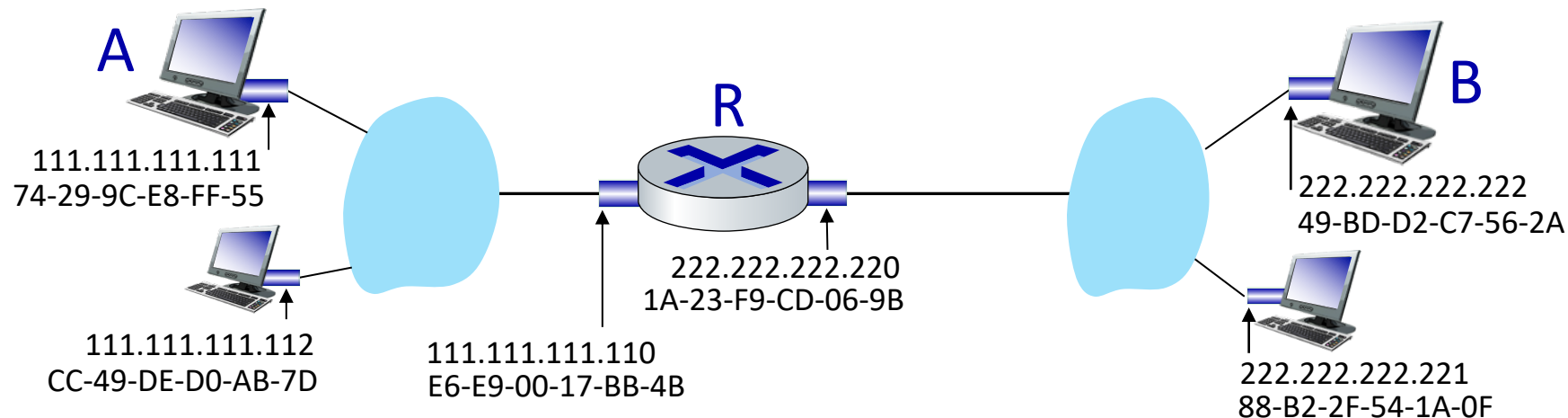
- qual è l'indirizzamento a livello IP (datagramma) e livello MAC (frame)?
- assumiamo che:
  - A conosce l'indirizzo IP di B
  - A conosce l'indirizzo IP del router del primo hop, R (come?)
  - A conosce l'indirizzo MAC di R (come?)



# Instradamento verso un'altra sottorete: indirizzamento

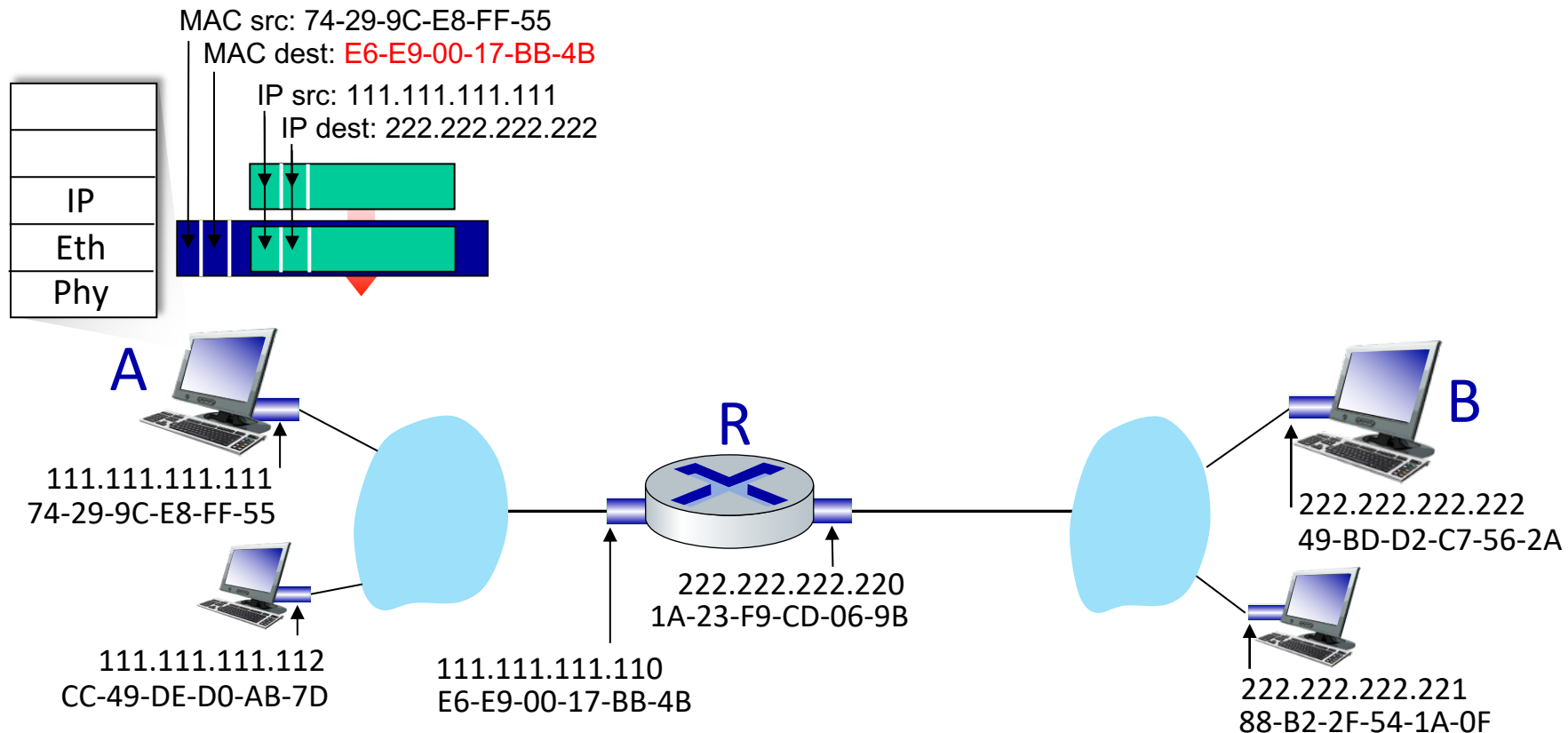
invio di un datagramma da A a B tramite R

- qual è l'indirizzamento a livello IP (datagramma) e livello MAC (frame)?
- assumiamo che:
  - A conosce l'indirizzo IP di B
  - A conosce l'indirizzo IP del router del primo hop, R (DHCP)
  - A conosce l'indirizzo MAC di R (ARP request con IP di gateway noto)



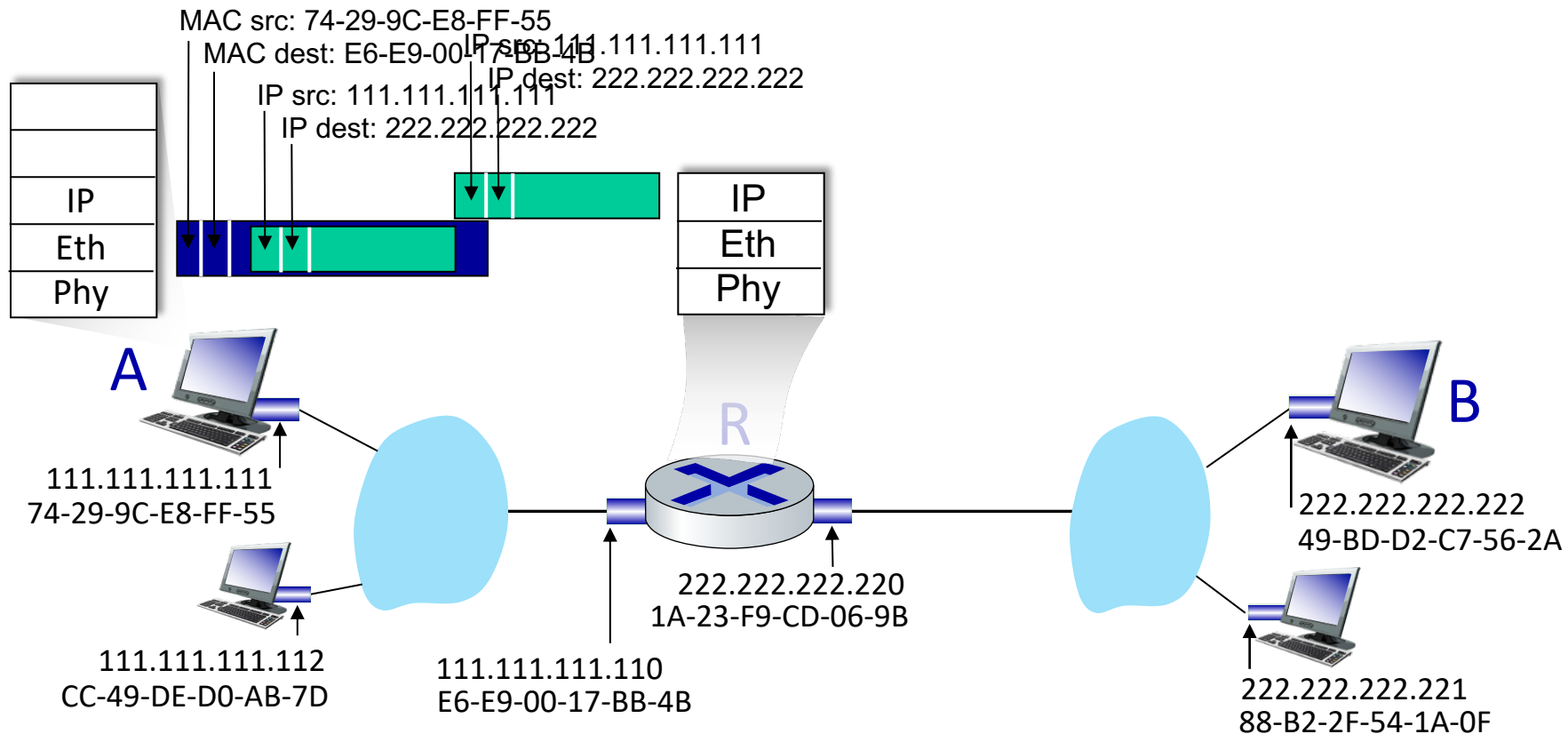
# Instradamento verso un'altra sottorete: indirizzamento

- A crea il datagramma IP con sorgente IP A, destinazione B
- A crea un frame (link layer) contenente il datagramma IP da A a B
  - il MAC di destinazione è quello di R!



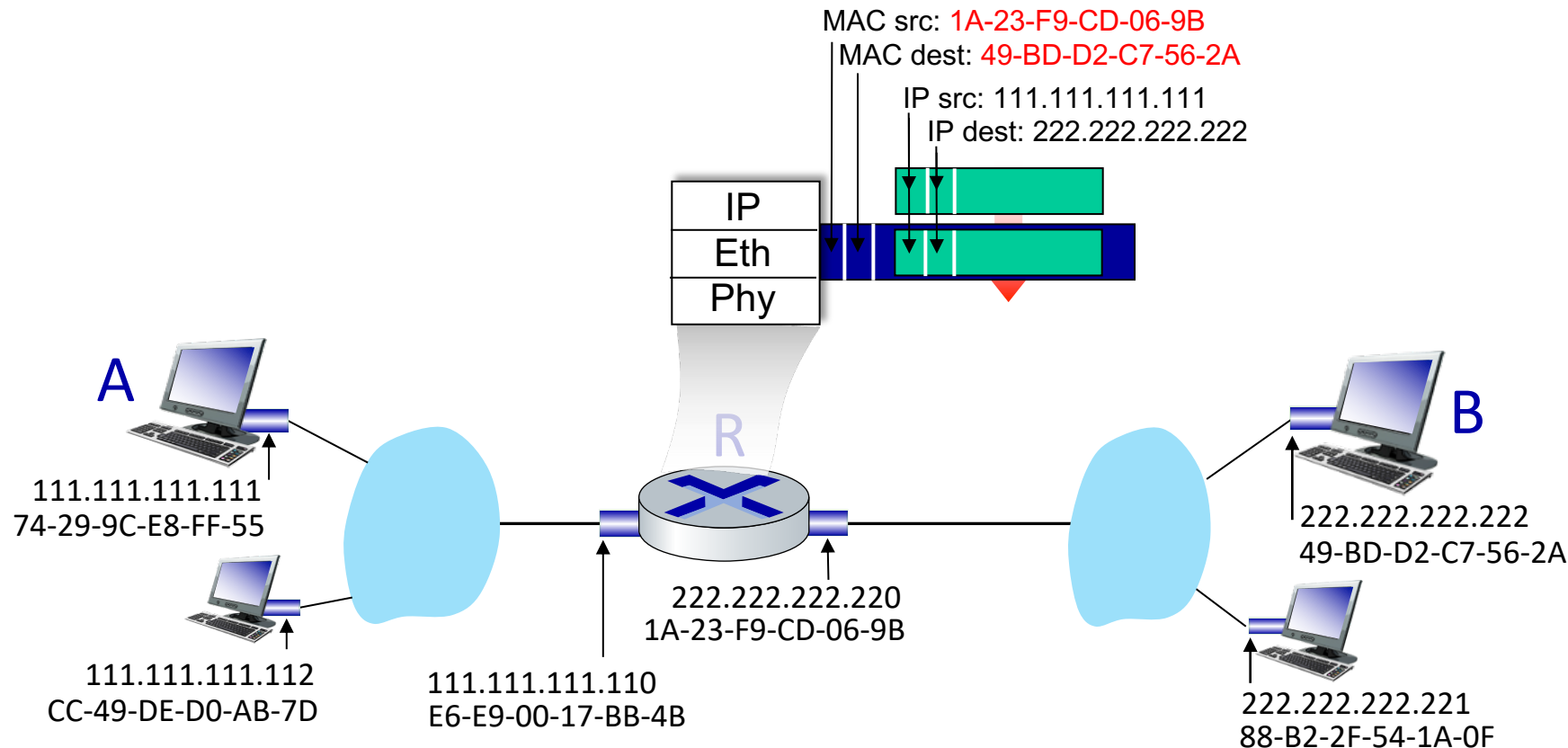
# Instradamento verso un'altra sottorete: indirizzamento

- Frame inviato da A a R
- R riceve il frame, estrae il datagramma, inviato al livello IP



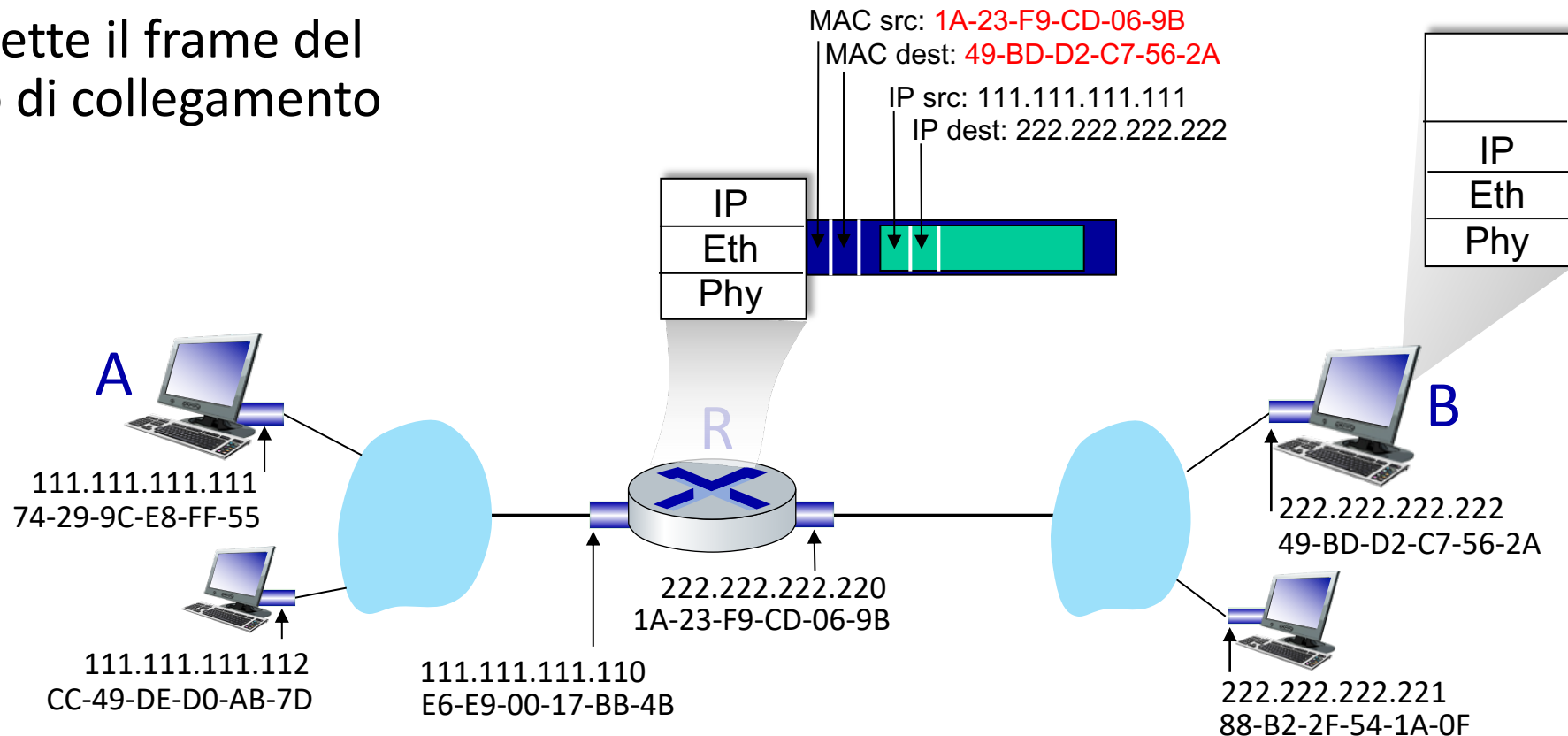
# Instradamento verso un'altra sottorete: indirizzamento

- R determina l'interfaccia in uscita, passa il datagramma con IP di src A e dest B al livello di collegamento
- R crea un frame contenente il datagramma IP da A a B.  
Indirizzo di destinazione del frame: l'indirizzo MAC di B



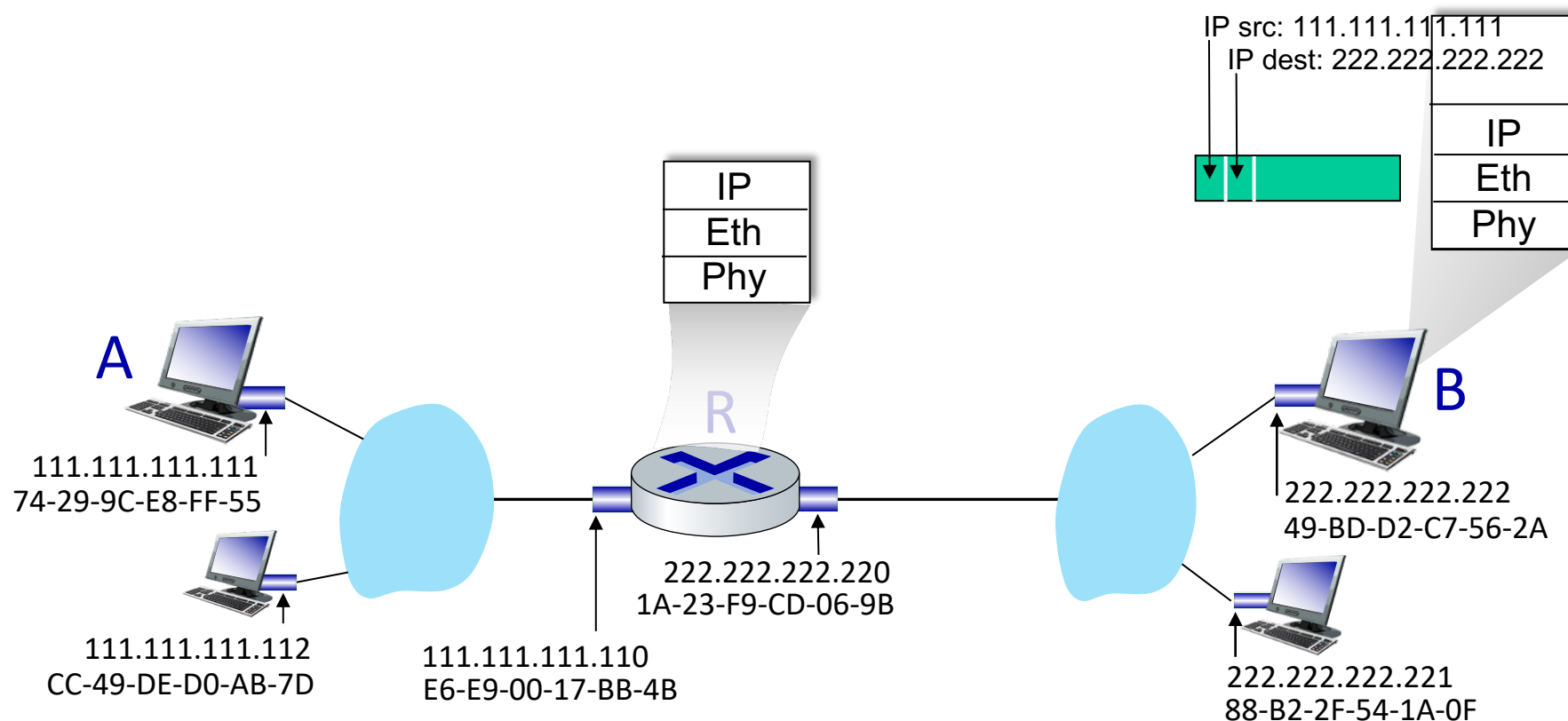
# Instradamento verso un'altra sottorete: indirizzamento

- R determina l'interfaccia in uscita, passa il datagramma con IP di src A e dest B al livello di collegamento
- R crea un frame contenente il datagramma IP da A a B.  
Indirizzo di destinazione del frame: l'indirizzo MAC di B
- trasmette il frame del livello di collegamento



# Instradamento verso un'altra sottorete: indirizzamento

- B riceve il frame, estrae il datagramma IP
- B passa il datagramma al livello superiore dello stack (rete)

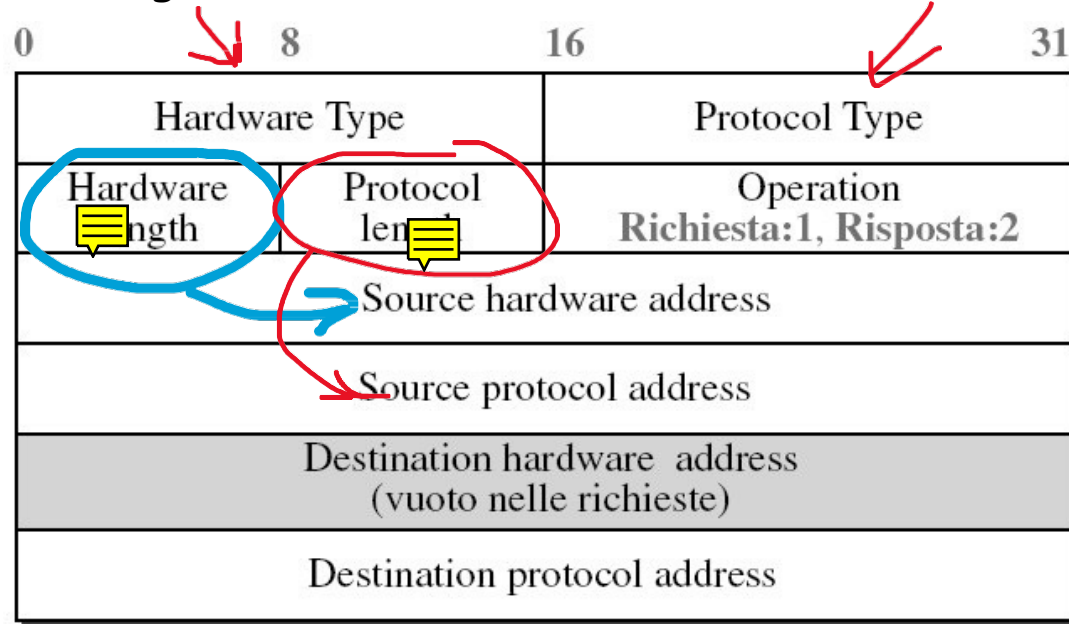




# Formato del pacchetto ARP

Protocollo del livello  
di collegamento (ES. Ethernet)

Protocollo del livello di rete (ES. IPv4)



**Hardware:** protocollo di collegamento della LAN o WAN

**Protocol:** protocollo del livello di rete

I pacchetti ARP vengono incapsulati direttamente all'interno di frame di livello di collegamento

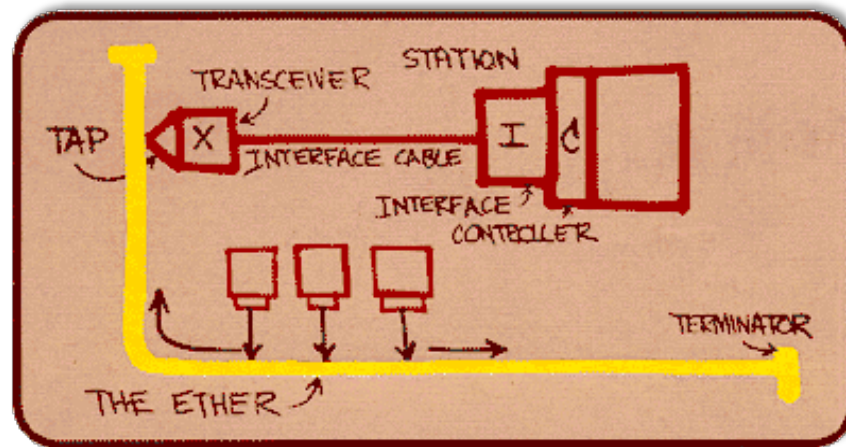
# Livello di collegamento e LAN: sommario

- introduzione
- rilevamento e correzione degli errori
- protocolli di accesso multiplo
- LAN
  - indirizzamento, ARP
  - Ethernet
  - switch
  - VLAN
- virtualizzazione dei collegamenti: MPLS
- data center
- un giorno nella vita di una richiesta web

# Ethernet

Tecnologia LAN cablata dominante:

- prima tecnologia LAN ampiamente utilizzata
- più semplice, economico delle altre
- ha tenuto il passo con velocità: 10 Mbps – 400 Gbps
- chip singolo, velocità multiple (ad esempio, Broadcom BCM5761)



*Schizzo Ethernet di Metcalfe*

# Ethernet: topologia fisica

- **bus:** popolare fino alla metà degli anni '90

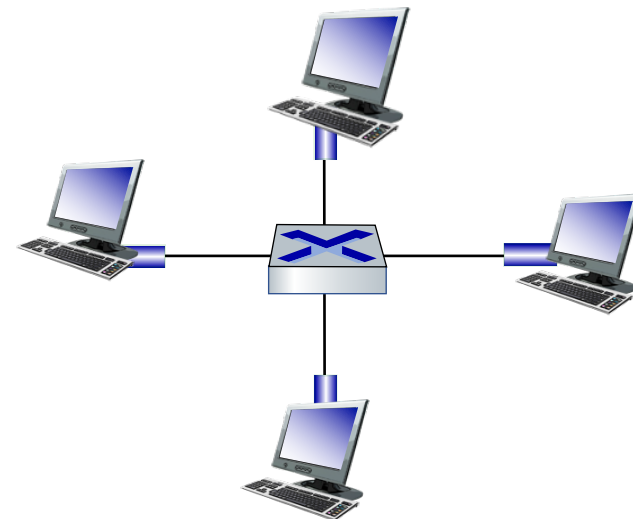
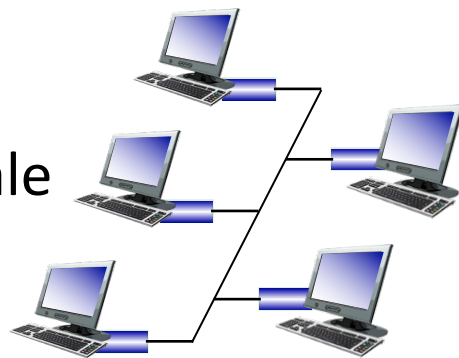


- tutti i nodi nello stesso dominio di collisione (possono entrare in collisione tra loro)
- si può anche usare un **hub** con più porte che fornisce la stessa topologia (bus comune)
- half-duplex

- **commutato (switched):** prevalente oggi

- *switch* di livello di collegamento attivo al centro della rete
- ogni ramo esegue un protocollo Ethernet (separato) (i nodi non entrano in collisione tra loro e sono tutti *potenzialmente* full-duplex)

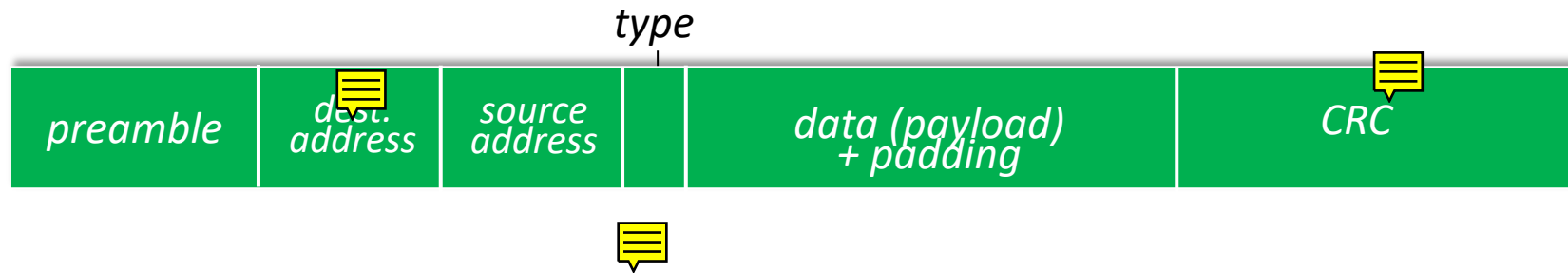
**bus:** cavo coassiale  
o hub



**switched**

# Struttura del frame Ethernet

l'interfaccia di invio incapsula il datagramma IP (o un altro pacchetto di protocollo del livello di rete) nel **frame Ethernet**



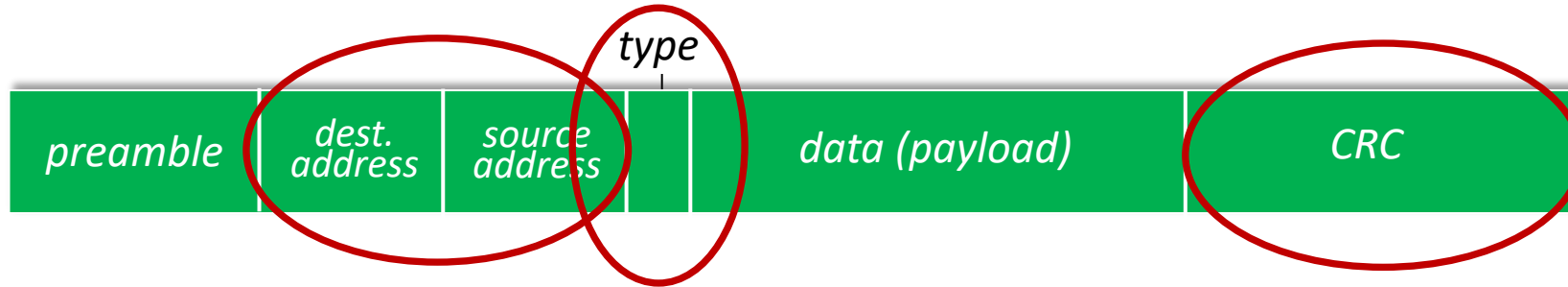
## *preamble:*

- utilizzato per sincronizzare le frequenze di clock di mittente e destinatario
- 7 bytes di 10101010 seguiti da un byte = 10101011

## *data:*

- lunghezza minima **46 byte**: padding aggiunto se necessario
- lunghezza massima MTU (tipicamente 1500 byte)

# Struttura del frame Ethernet



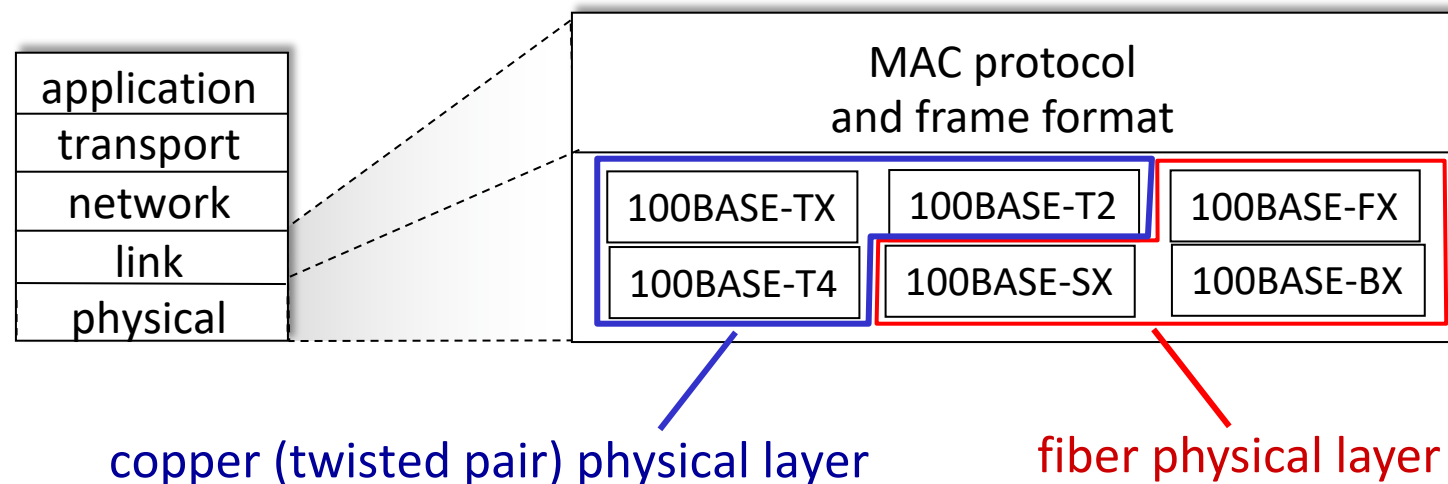
- **addresses:** 6 byte, indirizzi MAC di destinazione e mittente
  - se l'adattatore riceve frame con indirizzo di destinazione corrispondente o con indirizzo broadcast (ad es. pacchetto ARP), passa i dati nel frame al protocollo del livello di rete
  - in caso contrario, l'adattatore elimina il frame
- **type:** indica il protocollo di livello superiore
  - principalmente IP ma altri possibili, ad esempio Novell IPX, AppleTalk
  - utilizzato per demultiplare fino al ricevitore
- **CRC:** controllo di ridondanza ciclico al ricevitore
  - errore rilevato: il frame viene eliminato

# Ethernet: unreliable, connectionless

- **connectionless**: nessun handshaking tra le NIC di l'invio e ricezione
- **unreliable**: la NIC ricevente non risponde con ACK o NAK al NIC di invio
  - i dati nei frame eliminati vengono quindi recuperati solo se il mittente utilizza un protocollo di trasporto affidabile (ad esempio, TCP), altrimenti i dati eliminati verranno persi
- Protocollo MAC di Ethernet: CSMA/CD senza slot con **backoff**

# Standard Ethernet 802.3: livello fisico e di collegamento

- *esistono **molti** diversi standard Ethernet*
  - hanno in comune il protocollo MAC e il formato del frame
  - diverse velocità: 2 Mbps, 10 Mbps, 100 Mbps, 1 Gbps, 10 Gbps, 40 Gbps
  - diversi supporti di livello fisico: fibra, cavo





# Livello di collegamento e LAN: sommario

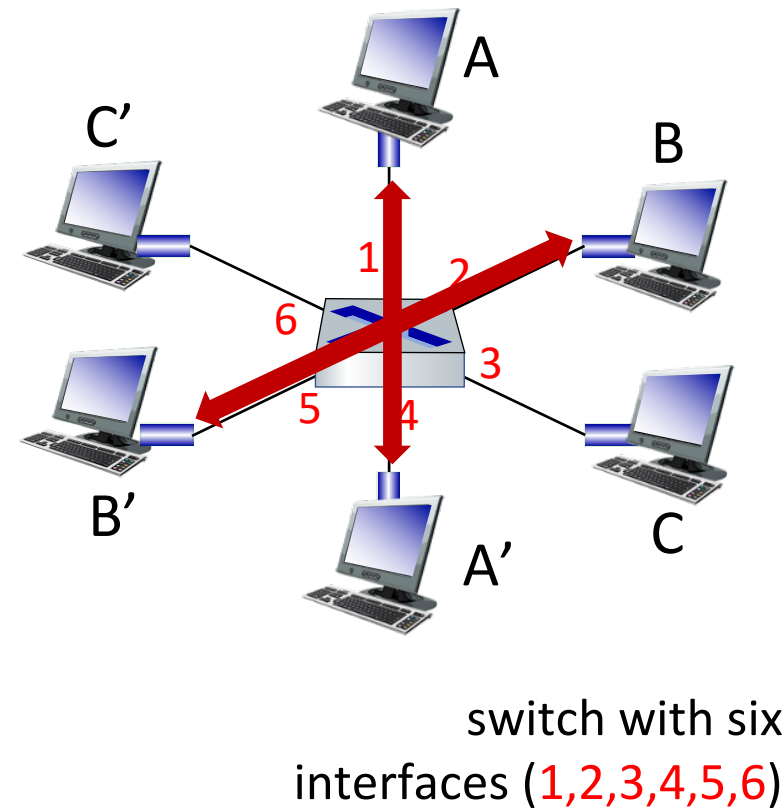
- introduzione
- rilevamento e correzione degli errori
- protocolli di accesso multiplo
- LAN
  - indirizzamento, ARP
  - Ethernet
  - switch
  - VLAN
- virtualizzazione dei collegamenti: MPLS
- data center
- un giorno nella vita di una richiesta web

# Switch Ethernet

- Lo switch è un dispositivo **a livello di collegamento**: assume un ruolo *attivo*:
  - garantisce che il segnale rimanga allo stesso livello (amplificatore)
  - memorizza e inoltra frame Ethernet
  - esamina l'indirizzo MAC del frame in entrata, inoltra *selettivamente* il frame a uno o più collegamenti in uscita quando il frame deve essere inoltrato sul segmento (di LAN), utilizza CSMA/CD per accedere al segmento (sottoinsieme di nodi connessi a una porta)
- **trasparente**: host *ignari* della presenza di switch
- **plug-and-play, self-learning**
  - non è necessario configurare gli switch
- **Motivazione iniziale**: aumentare la velocità di Ethernet richiedeva la diminuzione della lunghezza ( $T_r \geq 2T_p$ ) o l'abbandono del **paradigma di bus condiviso**

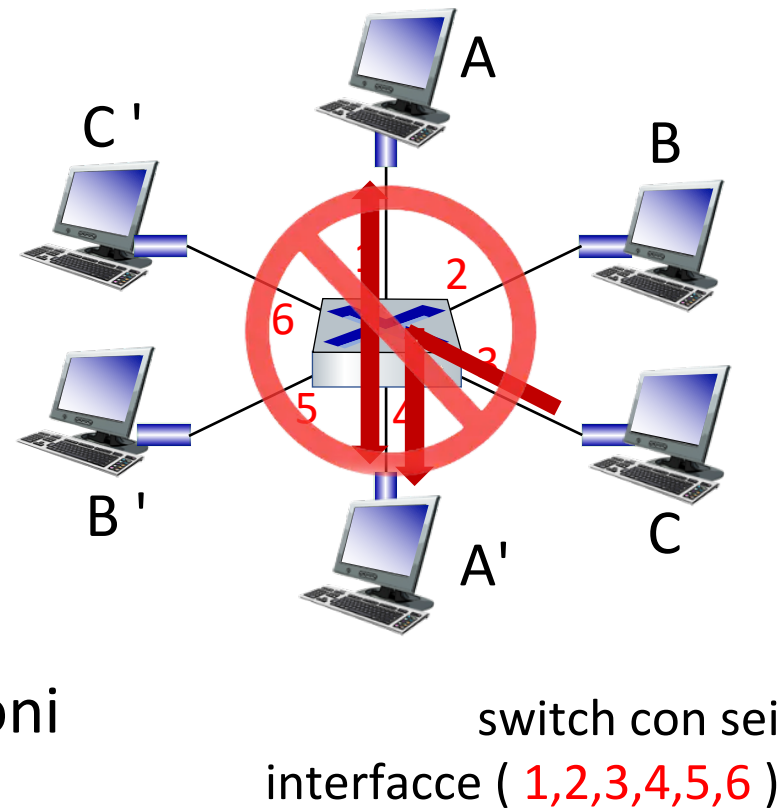
# Switch: più trasmissioni simultanee

- gli host dispongono di una connessione diretta dedicata allo switch
- commuta i pacchetti buffer
- Protocollo Ethernet utilizzato su *ogni* collegamento in entrata, quindi:
  - nessuna collisione, full-duplex (di default)
  - ogni collegamento è il proprio dominio di collisione
- **commutazione:** A-A' e B-B' possono trasmettere simultaneamente, senza collisioni



# Switch: più trasmissioni simultanee

- gli host dispongono di una connessione diretta dedicata allo switch
- commuta i pacchetti buffer
- Protocollo Ethernet utilizzato su *ogni* collegamento in entrata, quindi:
  - nessuna collisione, full-duplex (di default)
  - ogni collegamento è il proprio dominio di collisione
- **commutazione:** A-A' e B-B' possono trasmettere simultaneamente, senza collisioni
  - ma A-A' e C-A' *non possono* avvenire simultaneamente! Potenziale perdita di pacchetti



# Cambia tabella di inoltra

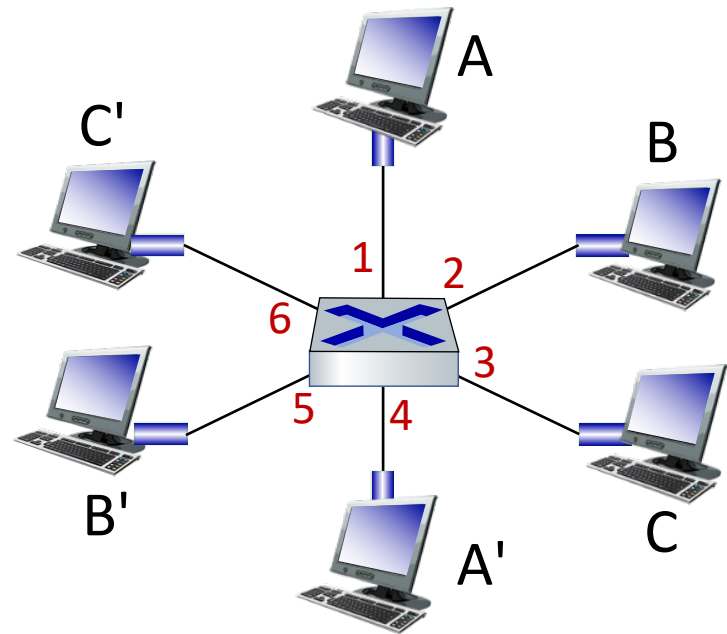
D: come fa lo switch a sapere che A' è raggiungibile tramite l'interfaccia 4, B' è raggiungibile tramite l'interfaccia 5?

R: ogni switch ha una **switch table**:

- (indirizzo MAC dell'host, interfaccia per raggiungere l'host, timestamp)
- sembra una tabella di routing!

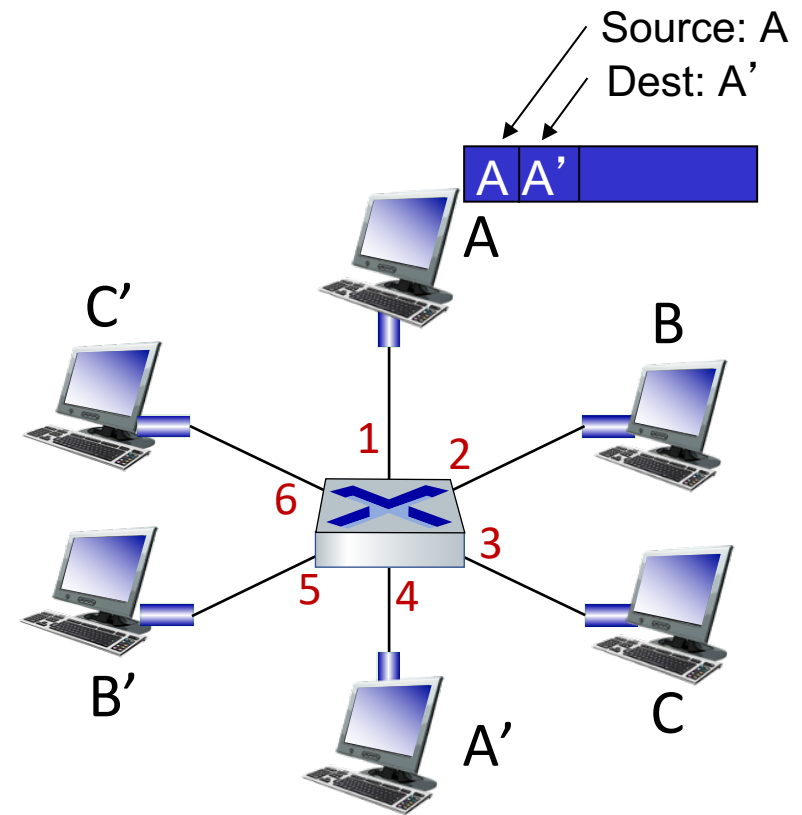
D: come vengono create e mantenute le voci nella switch table?

- qualcosa come un protocollo di routing?



# Switch: self-learning

- switch *impara* quali host possono essere raggiunti tramite quali interfacce
  - quando un frame viene ricevuto, lo switch "apprende" la posizione del mittente: segmento LAN in entrata
  - Inoltre conosce di quale host si tratta leggendo il source MAC address nel pacchetto
  - registra la coppia MAC/location nella switch table



MAC addr	interface	TTL
A	1	60

*Switch table  
(initially empty)*

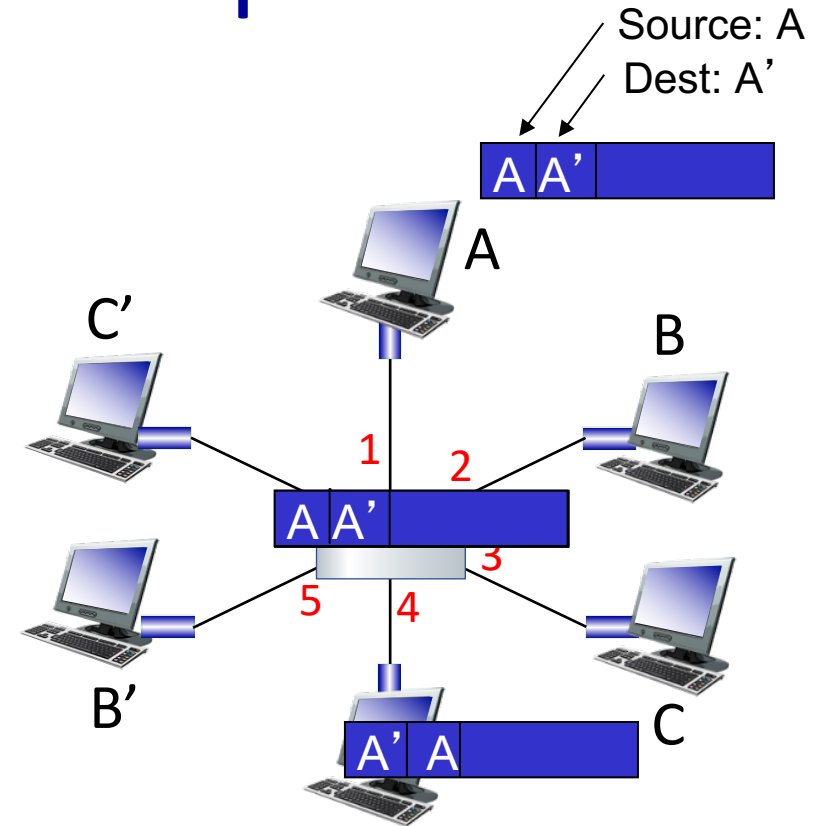
# Switch: frame filtering/forwarding

quando frame ricevuto allo switch:

1. registra il link in entrata e l'indirizzo MAC dell'host di invio
2. cerca nella switch table (exact match) usando l'indirizzo MAC di destinazione
3. **if** esiste una voce per la destinazione nella switch table  
    **then** {  
        **if** destinazione sul segmento da cui è arrivato il frame  
            **then** scarta il frame  
            **else** inoltra il frame sull'interfaccia indicata dalla voce  
    }  
    **else** flood /\* inoltra su tutte le interfacce tranne l'interfaccia in arrivo \*/

# Self-learning, forwarding: esempio

- destinazione del frame, A', location sconosciuta: **flood**
- destinazione A, location conosciuta: **invio selettivo su un solo link**



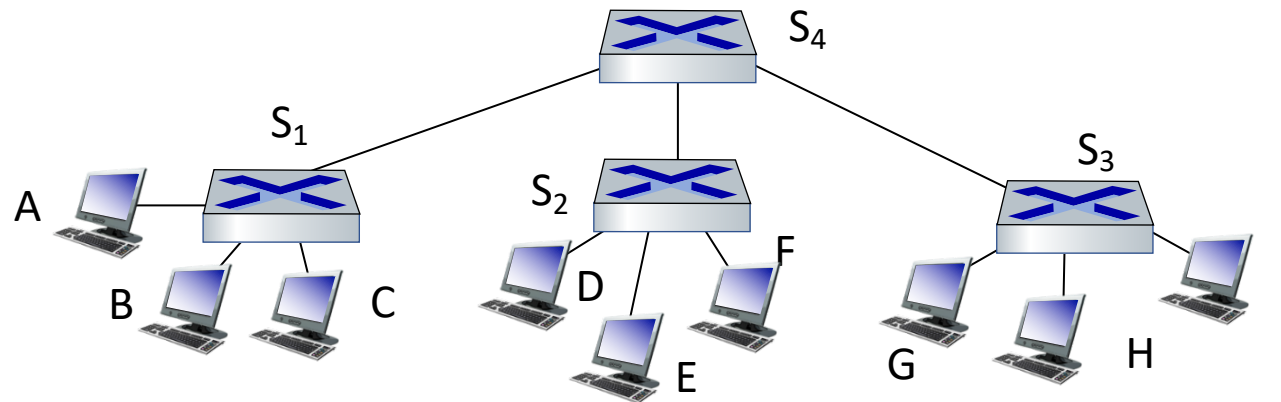
MAC addr	interface	TTL
A	1	60
A'	4	60

*switch table  
(inizialmente  
vuota)*



# Più switch interconnessi

gli switch possono essere connessi tra loro e mantengono la proprietà di self-learning

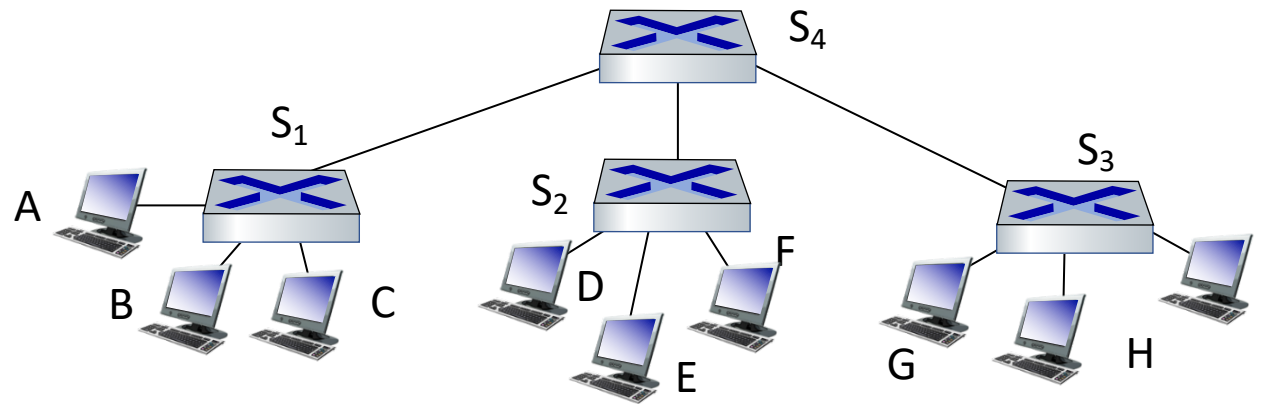


D: invio da A a G: come fa  $S_1$  a sapere di inoltrare via  $S_4$  e  $S_3$ ?

- R: self learning! (funziona esattamente come nel caso di un singolo switch!)

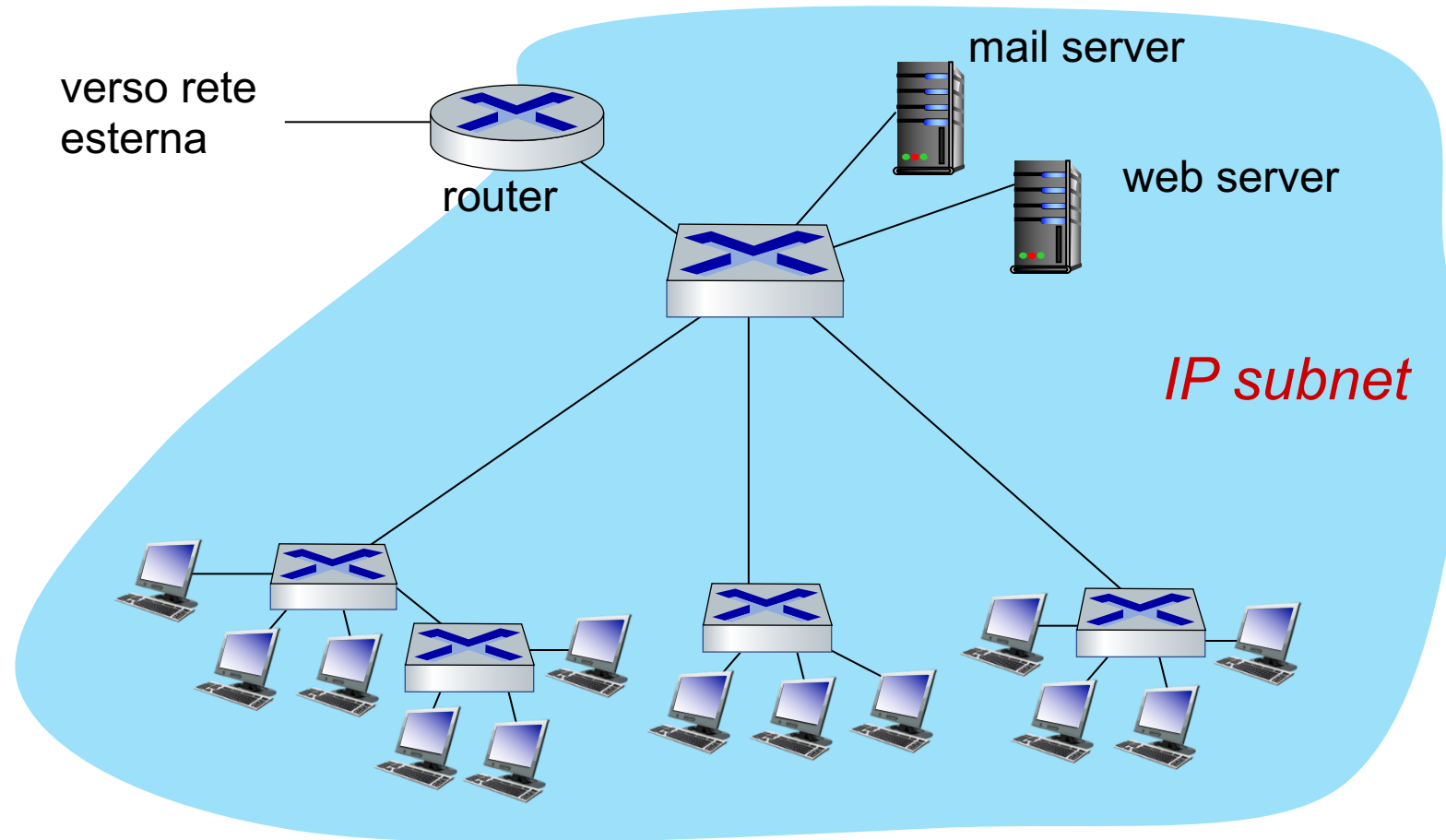
# Esempio di Self-learning multi-switch

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



D: descrivere l'inoltro dei pacchetti lungo  $S_1$ ,  $S_2$ ,  $S_3$ ,  $S_4$  e la corrispondente switch table

# Piccola rete istituzionale



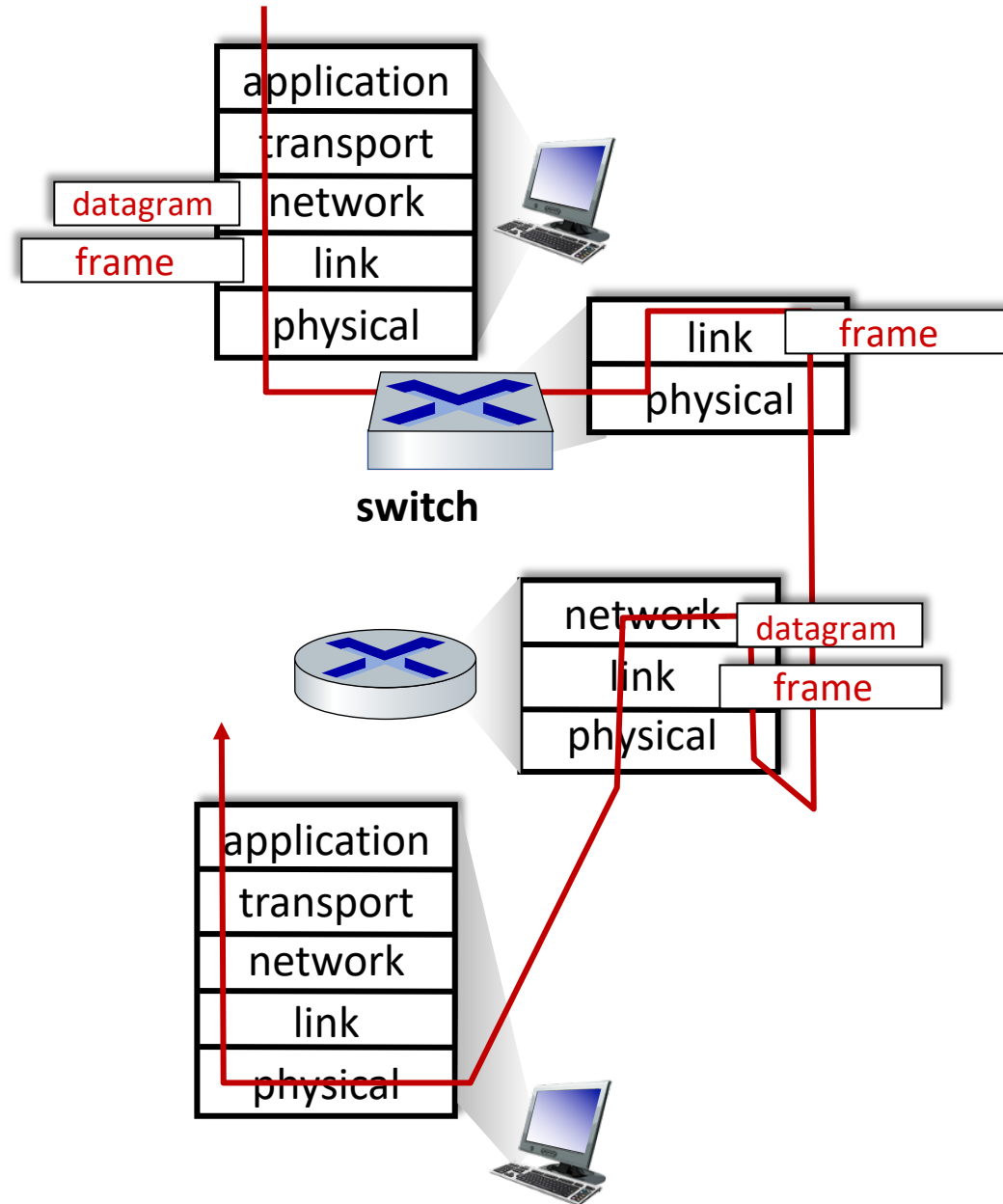
# Switches vs. routers

**entrambi sono store-and-forward:**

- **routers:** dispositivi a livello di rete (esamina l'header a livello di rete, crea un nuovo frame a ogni forwarding)
- **switch:** dispositivi a livello di collegamento (esamina le intestazioni a livello di collegamento, non modifica i frame)

**entrambi hanno tabelle di inoltramento:**

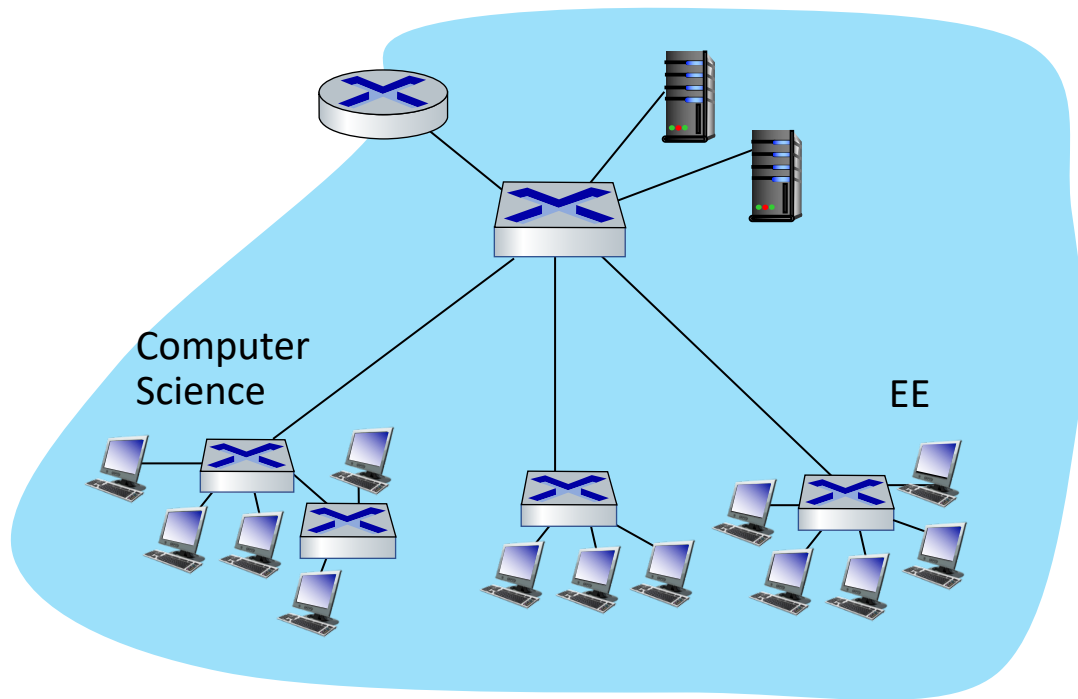
- **routers:** calcola forwarding utilizzando algoritmi di routing, indirizzi IP
- **switch:** apprende la switch table utilizzando flooding, apprendimento, indirizzi MAC



# Livello di collegamento e LAN: sommario

- introduzione
- rilevamento e correzione degli errori
- protocolli di accesso multiplo
- LAN
  - indirizzamento, ARP
  - Ethernet
  - switch
  - VLAN
- virtualizzazione dei collegamenti: MPLS
- data center
- un giorno nella vita di una richiesta web

# Virtual LANs (VLANs): motivazione

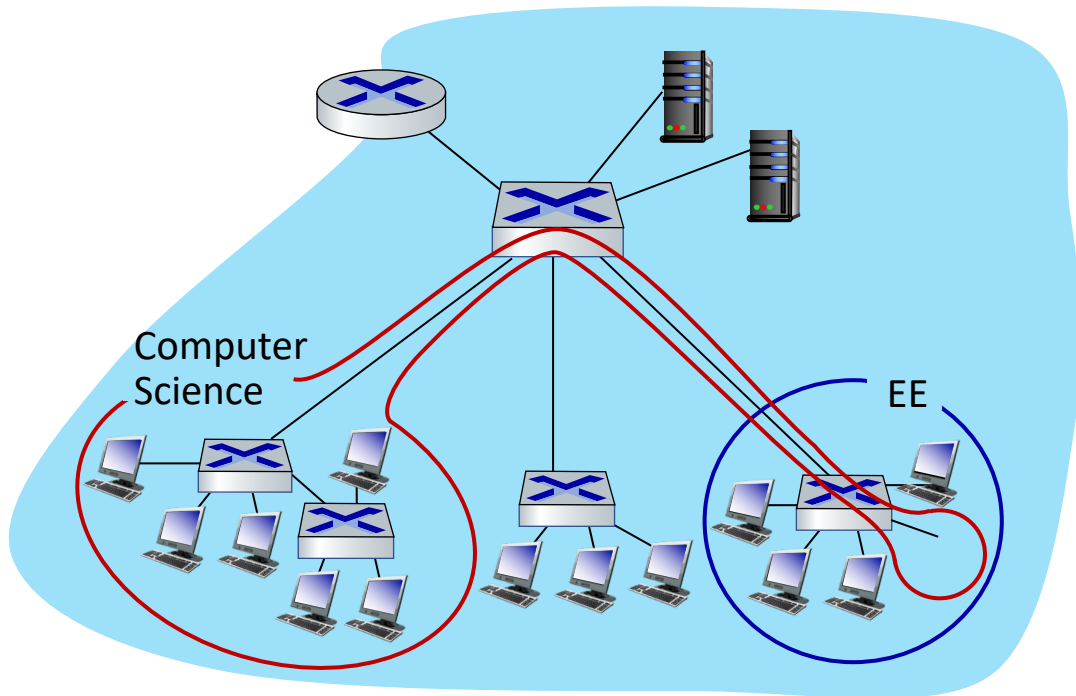


singolo dominio di broadcast:

- *scaling*: tutto il traffico broadcast di livello 2 (ARP, DHCP, MAC sconosciuto) deve attraversare l'intera LAN
- problemi di efficienza, sicurezza, privacy

# Virtual LANs (VLANs): motivazione

*D:* cosa succede quando le dimensioni della LAN crescono, e gli utenti cambiano il punto di collegamento?



singolo dominio di broadcast:

- *scaling*: tutto il traffico broadcast di livello 2 (ARP, DHCP, MAC sconosciuto) deve attraversare l'intera LAN
- problemi di efficienza, sicurezza, privacy

problemi amministrativi:

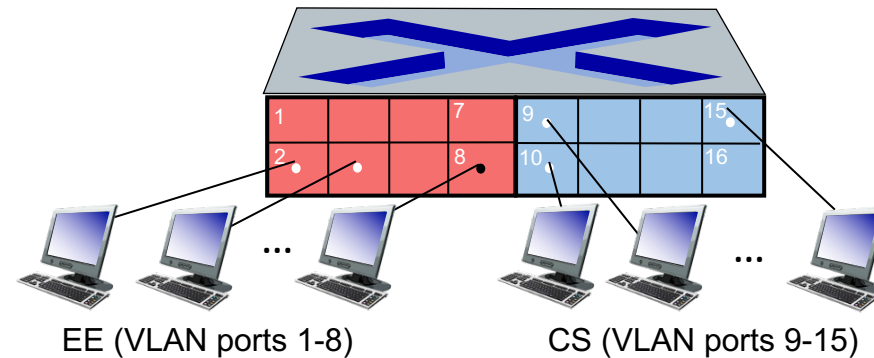
- L'utente CS sposta l'ufficio in EE: collegato *fisicamente* allo switch EE, ma desidera rimanere *logicamente* collegato allo switch CS

# Port-based VLANs

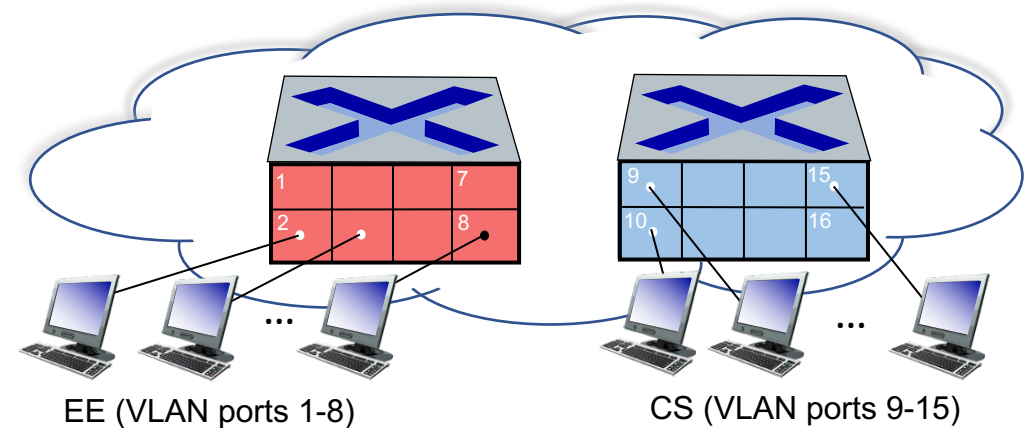
## Rete locale virtuale (VLAN)

gli switch che supportano le funzionalità VLAN possono essere configurati per definire più LAN *virtuali* su una singola infrastruttura LAN fisica

**port-based VLAN:** le porte dello switch vengono raggruppate (tramite software di gestione dello switch) in modo che il *singolo* switch ...



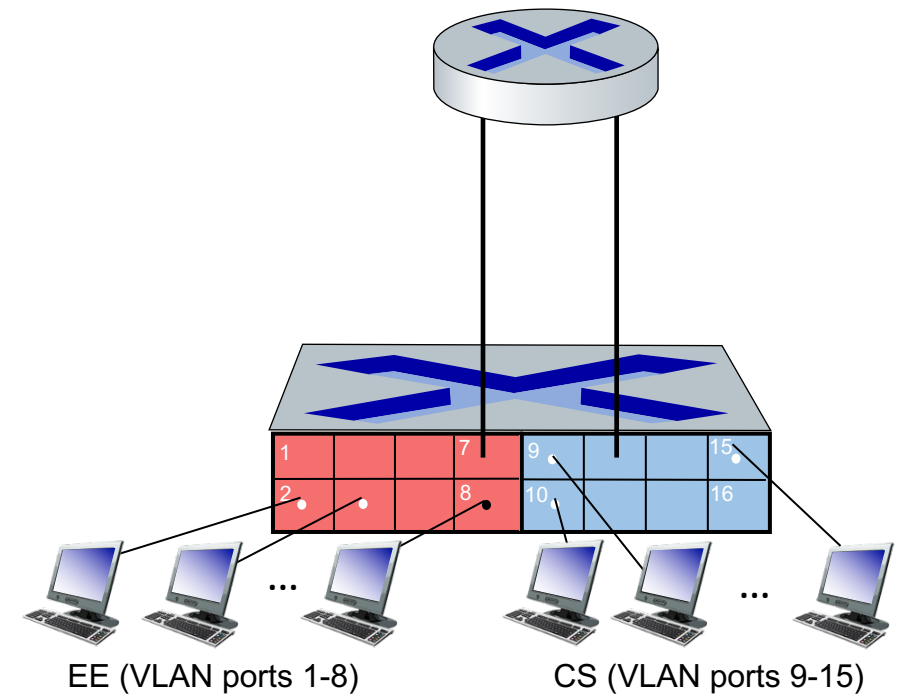
... operi come **più** switch virtuali



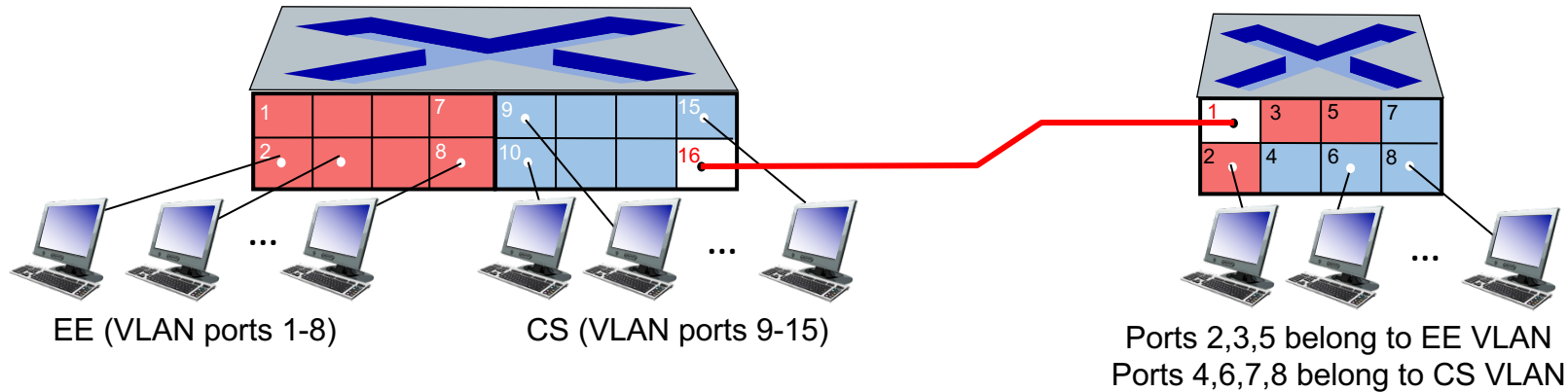


# Port-based VLANs

- **isolamento del traffico:** i frame da/verso le porte 1-8 possono raggiungere *solo le* porte 1-8
  - può anche definire la VLAN in base agli indirizzi MAC degli endpoint, piuttosto che alla porta dello switch
- **membership dinamica:** le porte possono essere assegnate dinamicamente tra le VLAN
- **forwarding tra VLANs:** via routing (come se fossero switch separati)
  - in pratica i device disponibili sono switch combinati con router



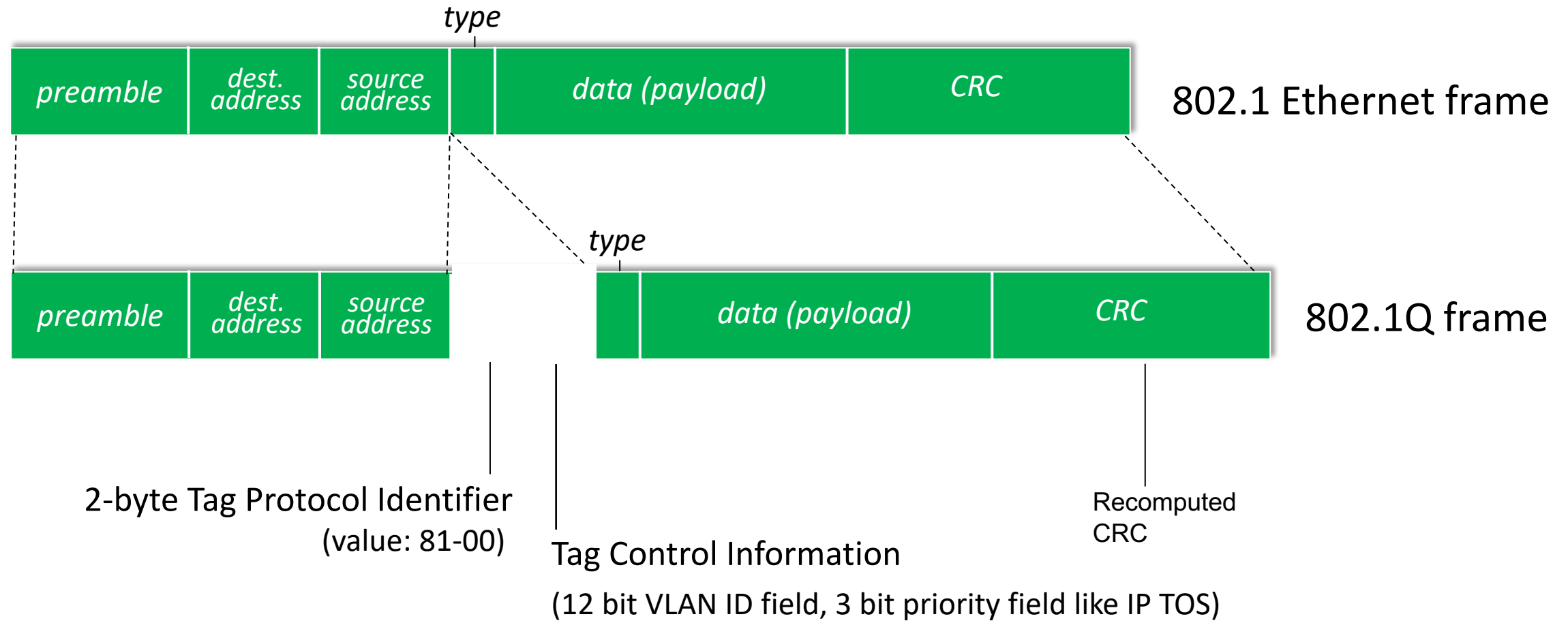
# VLAN che coprono più switch



**porta trunk:** trasporta frame tra VLAN definite su più switch fisici

- i frame inoltrati all'interno della VLAN tra gli switch non possono essere frame 802.1 classico: devono contenere informazioni sull'ID VLAN
- Il protocollo 802.1q aggiunge/rimuove ulteriori campi di intestazione per i frame inoltrati tra le porte trunk per identificare le VLAN

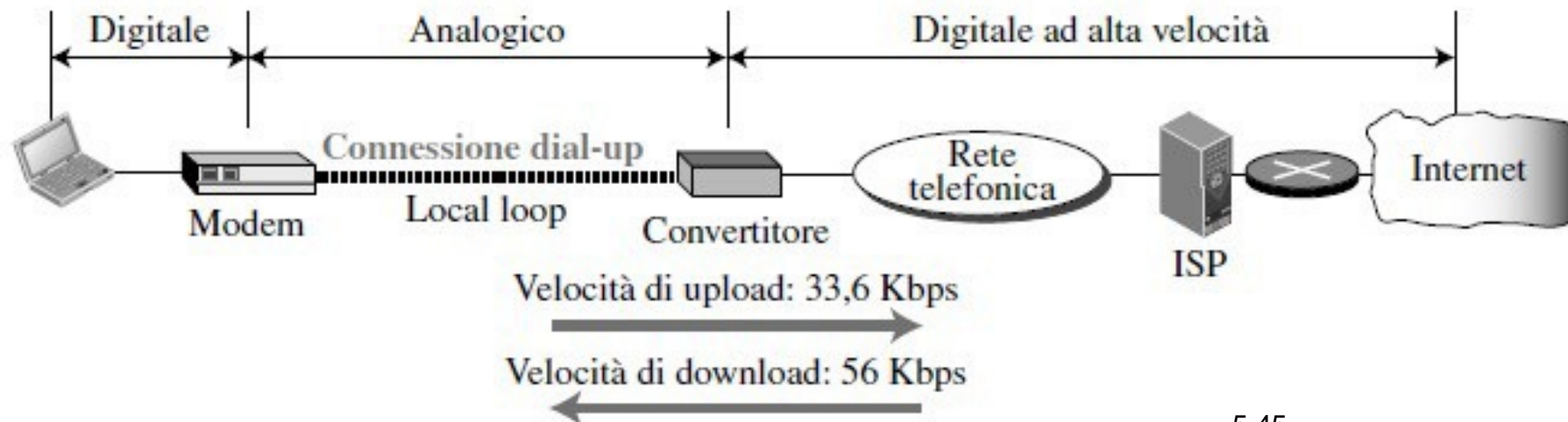
# 802.1Q VLAN frame format



Reti punto-punto

# Reti punto-punto

- Alcune reti punto-punto (telefoniche dial-up e ADSL) sono usate per fornire agli utenti accesso a Internet
- Collegamento dedicato tra due dispositivi
- Non utilizzano il controllo di accesso al mezzo condiviso (MAC) ma protocolli dedicati come il Point-to-point protocol



# Point-to-point protocol

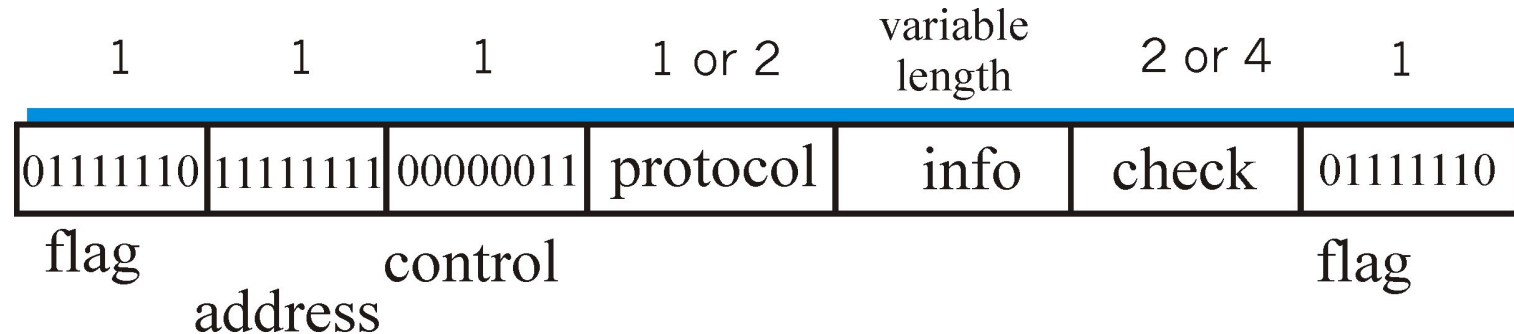
- Il protocollo Punto-punto è stato sviluppato dall' Internet Engineering Task Force (IETF) come mezzo per trasmettere dati per più di una rete sullo stesso collegamento seriale in un modo standard e indipendente dal produttore.
- Può trasportare traffico IP, Novell IPX, AppleTalk, DECnet
- Un mittente, un destinatario, un collegamento: estremamente semplice.
  - non c'è protocollo di accesso al mezzo (MAC)
  - non occorre indirizzamento MAC esplicito
  - il collegamento potrebbe essere una linea telefonica seriale commutata, un collegamento a fibra ottica

# Requisiti di IETF per il progetto PPP [RFC 1547]

- **Framing dei pacchetti:** il protocollo PPP del mittente incapsula un pacchetto a livello di rete all'interno di un pacchetto PPP a livello di link.
- **Rilevazione degli errori** (ma non la correzione)
- **Disponibilità della connessione:** il protocollo deve rilevare la presenza di eventuali guasti a livello di link e segnalare l'errore al livello di rete.

# Formato dei pacchetti dati PPP

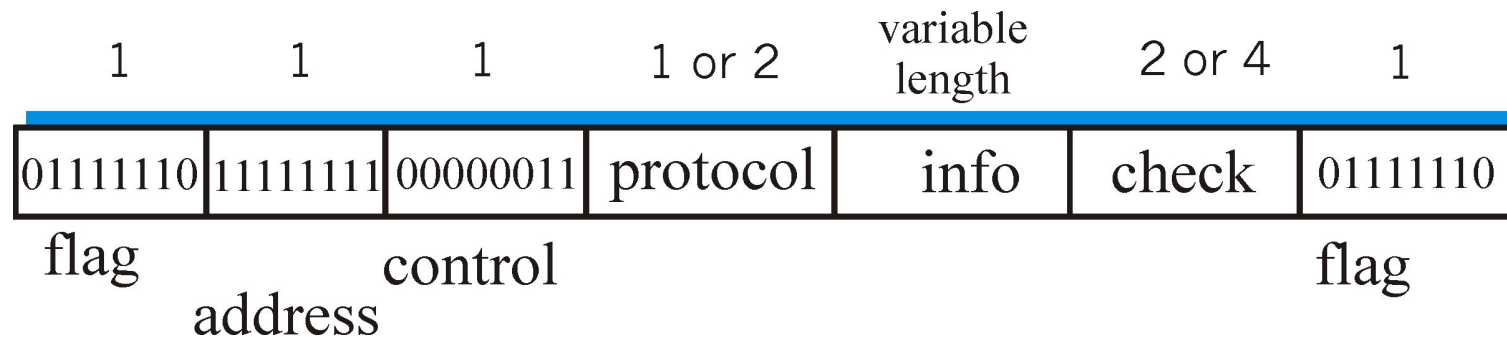
- **Flag:** ogni pacchetto inizia e termina con un byte con valore 01111110
- **Indirizzo:** unico valore (11111111)
- **Controllo:** unico valore; ulteriori valori potrebbero essere stabiliti in futuro
- **Protocollo:** indica al PPP del ricevente qual è il protocollo del livello superiore cui appartengono i dati incapsulati





# Formato dei pacchetti dati PPP

- **informazioni:** incapsula il pacchetto trasmesso da un protocollo del livello superiore (come IP) sul collegamento PPP.
- **checksum:** utilizzato per rilevare gli errori nei bit contenuti in un pacchetto; utilizza un codice a ridondanza ciclica HDLC a due o a quattro byte.



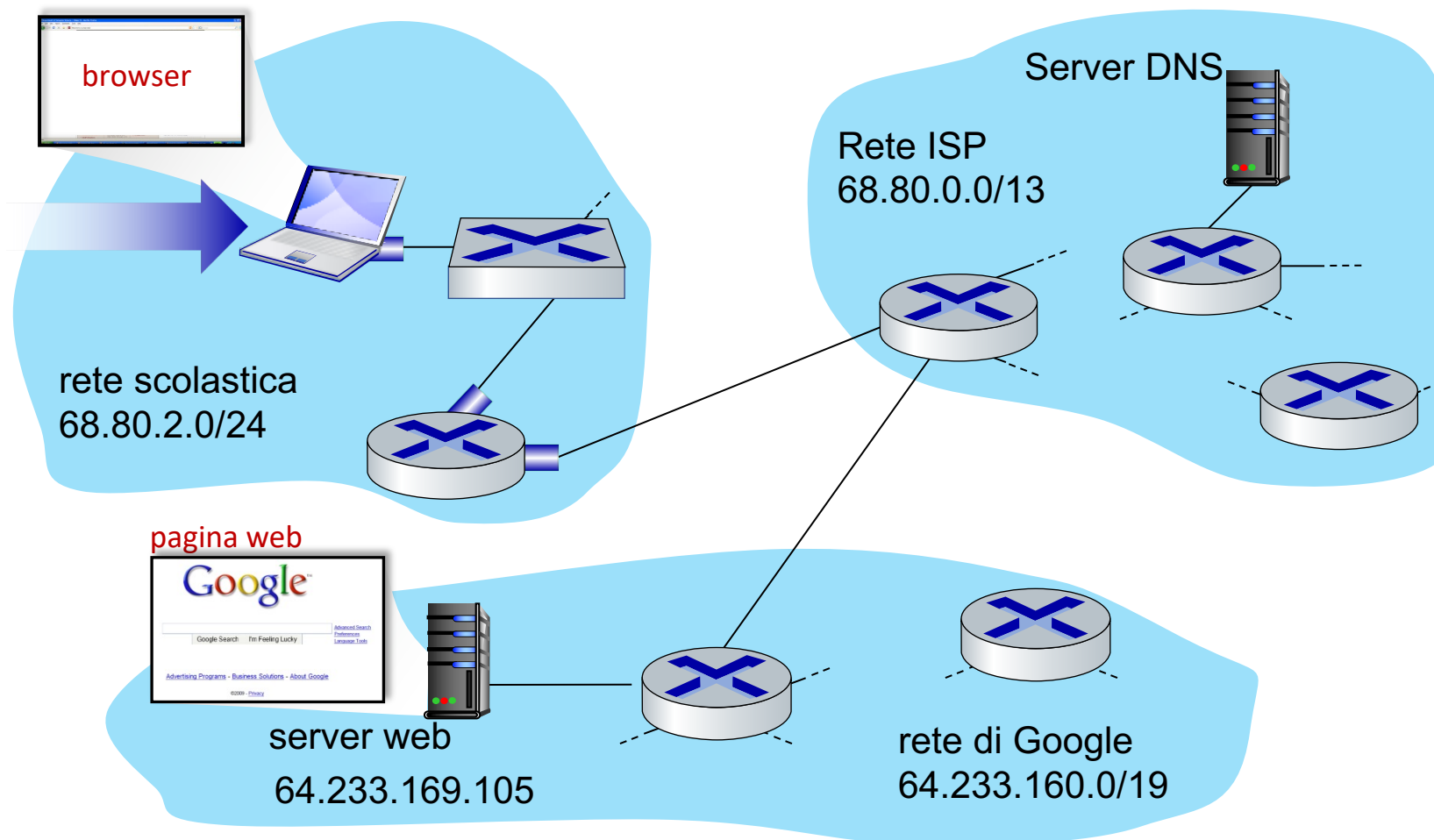
# Facciamo il punto

- Abbiamo completato il nostro viaggio attraverso la pila dei protocolli (ad eccezione del livello fisico)
- Abbiamo una solida conoscenza dei principi del networking, e anche degli aspetti pratici
- Provate a rivedervi la prima lezione dove abbiamo visto il viaggio di un pacchetto lungo tutti i protocolli
- Ci sono ancora argomenti da vedere!
  - Wireless
  - Sicurezza

# Esempio di navigazione web: scenario

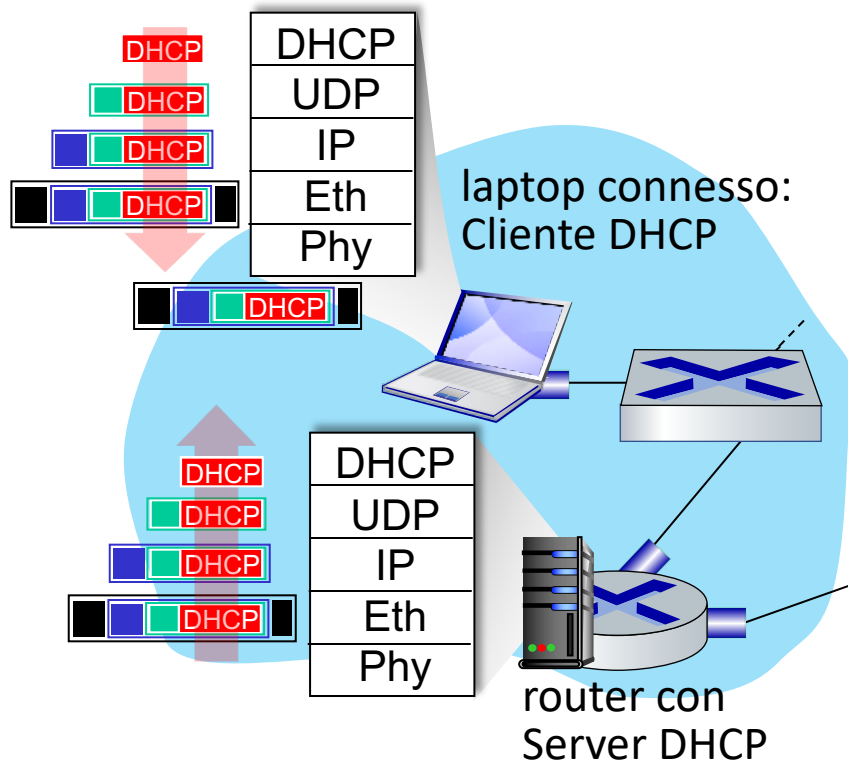
scenario:

- un client si collega alla rete scolastica
- richiede la pagina web: [www.google.com](http://www.google.com)



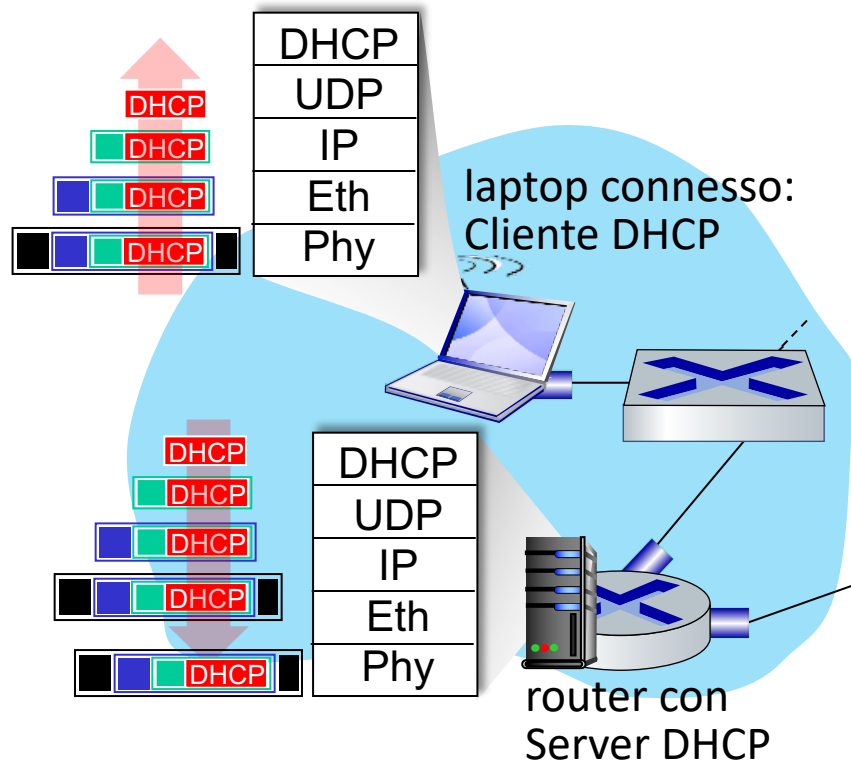
*Sembra  
semplice* !

# Esempio di navigazione web: connettersi a Internet



- il laptop che si connette deve ottenere il proprio indirizzo IP, l'indirizzo del router first-hop e del server DNS: usiamo **DHCP**
- Richiesta DHCP **incapsulata** in **UDP**, incapsulata in **IP**, incapsulata in **802.3 Ethernet**
- **Broadcast** frame Ethernet (dest: FFFFFFFFFFFFFFFF) su LAN, ricevuta dal router con server **DHCP**
- **Demux da** Ethernet > IP > UDP > DHCP

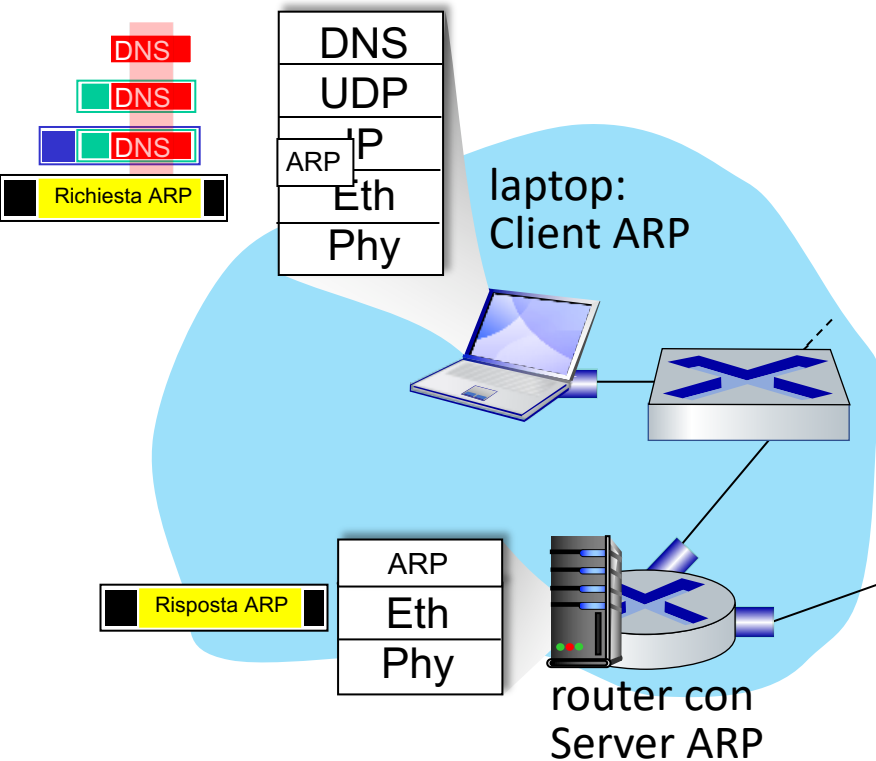
# Esempio di navigazione web: connettersi a Internet



- il server DHCP manda l'**ACK DHCP** contenente l'indirizzo IP del client , l'indirizzo IP del router first-hop, il nome e l'indirizzo IP del server DNS
- incapsulamento sul server DHCP, frame inoltrato (**switch table learning!**) tramite LAN, demultiplexing sul client
- Il client DHCP riceve la risposta DHCP ACK

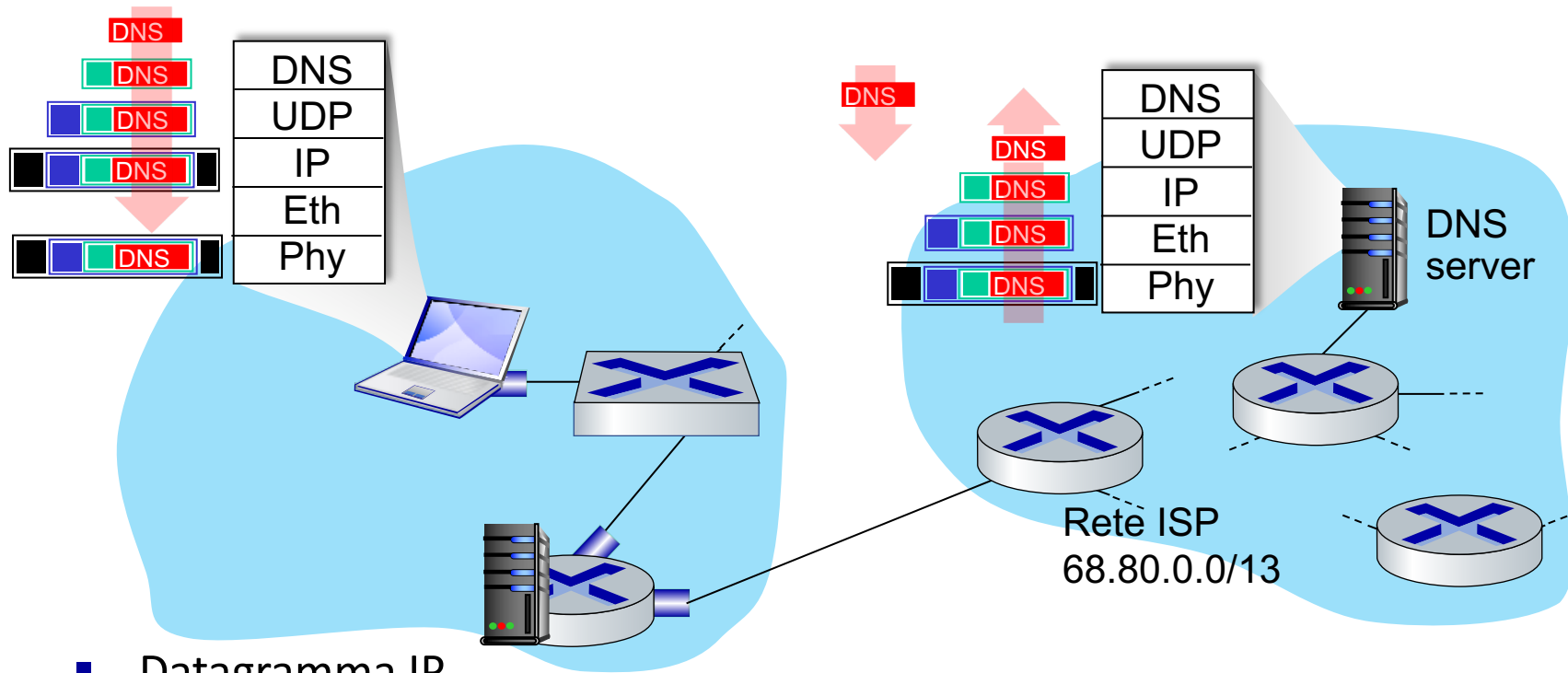
*Il client ora ha l'indirizzo IP, conosce il nome e l'indirizzo del DNS server, indirizzo IP del suo router first-hop*

# ARP (prima del DNS, prima dell'HTTP)



- prima di inviare la richiesta **HTTP**, è necessario l'indirizzo IP google.com: **DNS**
- Query DNS creata, incapsulata in UDP, incapsulata in IP, incapsulata in Eth. Per inviare frame al router, è necessario l'indirizzo MAC dell'interfaccia del router: **ARP**
- **query ARP**, ricevuta dal router, che risponde con una **risposta ARP** fornendo l'indirizzo MAC dell'interfaccia del router
- il client ora conosce l'indirizzo MAC del first-hop router, quindi ora può inviare un frame contenente una query DNS

# Esempio di navigazione web: DNS

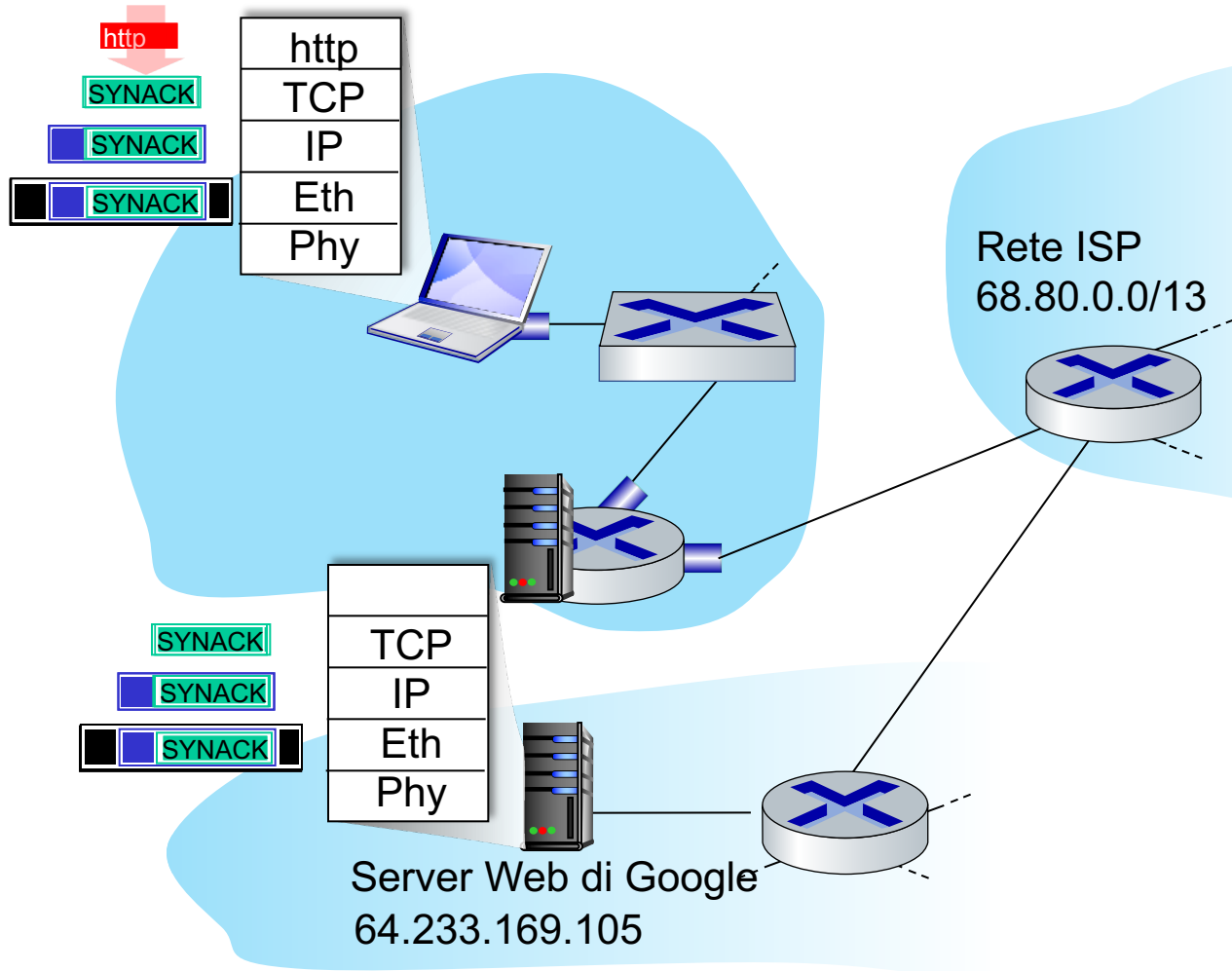


- Datagramma IP contenente la query DNS inoltrata tramite switch LAN dal client al router 1° hop

- Datagramma IP inoltrato dalla rete del campus alla rete ISP, instradato (tabelle create dai protocolli di routing **RIP**, **OSPF**, **IS-IS** e/o **BGP**) al server DNS

- demuxed al DNS
- Il DNS risponde al client con l'indirizzo IP di [www.google.com](http://www.google.com)

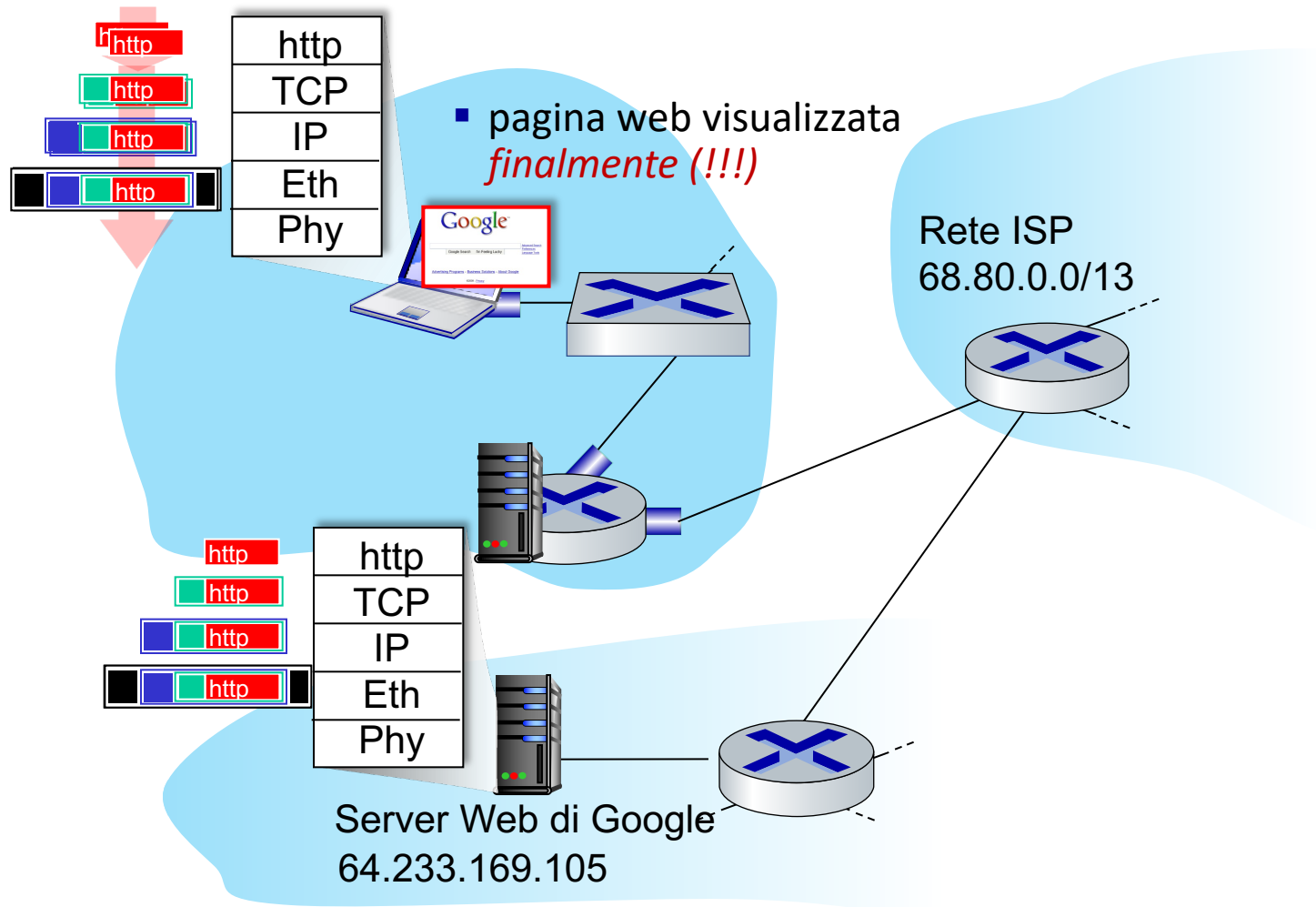
# Connessione TCP che trasporta HTTP



- per inviare la richiesta HTTP, il client prima deve aprire il **socket TCP** con il server web
- segmento TCP **SYN** (step 1 nell'handshake TCP a 3 vie) instradato al server web
- il server Web risponde con **TCP SYNACK** (step 2 nell'handshake TCP a 3 vie)
- connessione TCP **stabilita!**



# Richiesta/risposta HTTP




- **Richiesta HTTP** inviata al socket TCP
- Datagramma IP contenente la richiesta HTTP instradata a [www.google.com](http://www.google.com)
- il server web risponde con una **risposta HTTP** (contenente la pagina web)
- Datagramma IP contenente la risposta HTTP reinstradata al client

## Es. 1

- Due host in due reti diverse possono avere lo stesso indirizzo di livello di collegamento (MAC)?
- E lo stesso indirizzo di rete (IP)?

## Es. 2

1. Quattro stazioni sono collegate ad un hub in una rete Ethernet. Le distanze tra l'hub e le stazioni sono rispettivamente di 300m, 400m, 500m e 700m. Qual è la lunghezza di questa rete quando dobbiamo calcolare il tempo di propagazione? 
2. Come cambia il calcolo se sostituiamo l'hub con uno switch?

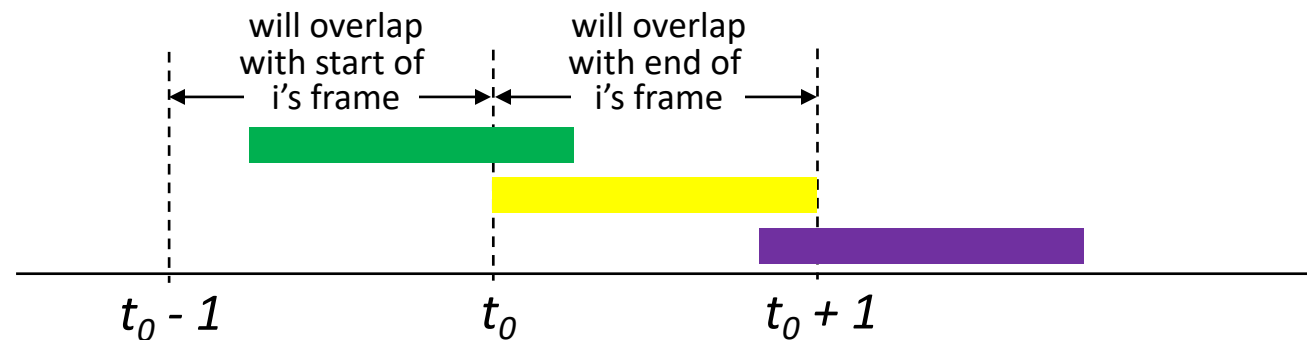
## Es. 3

- Le stazioni di una rete ALOHA puro inviano frame da 1000 bit alla velocità di 1Mbps. Qual è il tempo di vulnerabilità\* per tale rete?

\* tempo durante il quale un frame trasmesso da un'altra stazione può collidere con quello in oggetto

# Pure ALOHA

- Aloha senza slot: più semplice, nessuna sincronizzazione
  - quando arriva un frame viene trasmesso senza aspettare l'inizio di uno slot
- la probabilità di collisione aumenta senza sincronizzazione:
  - il frame inviato a  $t_0$  andrà in collisione con frame inviati nell'intervallo temporale  $[t_0-1, t_0+1]$  (due volte il tempo di trasmissione di un frame, questo intervallo è anche noto come **tempo di vulnerabilità**)



- Efficienza di pure Aloha: 18%!

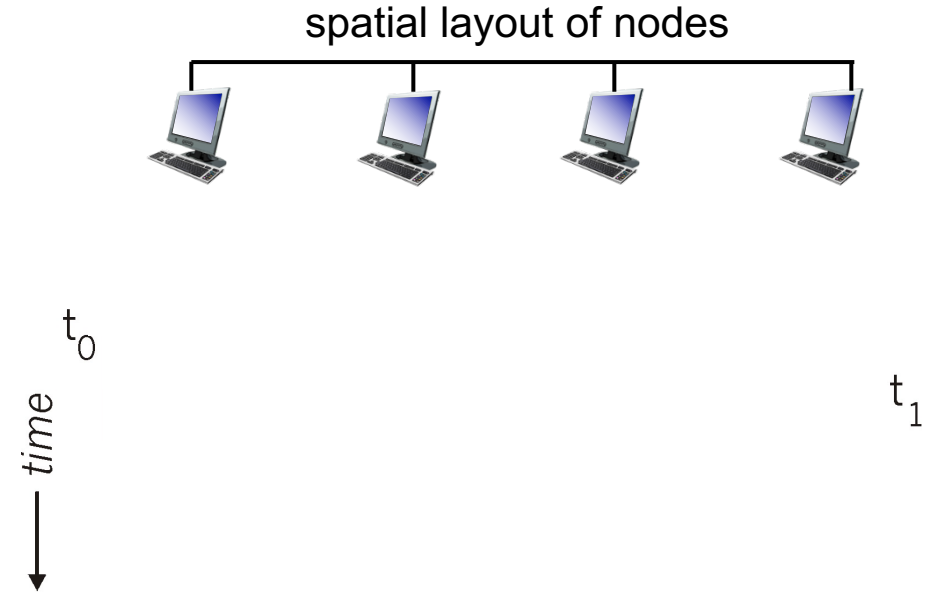
## Es. 4

- Le stazioni di una rete CSMA/CD inviano frame da 1000 bit alla velocità di 1Mbps su un cavo di 500m. Qual è il tempo di vulnerabilità\* per tale rete (velocità della luce  $2,5E8$ )

\* tempo durante il quale un frame trasmesso da un'altra stazione può collidere con quello in oggetto

# CSMA: collisioni

- collisioni *possono* ancora verificarsi con il carrier sense:
  - ritardo di propagazione significa che due nodi potrebbero ognuno non sentire la trasmissione appena iniziata dall'altro
- **collisione**: tempo di trasmissione dell'intero pacchetto sprecato
  - la distanza e il ritardo di propagazione svolgono un ruolo nel determinare la probabilità di collisione
- **Tempo di vulnerabilità**:  $T_p$



## Es. 5

- Assumendo che il ritardo di propagazione in una rete CSMA/CD broadcast sia  $5\mu s$  e che il tempo di trasmissione del frame sia  $10\mu s$ 
  1. Quanto impiega l'ultimo bit per raggiungere la destinazione dopo che è arrivato il primo?
  2. Per quanto tempo la rete è occupata da questo frame?
  3. Quanto è grande il tempo di vulnerabilità di questo frame?



## Es. 6

Si vuole progettare un campo CRC. Qual è l'effetto massimo di un rumore di 2 ms sui dati trasmessi alle seguenti velocità?

1. 1500 bps
2. 12 kbps

## Es. 7

- Ci sono solo tre stazioni attive in una rete Slotted Aloha: A,B,C.
  - Dato uno slot di tempo ogni stazione genera un frame rispettivamente con probabilità  $p_A=0.2$ ,  $p_B=0.3$ ,  $p_C=0.4$
1. Qual è l'efficienza (tasso di frame utili aka throughput) di ogni stazione?
  2. Qual è l'efficienza della rete?