Ripasso sulle reti

Obiettivi:

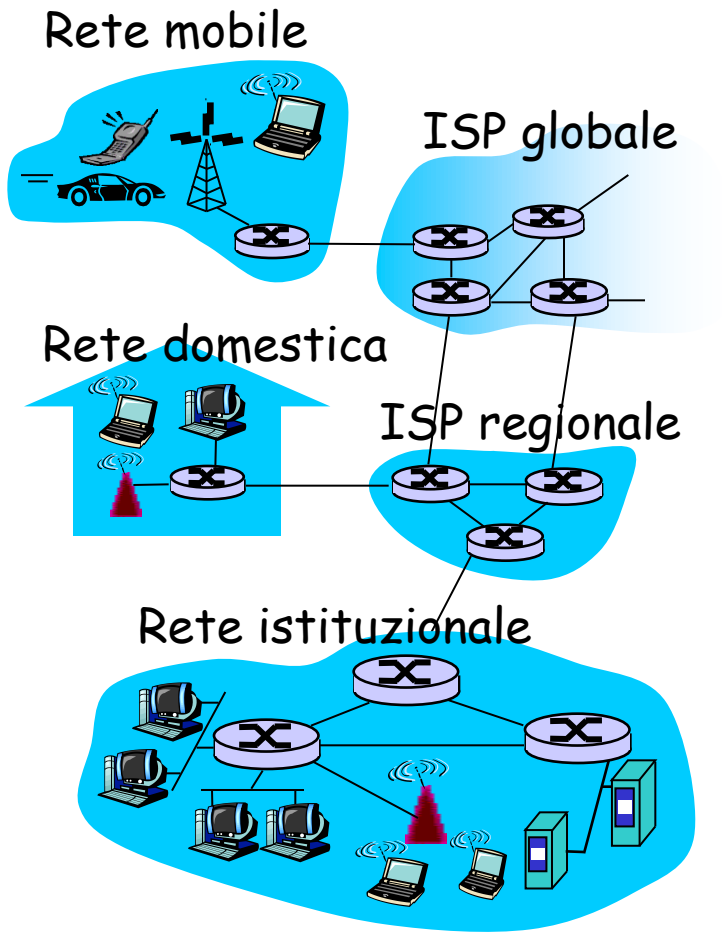
- Richiamare concetti chiave del corso introduttivo sulle reti di calcolatori
  - Rinfrescare la memoria su idee fondamentali
  - Creare una base di partenza comune
  - Identificare possibili lacune e lavoro di ripasso
  - Consolidare la terminologia

Sommario:

- Panoramica ad alto-livello
- Controllo di errore
- Controllo di flusso
- Controllo di congestione
- Indirizzamento
- Livello rete
- Livello link
- Controllo

Cos'è Internet: dal punto di vista fisico

- *Internet:* "rete di reti"
  - Approssimativamente gerarchica
  - Internet pubblica vs Intranet privata
- *protocolli* controllano invio e ricezione di messaggi
  - Es: TCP, IP, HTTP, Skype, Ethernet
- Internet standards
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force



Cos'è un protocollo?

Protocolli umani:

- "Mi scusi. Sa dirmi che ora è?"
- "Ho una domanda"
- Saluti tra le persone

… specifici messaggi da inviare

… specifiche azioni eseguite quando si ricevono messaggi, o altri eventi

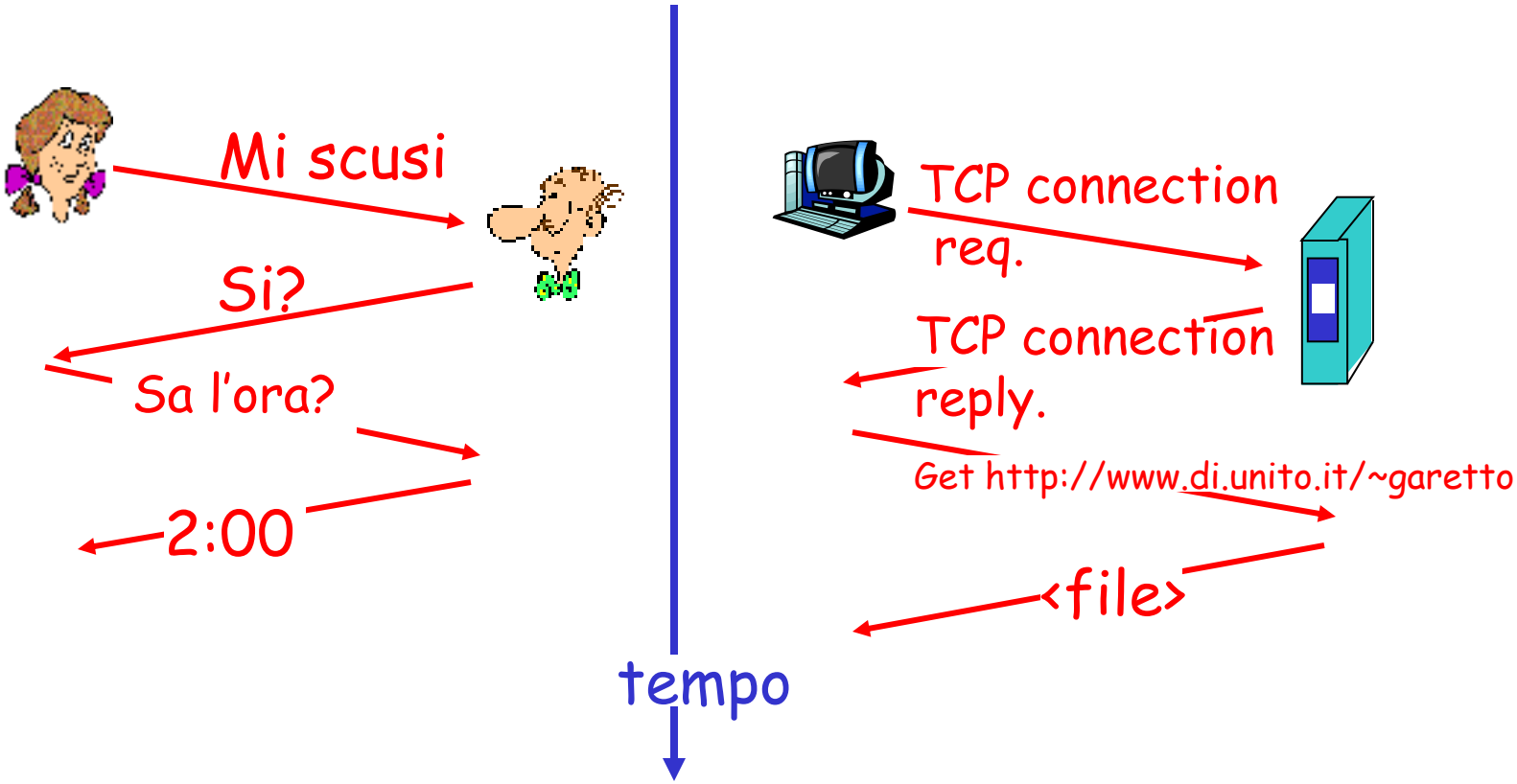
Protocolli di rete:

- macchine invece che umani
- Tutta l'attività di comunicazione in Internet è governata da protocolli

I protocolli definiscono il formato, l'ordine dei messaggi inviati e ricevuti tra le entità di rete, e le azioni intraprese quando si tx/rx i messaggi

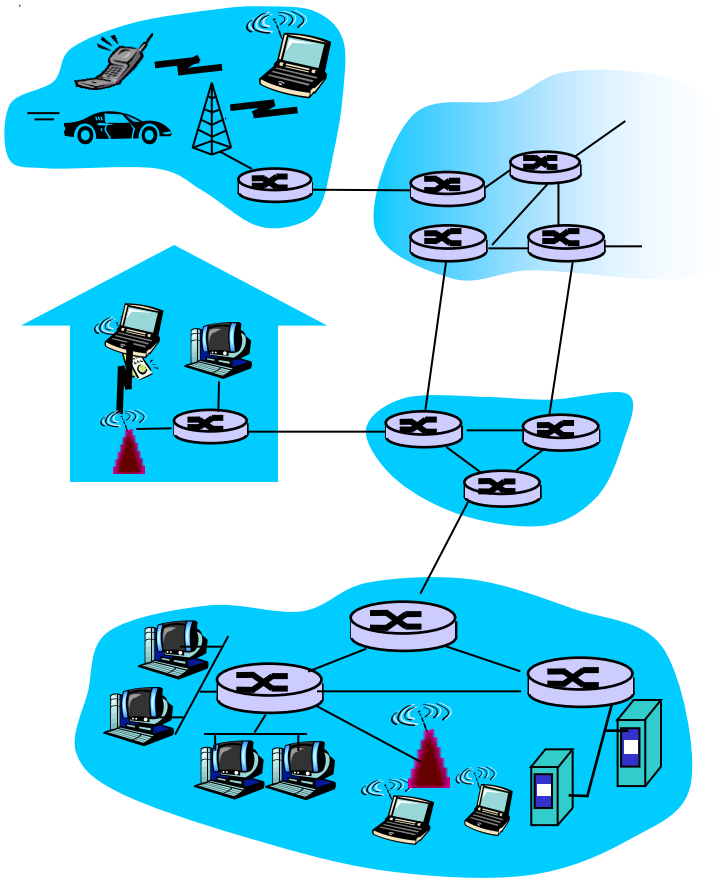
Cos'è un protocollo?

Un protocollo umano e un protocollo di Internet:



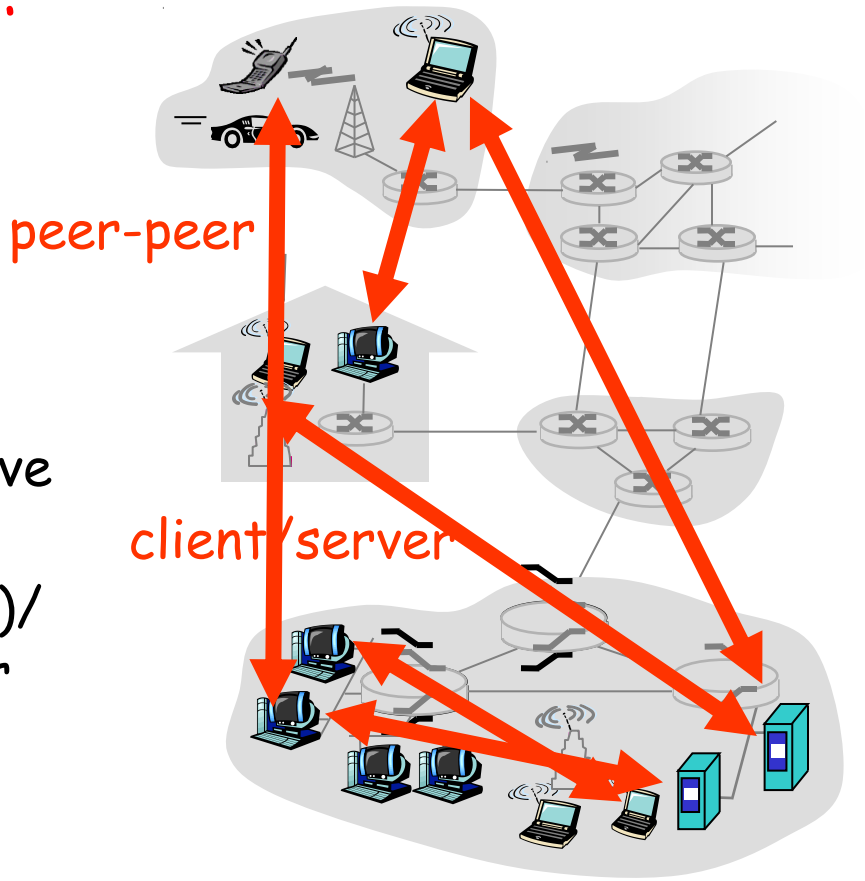
Una visione più da vicino della struttura della rete

- Confini della rete (network edge):
  applicazioni e hosts
- Nucleo della rete (network core):
  - routers
  - "rete di reti"
- Varietà di reti di accesso
  - mezzi trasmissivi
  - canali di comunicazione



Il bordo della rete (network edge)

- Sistemi terminali (hosts):
  - Eseguono programmi applicativi
  - Es: WWW, email
  - Agli estremi della rete
- Modello client/server
  - L'host client richiede, riceve servizio dal server
  - Es: WWW client (browser)/ server; email client/server
- Modello peer-peer:
  - interazione tra gli host simmetrica
  - Es: BitTorrent, Skype



Network edge: servizio di trasferimento dati affidabile

Obiettivo:

trasferimento di dati tra sistemi terminali

- *handshaking:* setup preliminare del trasferimento di dati
  - "Mi scusi" / "Si?" Nel caso del protocollo umano
  - *Creazione di uno "stato"* nei due hosts in comunicazione
- Trasferimento senza errori di dati su Internet

Servizio TCP [RFC 793]

Transmission Control Protocol

- *Trasferimento affidabile, in ordine di una sequenza di byte*
  - In caso di perdita di dati: ACKs e ritrasmissioni
- *Controllo di flusso:*
  - Trasmettitore non sovraccarica il ricevitore
- *Controllo di congestione:*
  - Trasmettitori rallentano quando la rete è congestionata

Network edge: servizio di trasferimento dati "best effort" (non affidabile)

Obiettivo:

trasferimento di dati tra sistemi terminali

- Come prima!

- UDP - User Datagram Protocol [RFC 768]:
  - Senza connessione
  - Trasferimento di dati non affidabile
  - no controllo di flusso
  - no controllo di congestione

Applicazioni che usano TCP:

- HTTP (Web), FTP (file transfer), ssh (login remoto), SMTP (email)

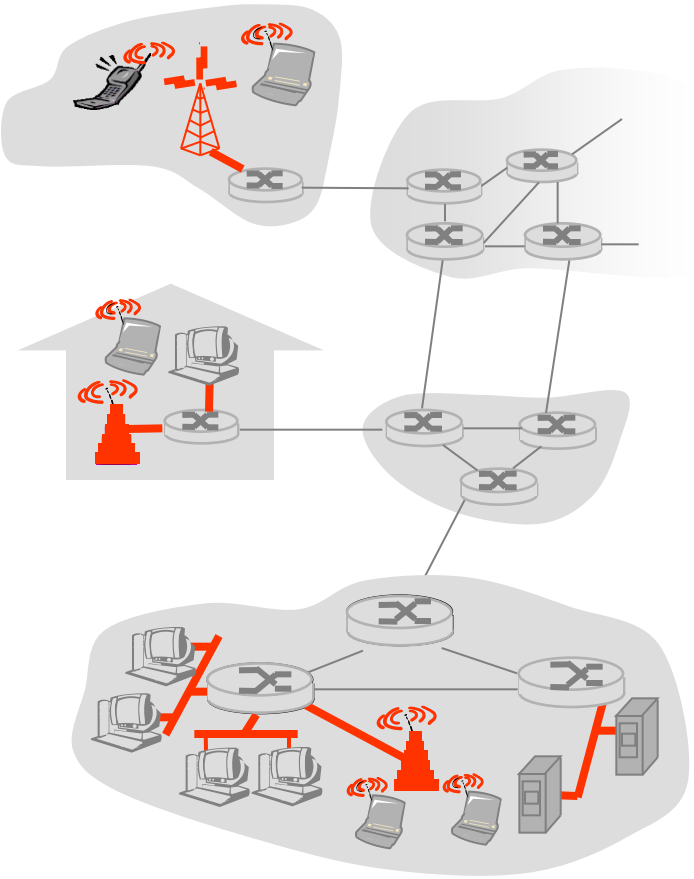
Applicazioni che usano UDP:

- streaming multimediale, videoconferenze, DNS, telefonia su Internet

Reti di accesso, mezzi trasmissivi

Domanda: Come collegare i sistemi terminali ai router di bordo?

- reti di accesso residenziali
- reti di accesso istituzionali (università, aziende)
- reti di accesso wireless

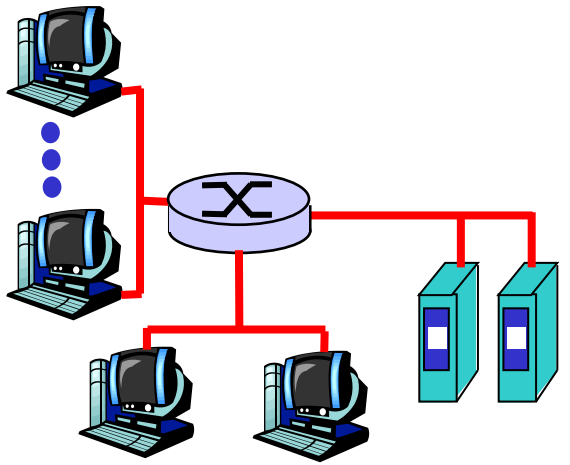


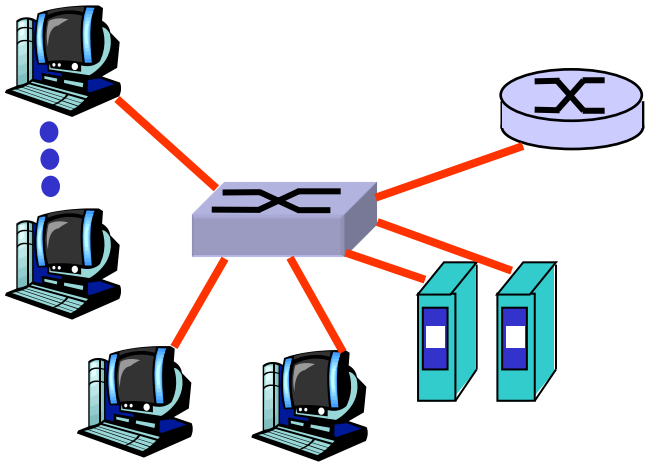
Ricordate:

- ampiezza di banda (bit al secondo) della rete di accesso?
- condivisa o dedicata?

Reti locali (LANs) istituzionali

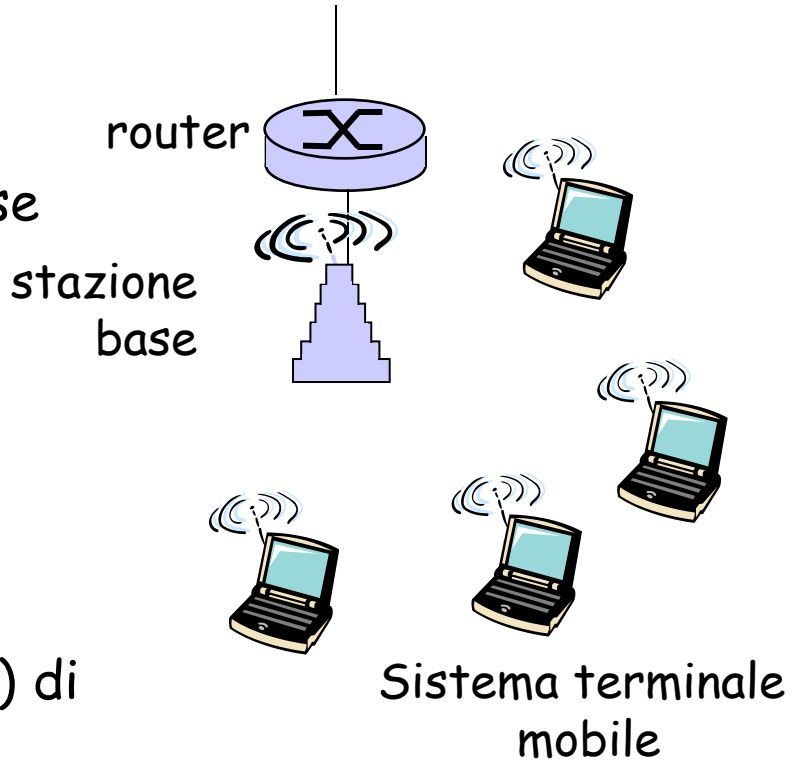
- local area network (LAN) connettono sistemi terminali ai router di bordo in università/enti/aziende
- Ethernet:
  - 10 Mbs, 100Mbps, 1Gbps, 10Gbps Ethernet
  - configurazione moderna: sistemi terminali connessi a *Ethernet switch*
- *Domanda:* switch o router?
  -
  -





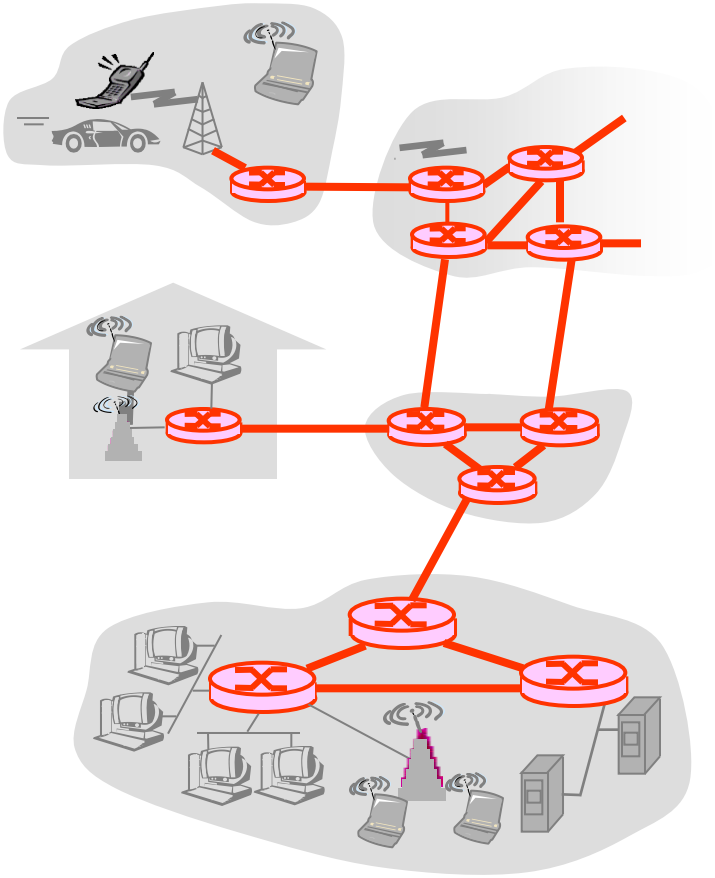
Reti di accesso wireless

- Rete di accesso su canale radio condiviso che connette sistemi terminali al router
  - Attraverso una stazione radio base (base station) o "access point"
- wireless LANs:
  - 802.11g/n (WiFi): 54, 540 Mbps
- rete d'accesso wireless geografica
  - gestita da un operatore (provider) di telecomunicazioni
  - ~10Mbps in reti cellulari (LTE)
  - In futuro (1Gbps): 5G ?



Il nucleo della rete

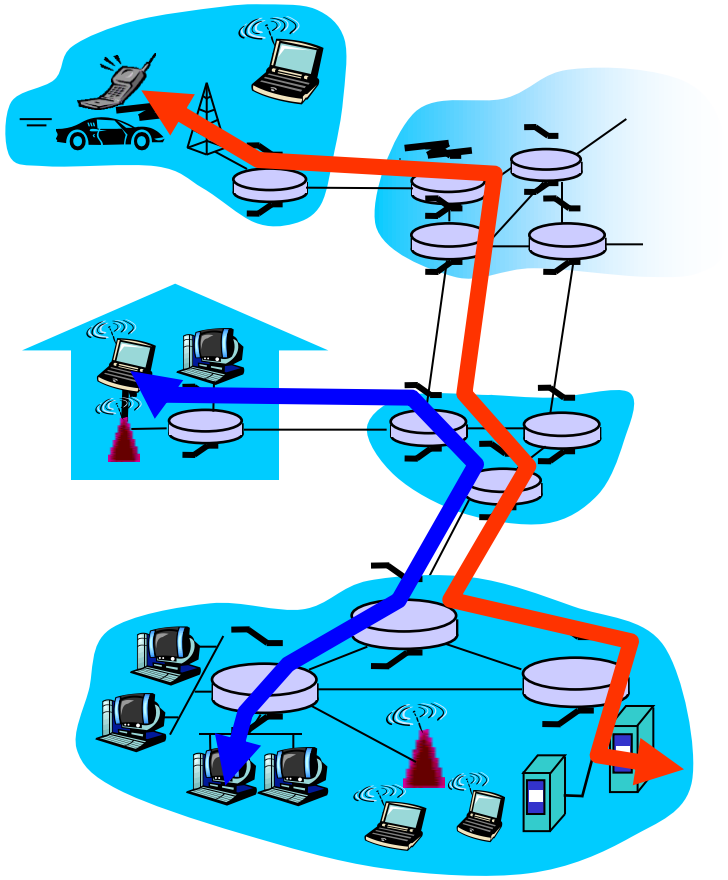
- Rete magliata di router che interconnettono i sistemi terminali
- il quesito fondamentale: come vengono trasferiti i dati attraverso la rete?
  - commutazione di circuito: circuito dedicato per l'intera durata della sessione
  - commutazione di pacchetto: dati di una sessione inviati attraverso la rete in parti discrete (pacchetti), utilizzando risorse al momento disponibili



Il nucleo della rete: commutazione di circuito

Risorse end-to-end dedicate per ogni connessione

- Ampiezza di banda, capacità dei communtatori
- Risorse dedicate: non c'è condivisione
- Prestazioni stile circuito (garantite)
- necessaria l'impostazione della chiamata (call setup)



Il nucleo della rete: commutazione di circuito

Le risorse della rete (es: ampiezza di banda) sono divise in "pezzi"

- I "pezzi" vengono allocati ai vari collegamenti
- Un pezzo di risorsa resta *inattivo* se non è utilizzato dalla connessione cui viene allocato (*non c'è condivisione*)

Domanda: come si suddivide la banda in pezzi?

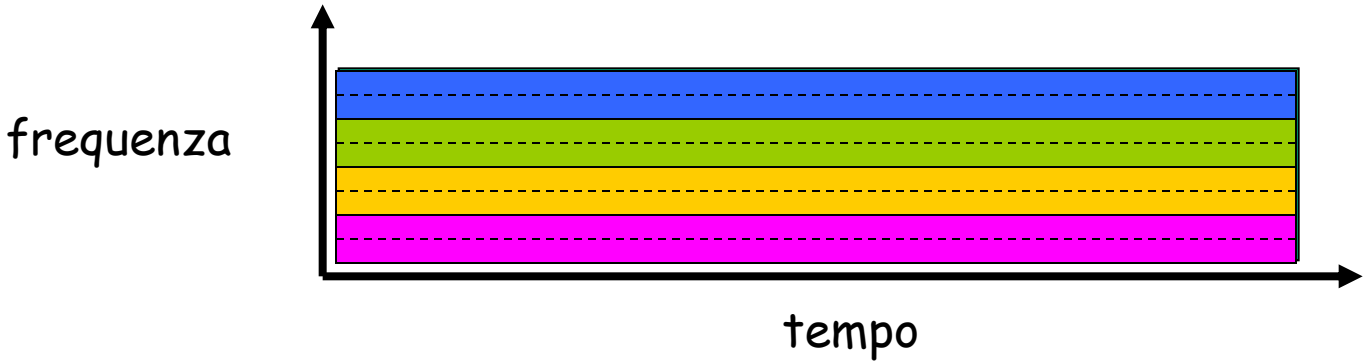
-
-

FDM e TDM

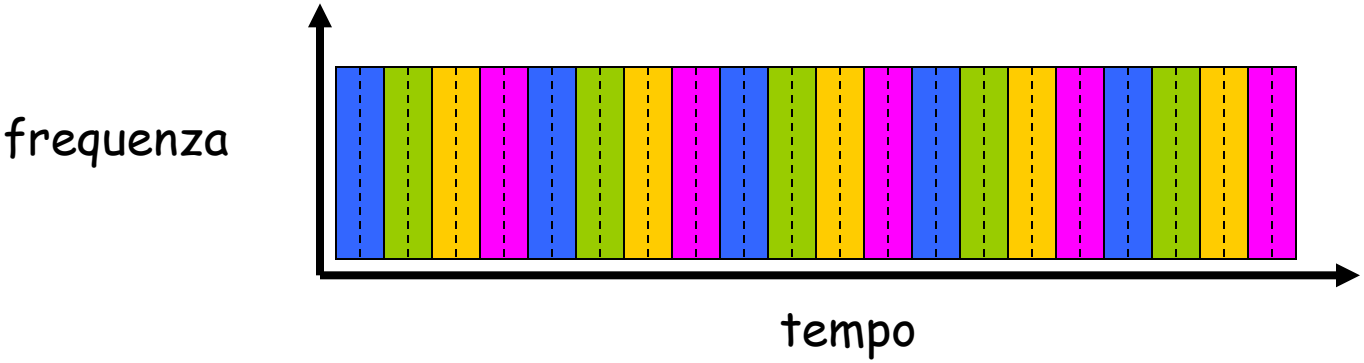
Esempio:

4 utenti

FDM



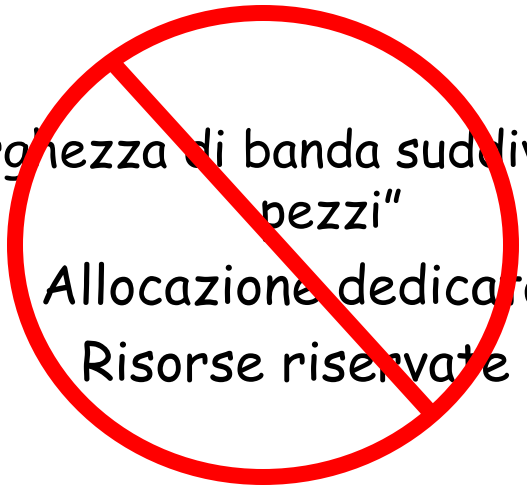
TDM



Il nucleo della rete: commutazione di pacchetto

Il flusso di dati punto-punto viene suddiviso in *pacchetti*

- i pacchetti degli utenti A e B *condividono* le risorse di rete
- ciascun pacchetto utilizza completamente il canale
- Le risorse vengono usate *a seconda delle necessità*



Contesa per le risorse

- La richiesta di risorse può eccedere il quantitativo disponibile
- **congestione**: accodamento dei pacchetti, attesa per l'utilizzo del collegamento
- Modalità **store and forward**: il commutatore deve
  - ricevere l'intero pacchetto
  - aspettare che il collegamento in uscita sia disponibile
  - trasmettere il pacchetto sul collegamento in uscita

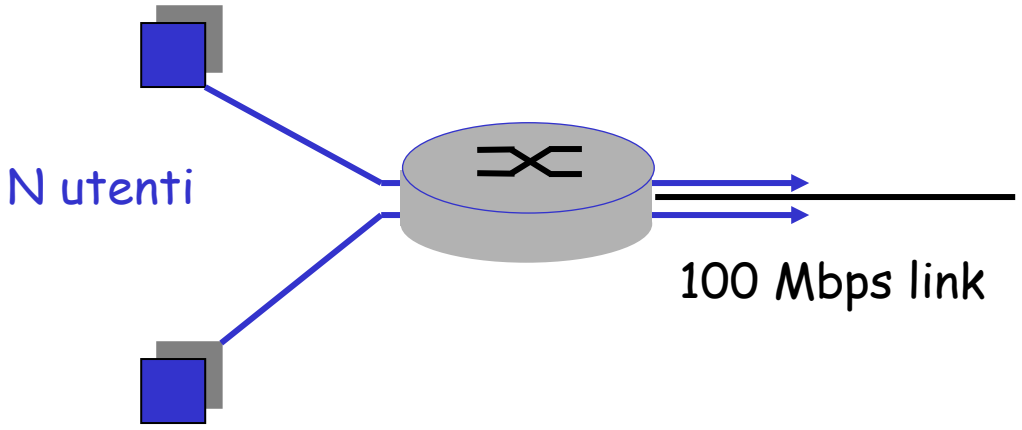
Commutazione di circuito vs commutazione di pacchetto: vantaggi e svantaggi

- commutazione di circuito:
  -
- commutazione di pacchetto:
  -

Comm. di circuito vs comm. di pacchetto

La commutazione di pacchetto permette a più utenti di usare la rete!

- Singolo collegamento a 100 Mb/s link
- Ogni utente:
  - Consuma 10 Mb/s quando è "attivo"
  - Attivo per il 10% del tempo
- Commutazione di circuito:
  - 10 utenti
- Commutazione di pacchetto:
  - con 35 utenti, la probabilità di averne > 10 attivi è inferiore allo 0,0004

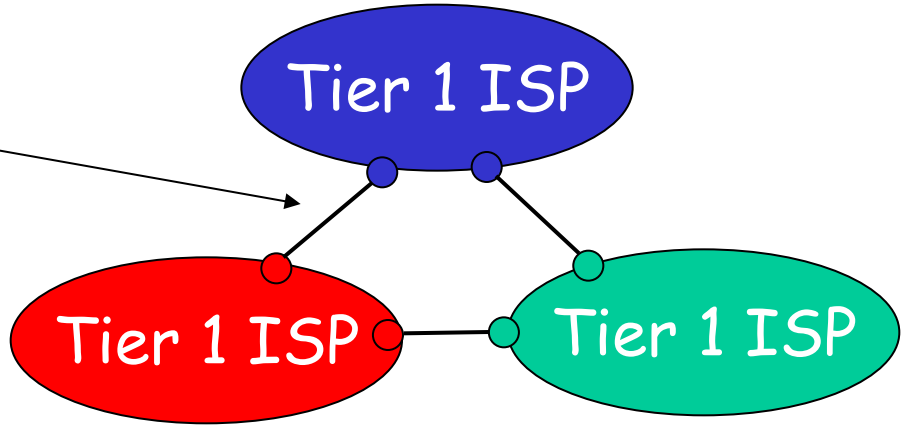


D: come è stato ottenuto il valore 0,0004?

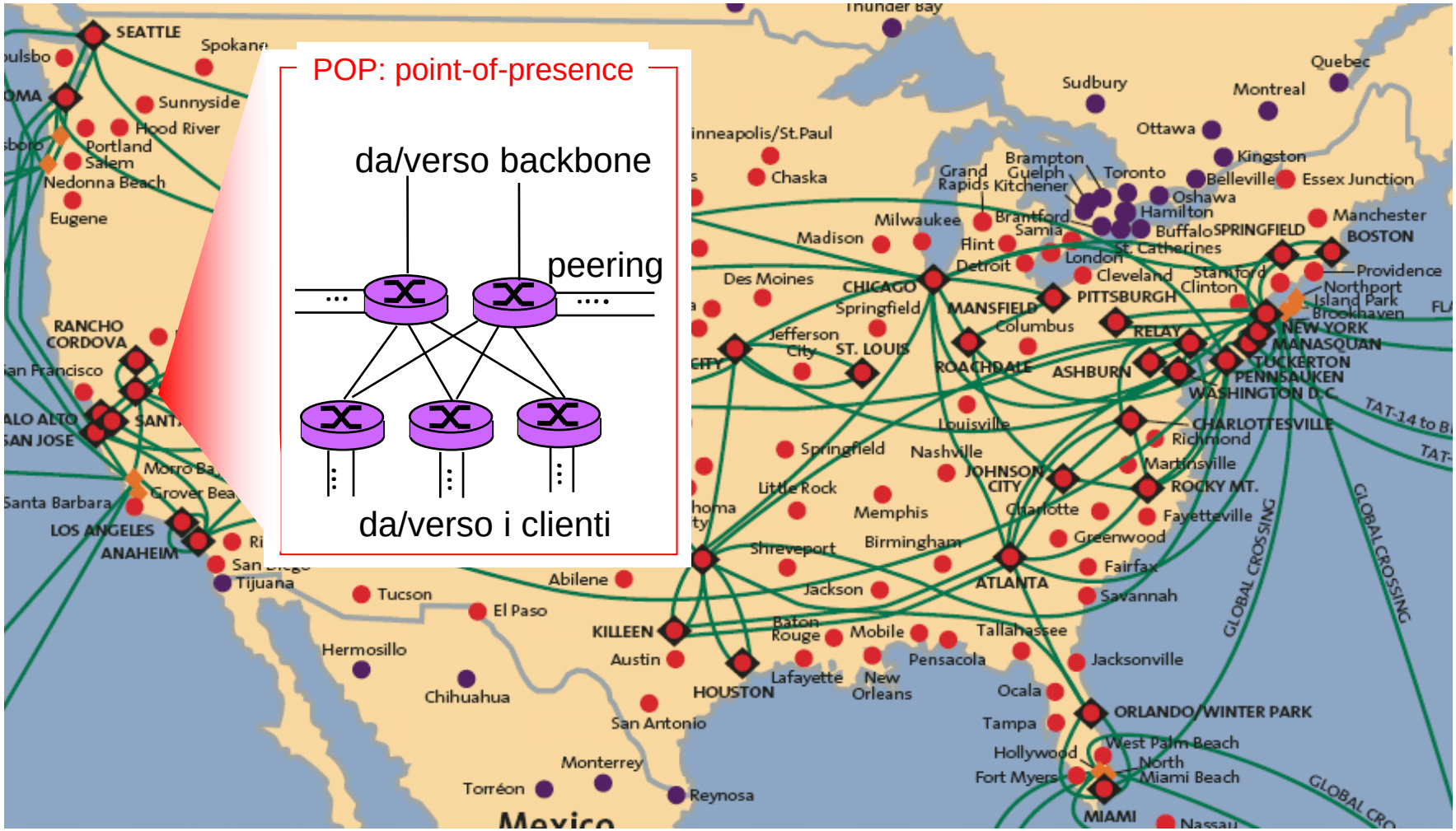
Struttura di Internet: rete di reti

- Essenzialmente gerarchica
- Al centro: "ISP di livello 1" o "reti dorsali di Internet" (e.s., Sprint, AT&T, Geant), copertura nazionale/internazionale
  - Si trattano gli uni con gli altri alla pari (peer)

Operatori di livello 1 si interconnettono privatamente

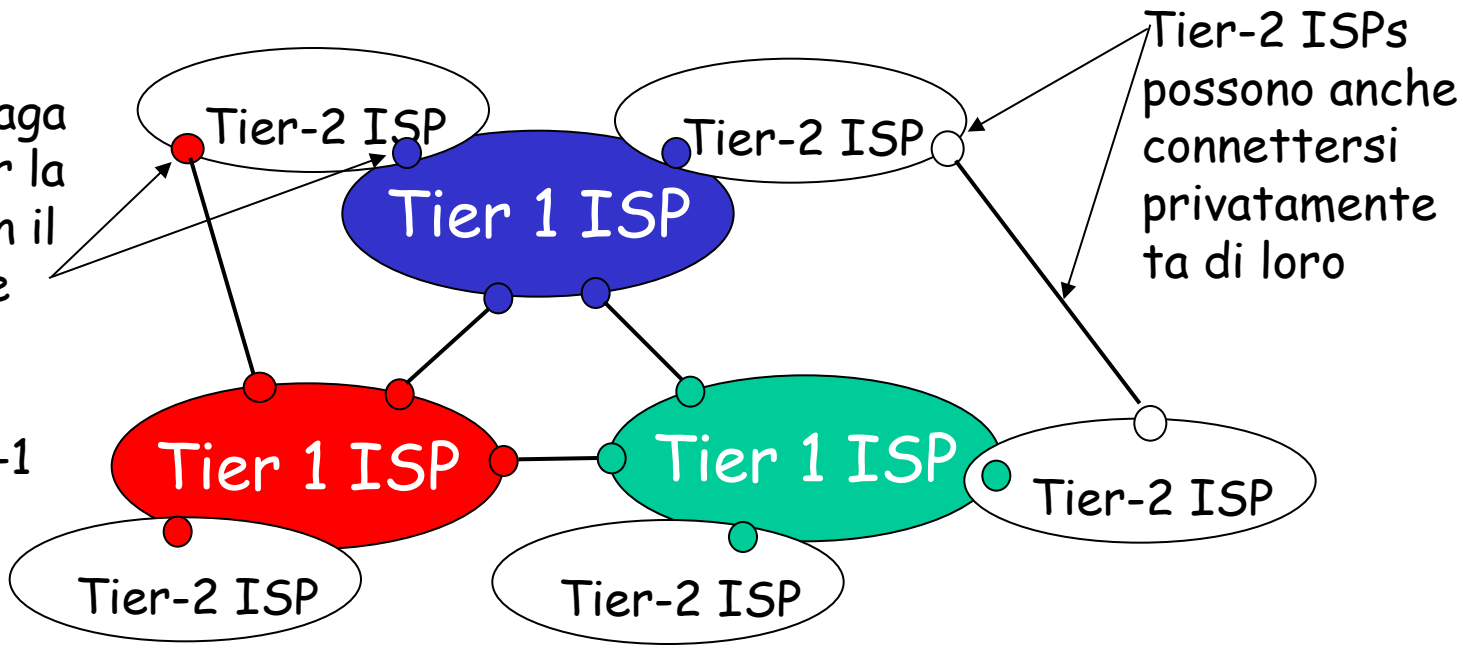


ISP di livello 1: es.: Sprint



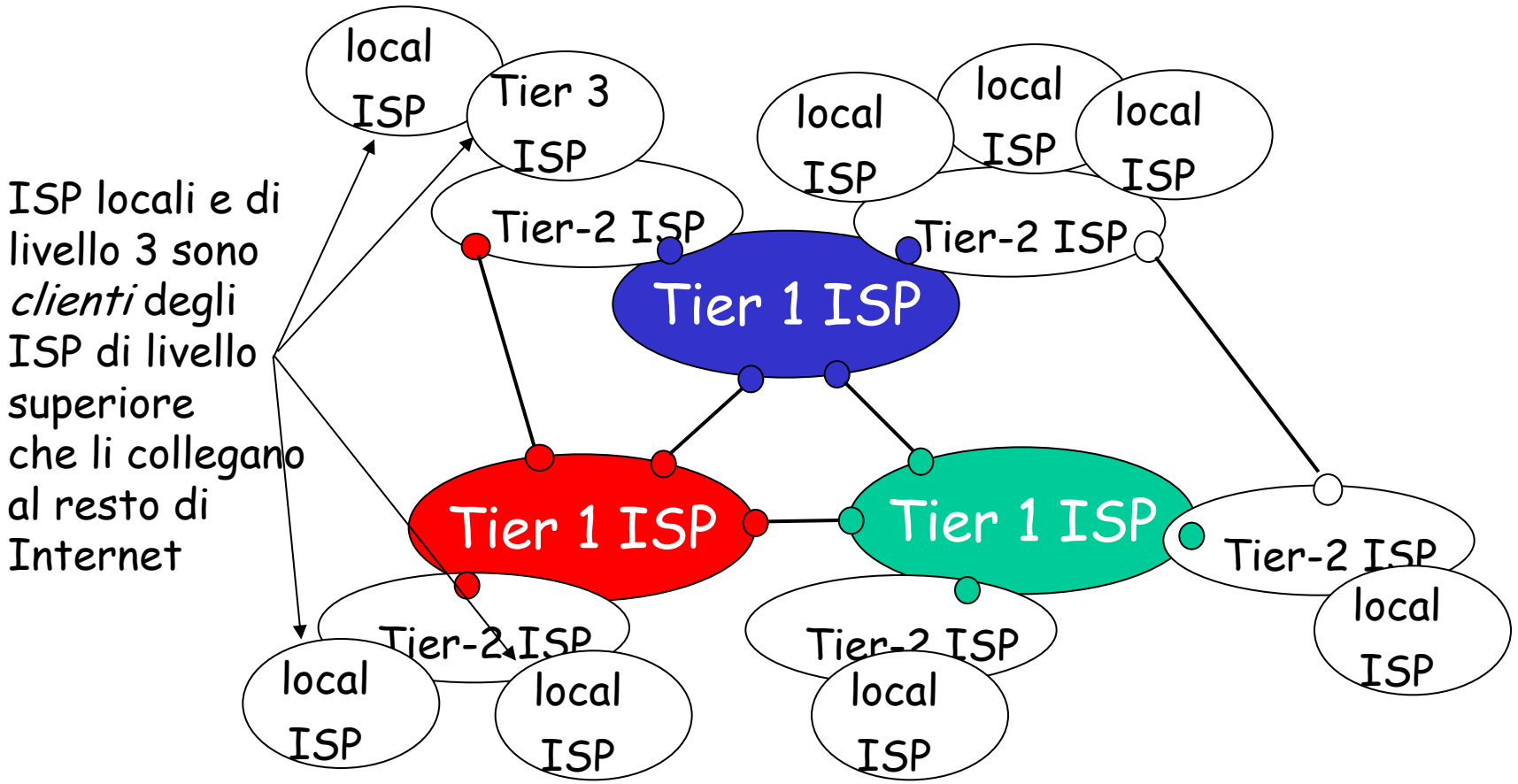
Struttura di Internet: rete di reti

- ISP di livello 2: ISP più piccoli (nazionali o distrettuali)
  - Si connettono a uno o più ISP di livello 1, e possibilmente ad altri ISP di livello 2



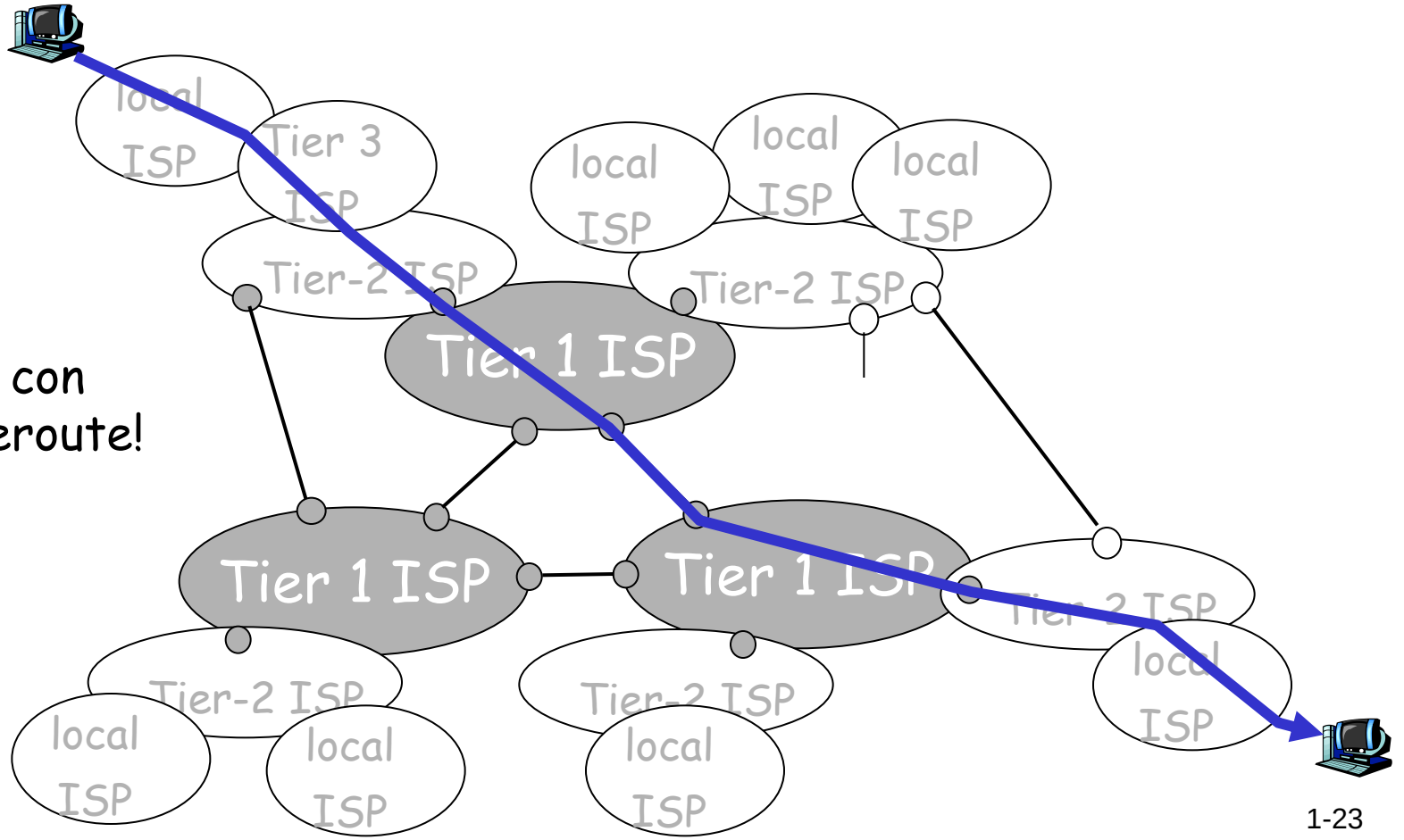
Struttura di Internet: rete di reti

- ISP di livello 3 e ISP locali (ISP di accesso)



Struttura di Internet: rete di reti

- Un pacchetto passa attraverso molte reti!



La stratificazione dei protocolli

Le reti sono complesse!

- molti"pezzi":
  - Sistemi terminali
  - svariate tipologie di mezzi trasmissivi
  - applicazioni
  - protocolli
  - hardware, software

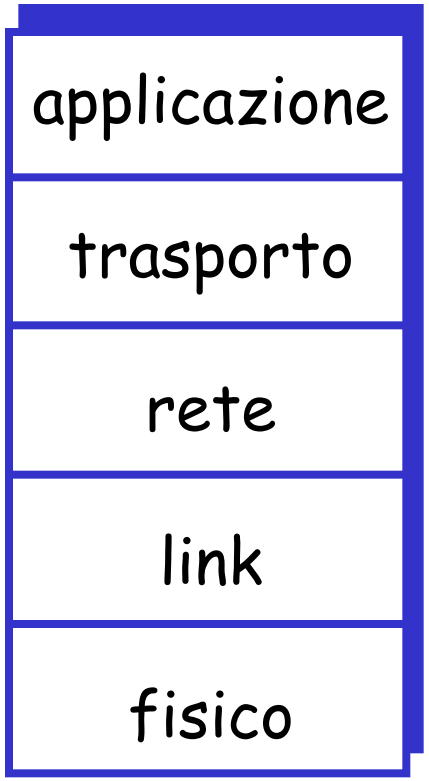
Domanda:

C'è qualche speranza di *organizzare* l'architettura delle reti?

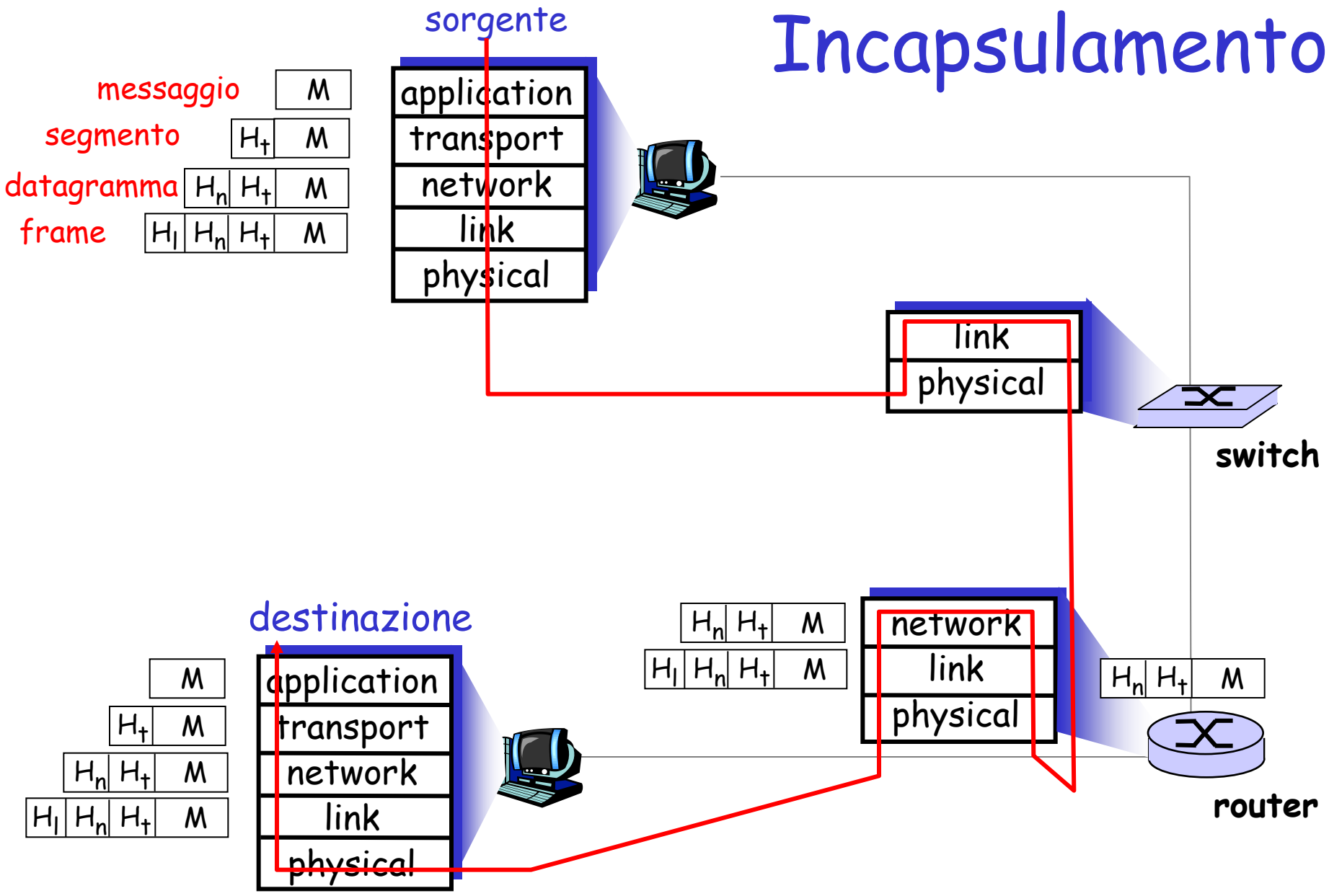
O almeno la nostra trattazione sulle reti?

La pila protocollare di Internet

- applicazione: supporta le applicazioni di rete
  - ftp, smtp, http, BitTorrent
- trasporto: trasferimento end-to-end dei datagrammi
  - TCP, UDP
- rete: instradamento dei datagrammi da sorgente a destinazione
  - IP, protocolli di istradamento
- link (collegamento): trasferimento dei datagrammi tra due nodi consecutivi della rete
  - ppp, ethernet
- fisico: trasferimento dei singoli bit sul canale



Incapsulamento



Ripasso sulle reti

Obiettivi:

- Richiamare concetti chiave del corso introduttivo sulle reti di calcolatori
  - Rinfrescare la memoria su idee fondamentali
  - Creare una base di partenza comune
  - Identificare possibili lacune e lavoro di ripasso
  - Consolidare la terminologia

Sommario:

- Panoramica ad alto-livello
- Controllo di errore
- Controllo di flusso
- Controllo di congestione
- Indirizzamento
- Livello rete
- Livello link
- Controllo

Ripasso sulle reti

Obiettivi:

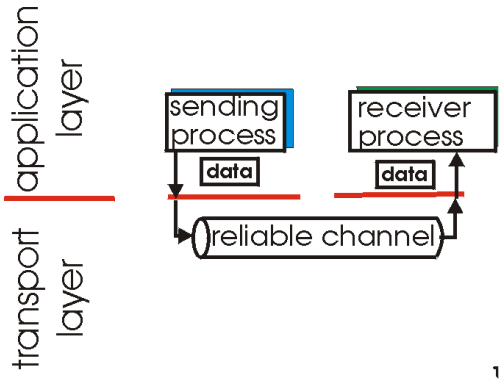
- Richiamare concetti chiave del corso introduttivo sulle reti di calcolatori
  - Rinfrescare la memoria su idee fondamentali
  - Creare una base di partenza comune
  - Identificare possibili lacune e lavoro di ripasso
  - Consolidare la terminologia

Sommario:

- Panoramica ad alto-livello
- Controllo di errore
- Controllo di flusso
- Controllo di congestione
- Indirizzamento
- Livello rete
- Livello link
- Controllo

Controllo di errore

- Comunicazioni punto-punto affidabili
  - Un problema *generico*: app-to-app, su path, su singolo link
- Modello di errore?
  - bits scambiati entro il pacchetto
  - Pacchetti persi
  - Pacchetti ritardati o riordinati

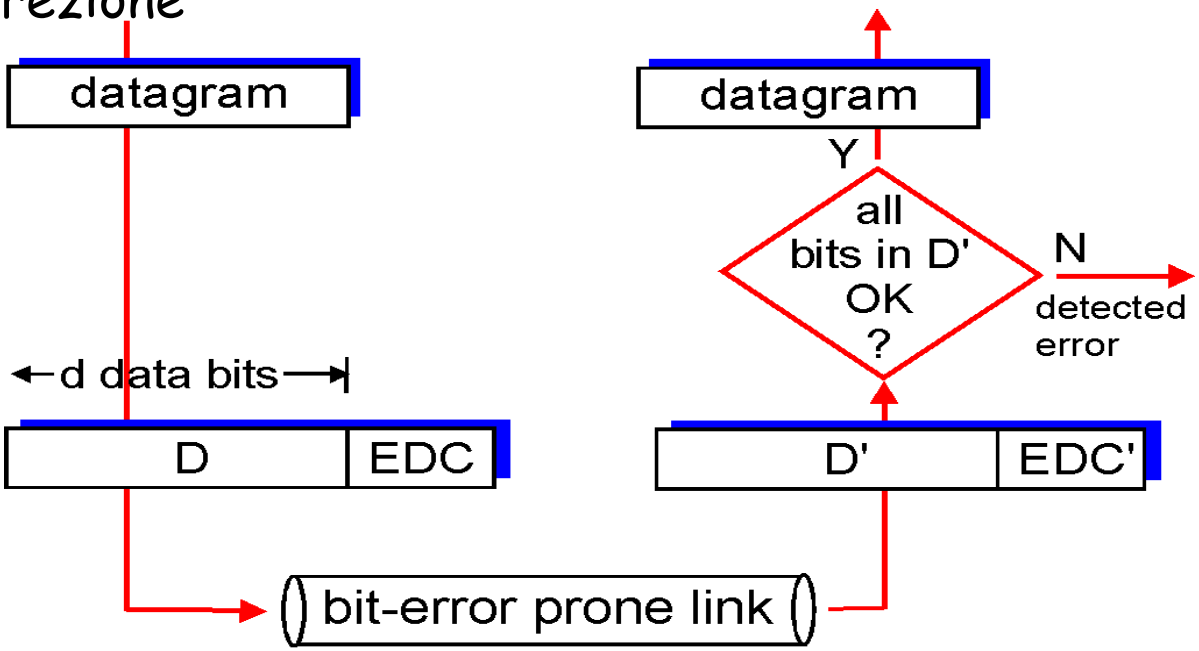


servizio fornito

Rilevamento di errori a livello di bit

EDC= Error Detection and Correction bits (bit ridondanti)
D = Dati protetti dal controllo di errore, incluse eventuali intestazioni (headers)

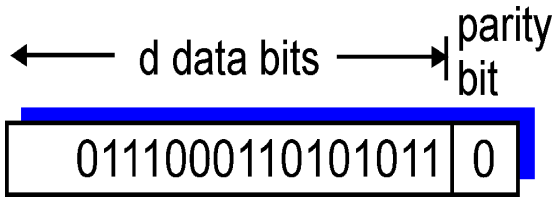
- Il rilevamento di errore non è affidabile al 100%!
  - il protocollo può non rilevare alcuni errori, ma è un evento raro
  - pià ampio il campo EDC migliori le capacità di rilevamento e correzione



Controllo di parità (Parity Checking)

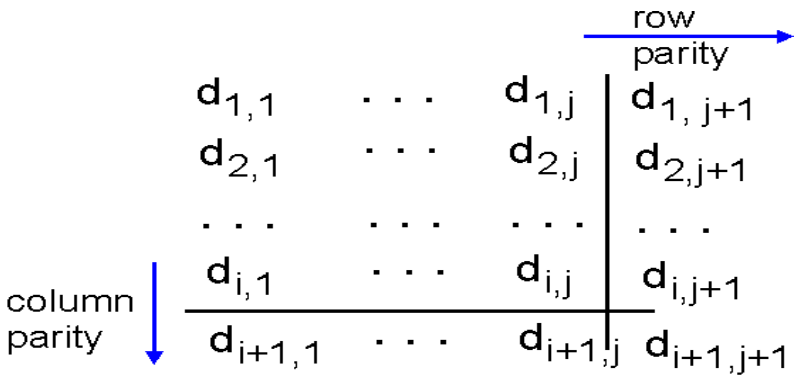
Singolo bit di parità

Rileva errori su singolo bit



Parità bi-dimensionale:

Rileva e *corregge* errori su singolo bit



Esistono schemi molto più sofisticati e potenti per fare rilevamento/correzione di errori su molteplici bits:
Es: Cyclic Redundancy Check (CRC)

Internet checksum

Obiettivo: rilevare "errori" (ovvero: bit alterati) nei segmenti trasmessi (nota: usato a livello trasporto)

Mittente:

- Tratta il contenuto del segmento come una sequenza di interi da 16 bit
- checksum: somma (in complemento a 1) gli interi contenuti nel segmento
- Il mittente pone il valore della checksum nel campo checksum del segmento

Ricevente:

- calcola la checksum del segmento ricevuto
- controlla se la checksum calcolata è uguale al valore del campo checksum:
  - No - errore rilevato
  - Sì - nessun errore rilevato. *Ma potrebbero esserci errori nonostante questo?*

Recupero dei pacchetti persi

- Perchè si possono perdere pacchetti?
  - memoria finita, pacchetti scartati in caso di congestione
  - malfunzionamenti di rete: portano a nuovo instradamento che aggiri elementi guasti (sperabilmenti in tempi ~sec)
  - scartati dal sistema terminale (e.s.: dalla scheda di ricezione)
- ARQ: Automatic Repeat reQuest
  - mittente inserisce nei pacchetti dei numeri di sequenza (perchè?)
  - ricevente manda ACKs o NACKs in risposta ai pacchetti ricevuti
  - mittente fa partire un timer (logico) per ogni pacchetto, con meccanismo di timeout e ritrasmissione

rdt3.0: canali con errori *e* perdite

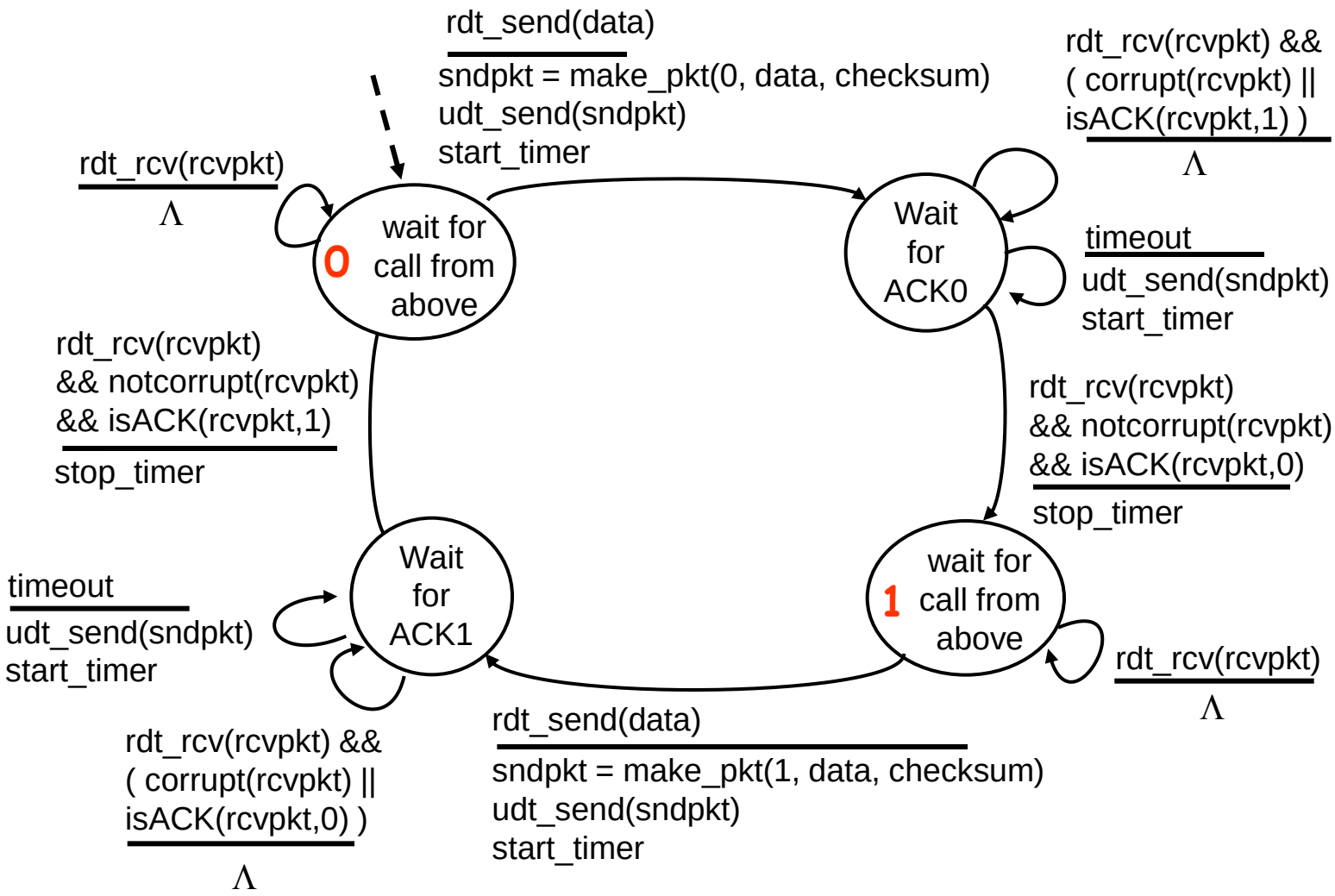
Ipotesi: il canale sottostante può corrompere, perdere pacchetti (dati o ACKs)

- occorrono checksum, numeri di sequenza, ACKs, ritrasmissioni, timer
- numeri di sequenza
  - Rilevano riordinamento
  - Rilevano pacchetti mancanti
  - Rilevano pacchetti duplicati (per ritrasmissione)

Approccio: mittente attende un tempo "ragionevole" di ricevere l'ACK

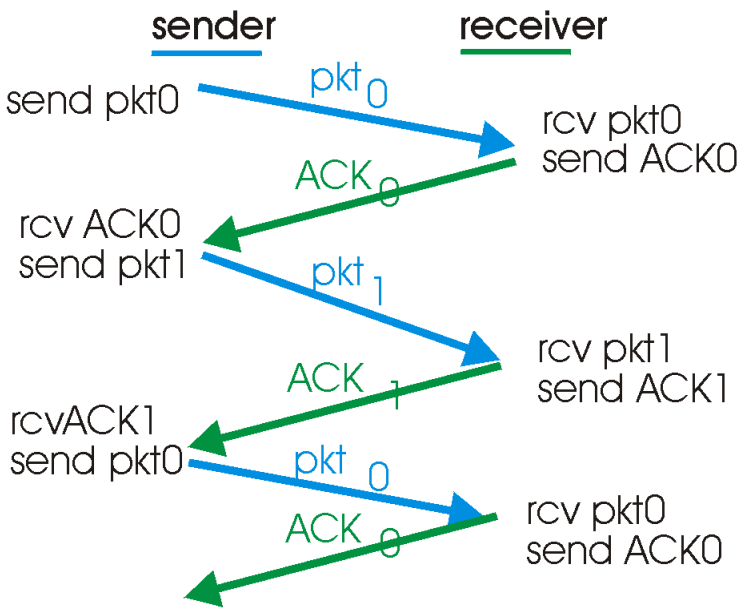
- Ritrasmette se ACK non ricevuto entro questo tempo
- Se pacchetto (dati o ACK) viene semplicemnete ritardato (non perso):
  - La ritrasmissione genera un *duplicato,* ma l'uso dei # di seq. gestisce questo caso
  - Ricevitore deve specificare il # di seq. del pacchetto nel ACK
- Occorrono timer al mittente

rdt3.0 lato mittente

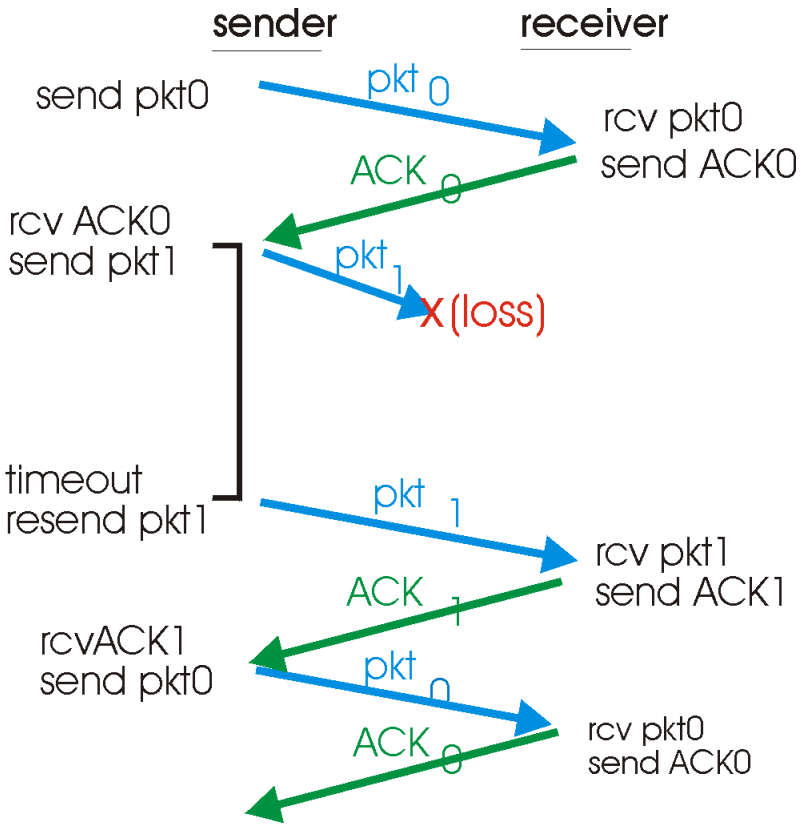


Macchina a stati finita del mittente (dettagli non importanti)

rdt3.0 in azione

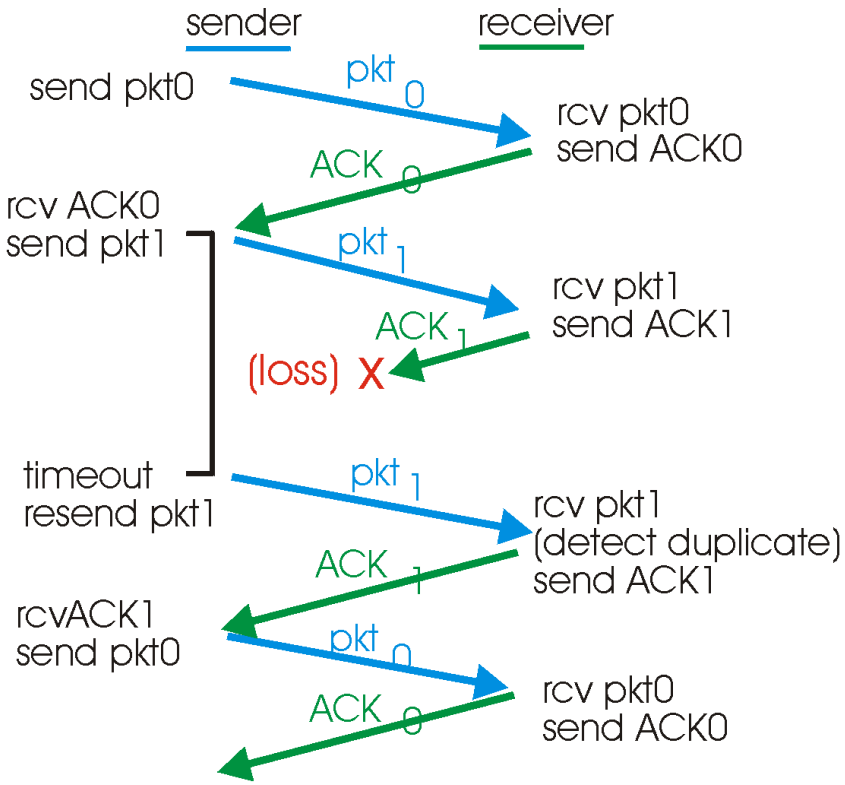


(a) operation with no loss

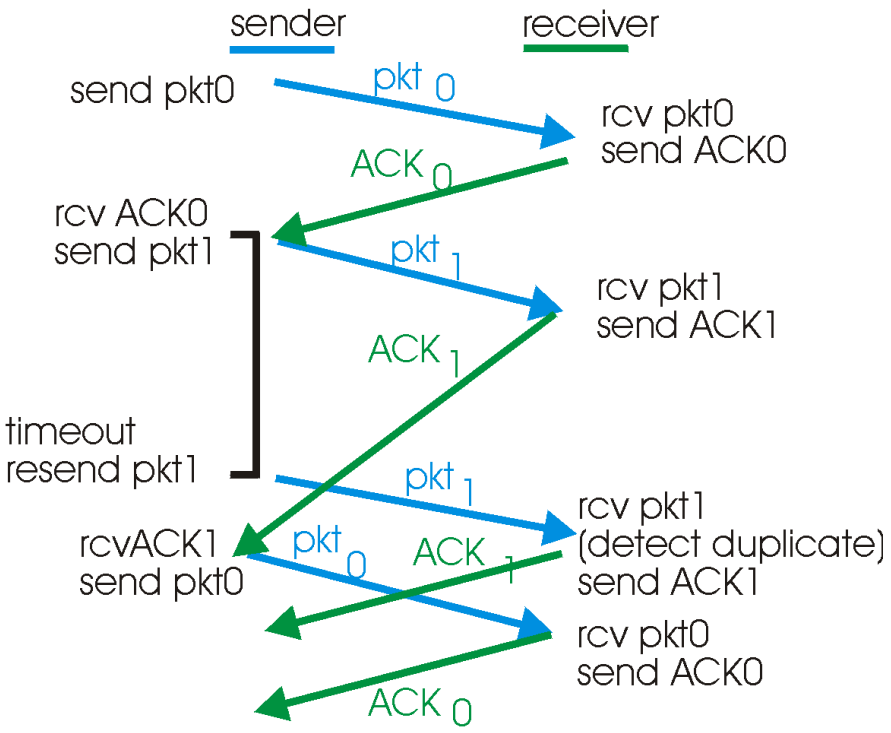


(b) lost packet

rdt3.0 in azione



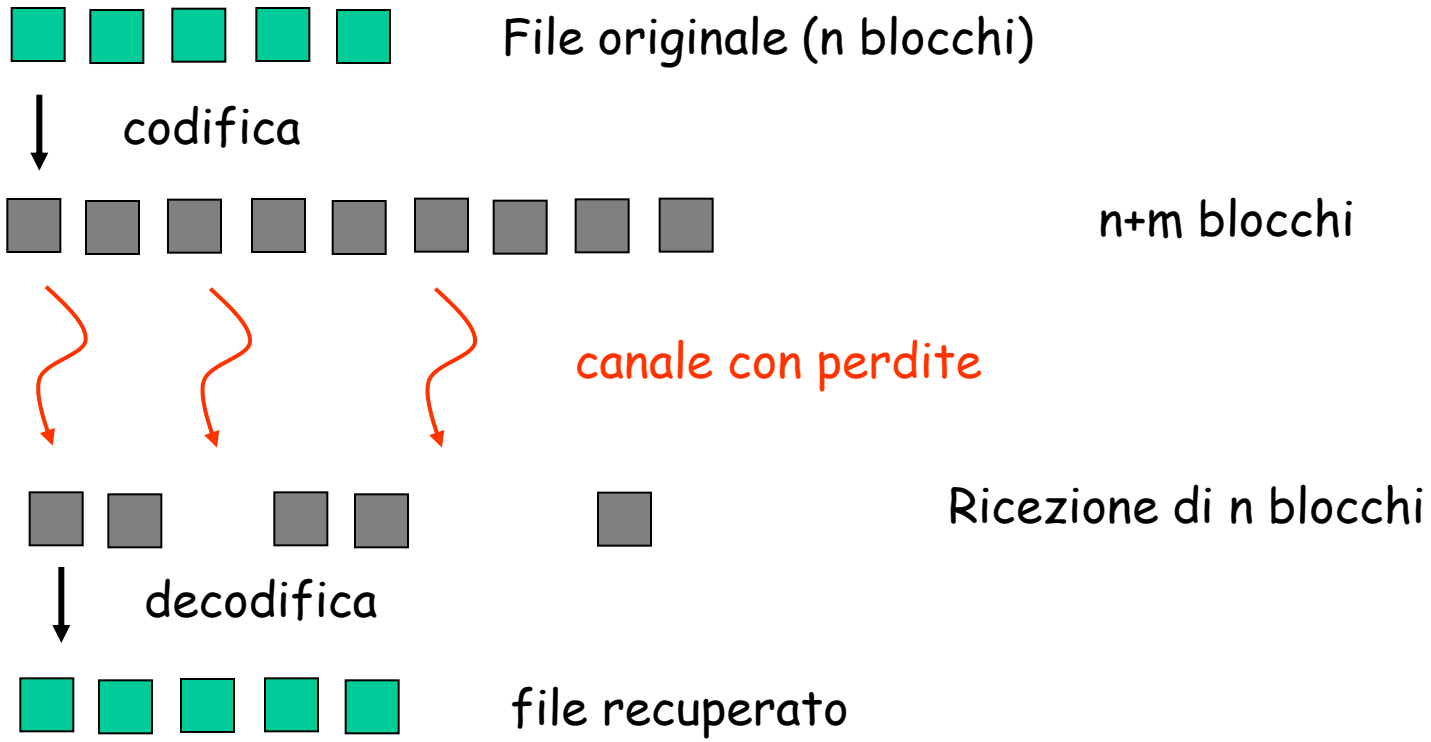
(c) lost ACK



(d) premature timeout

Controllo di errore in avanti (Forward error control)

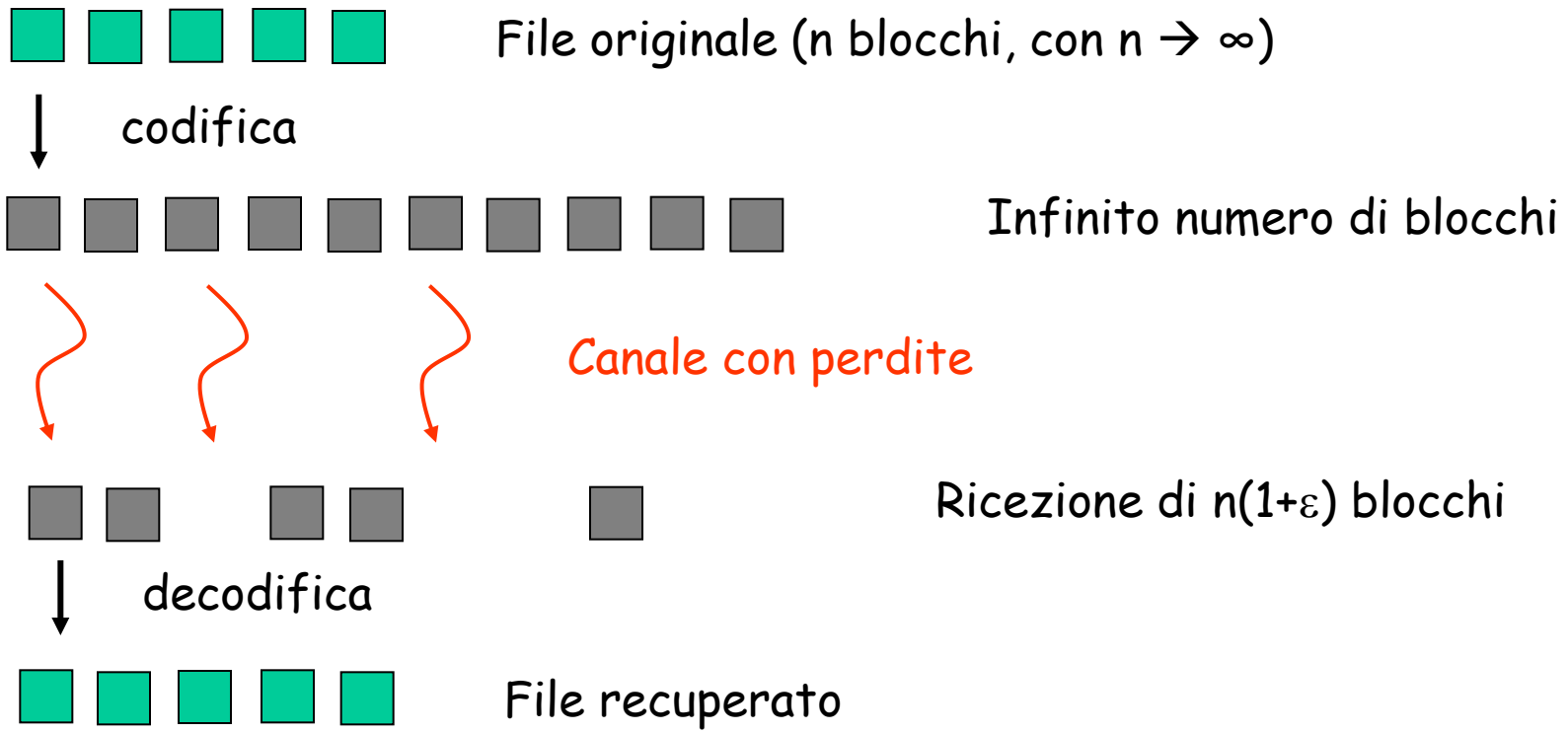
- Aggiunta di ridondanza per recuperare le perdite



Codici Reed-Solomon O(nm)

Forward error control

Se ci sono più di m perdite?



Forward error control

- I codici "rateless" permettono sequenze infinite di blocchi
  - Codici LT/Rapture, $O(n \ln 1/\varepsilon)$
- ε controlla costo computazionale, richiesta di banda
- Usato per distribuzione di video; trasferimento di grandi files
- network coding

Ripasso sulle reti

Obiettivi:

- Richiamare concetti chiave del corso introduttivo sulle reti di calcolatori
  - Rinfrescare la memoria su idee fondamentali
  - Creare una base di partenza comune
  - Identificare possibili lacune e lavoro di ripasso
  - Consolidare la terminologia

Sommario:

- Panoramica ad alto-livello
- Controllo di errore
- Controllo di flusso
- Controllo di congestione
- Indirizzamento
- Livello rete
- Livello link
- Controllo

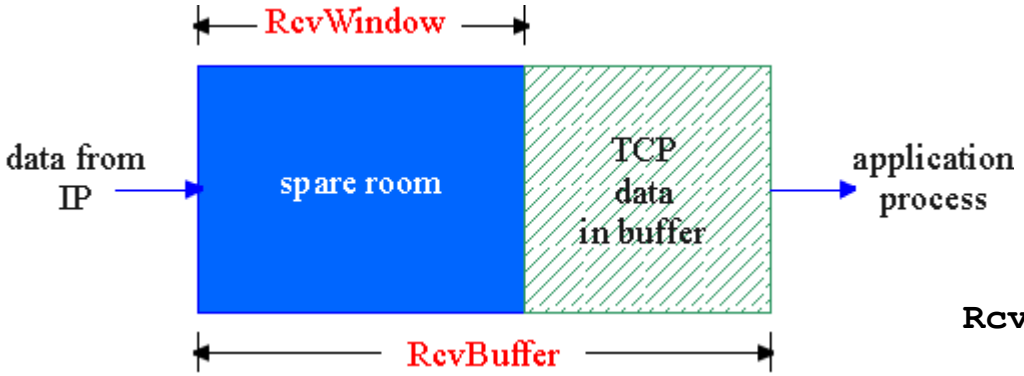
Controllo di flusso (in TCP)

Controllo di flusso
mittente non deve sovraccaricare i buffer del ricevitore trasmettendo troppi dati, troppo velocemente

ricevente: informa esplicitamente il mittente (dinamicamente nel tempo) la quantità di memoria libera nel buffer

- campo **RcvWindow** in segmenti TCP

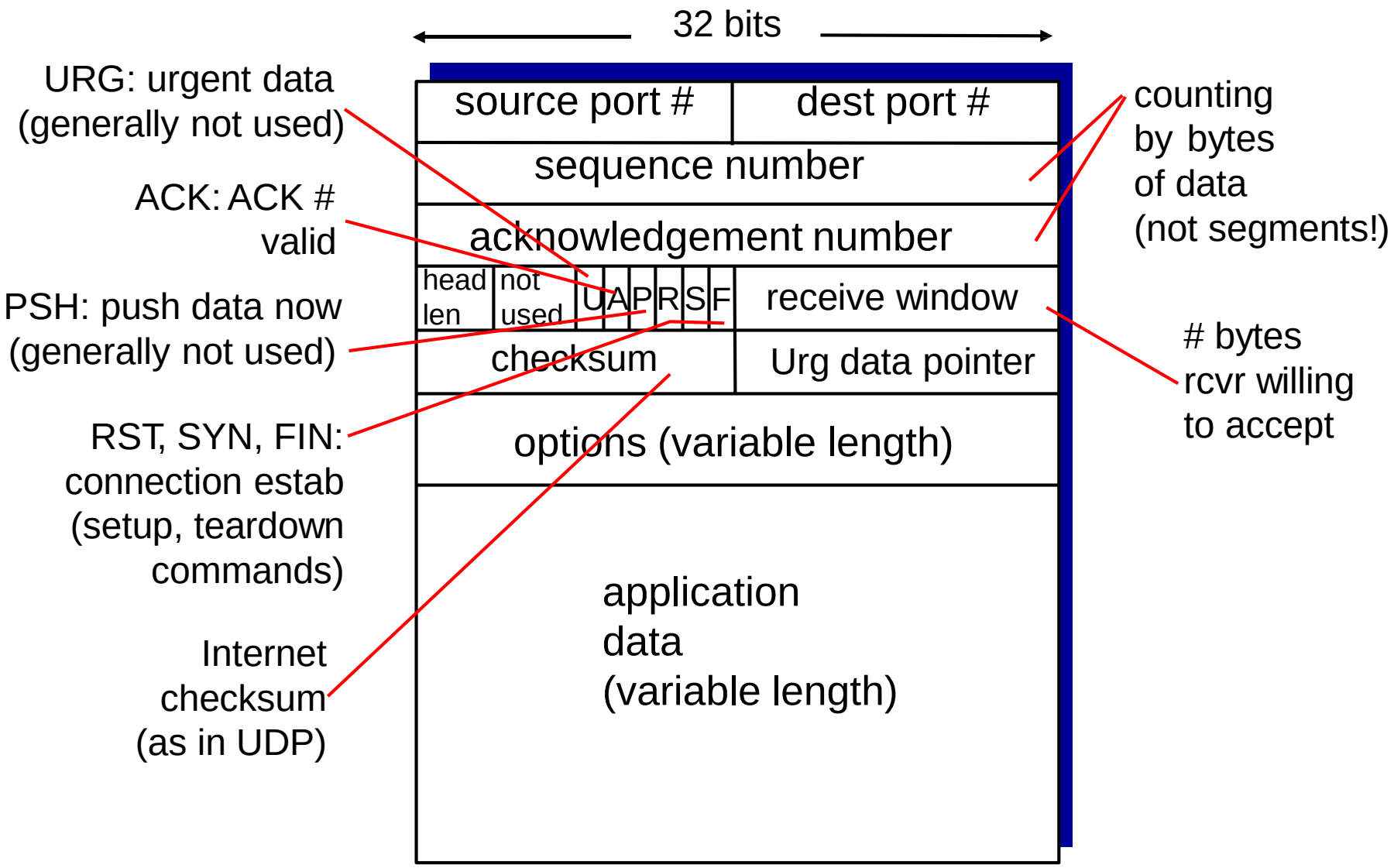
mittente: mantiene la quantità di dati trasmessi, non ancora riscontrati da ACKs, più piccola del più recente valore di **RcvWindow** ricevuto



buffer del ricevitore

`RcvBuffer` = dimensione del buffer di ricezione TCP
`RcvWindow` = quantità di spazio libero nel buffer

Struttura del segmento TCP



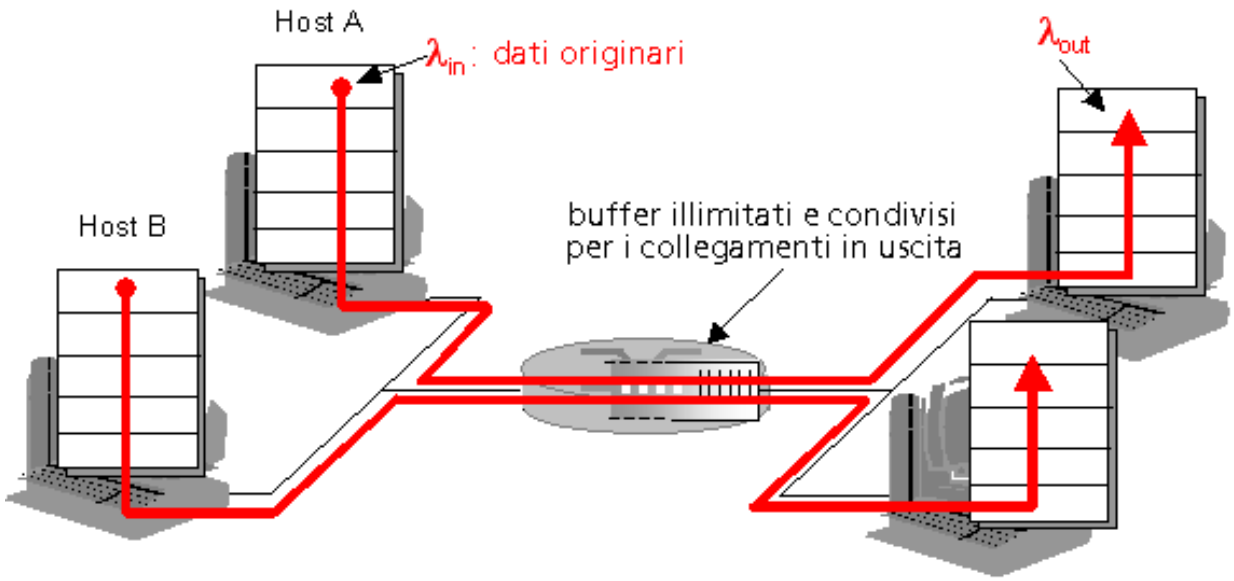
Principi del controllo di congestione

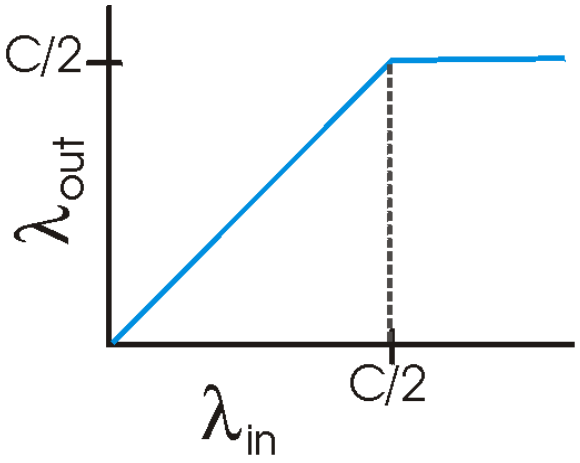
Congestione:

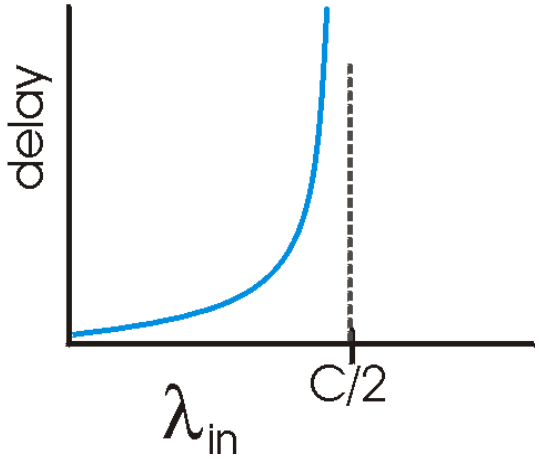
- informalmente: "troppe sorgenti trasmettono troppi dati, a una velocità talmente elevata che la *rete* non è in grado di gestirli"
- diverso dal controllo di flusso!
- si manifesta con:
  - pacchetti persi (overflow nei buffer dei router)
  - lunghi ritardi (accodamento nei buffer dei router)

Cause/costi della congestione: scenario 1

- due mittenti, due destinatari
- un router con buffer illimitati
- nessuna ritrasmissione



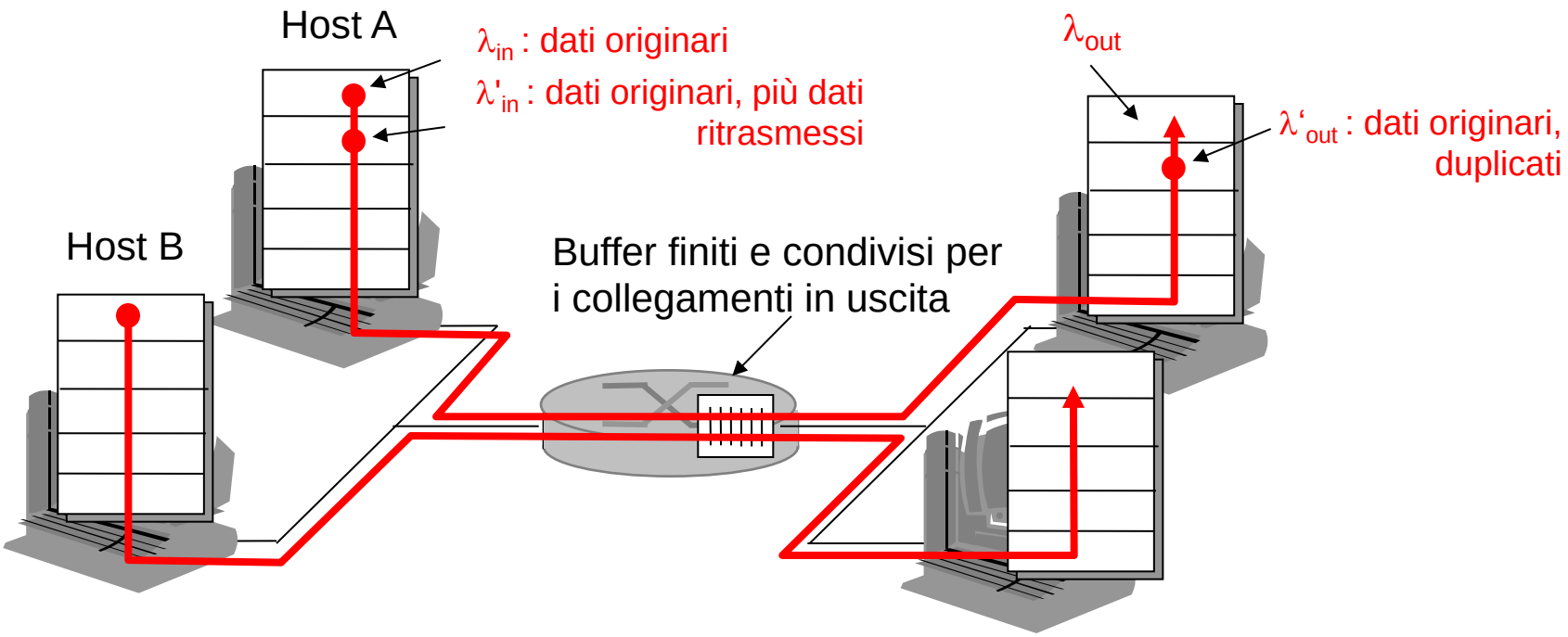




- grandi ritardi in caso di congestione
- throughput massimo

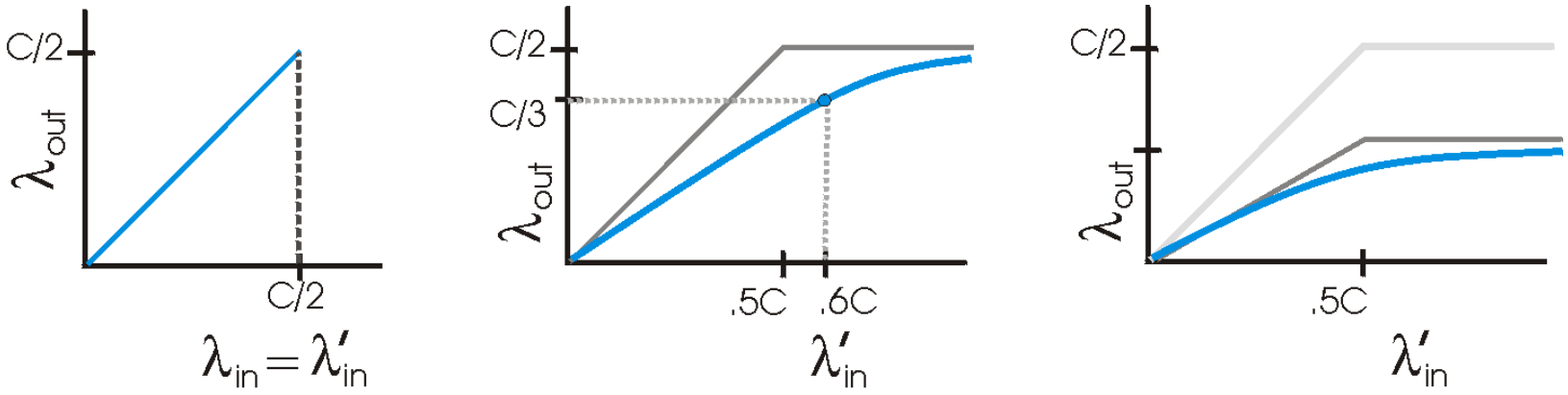
Cause/costi della congestione: scenario 2

- un router, buffer *finiti*
- il mittente ritrasmette i pacchetti persi



Cause/costi della congestione: scenario 2

- sempre: $\lambda_{in} = \lambda_{out}$ (goodput)
- ritrasmissione "perfetta" (solo di pacchetti persi): $\lambda_{in} = \lambda'_{out}$
- la ritrasmissione di pacchetti ritardati (nom persi) rende λ'_{out} più grande (rispetto al caso perfetto) di λ_{out}



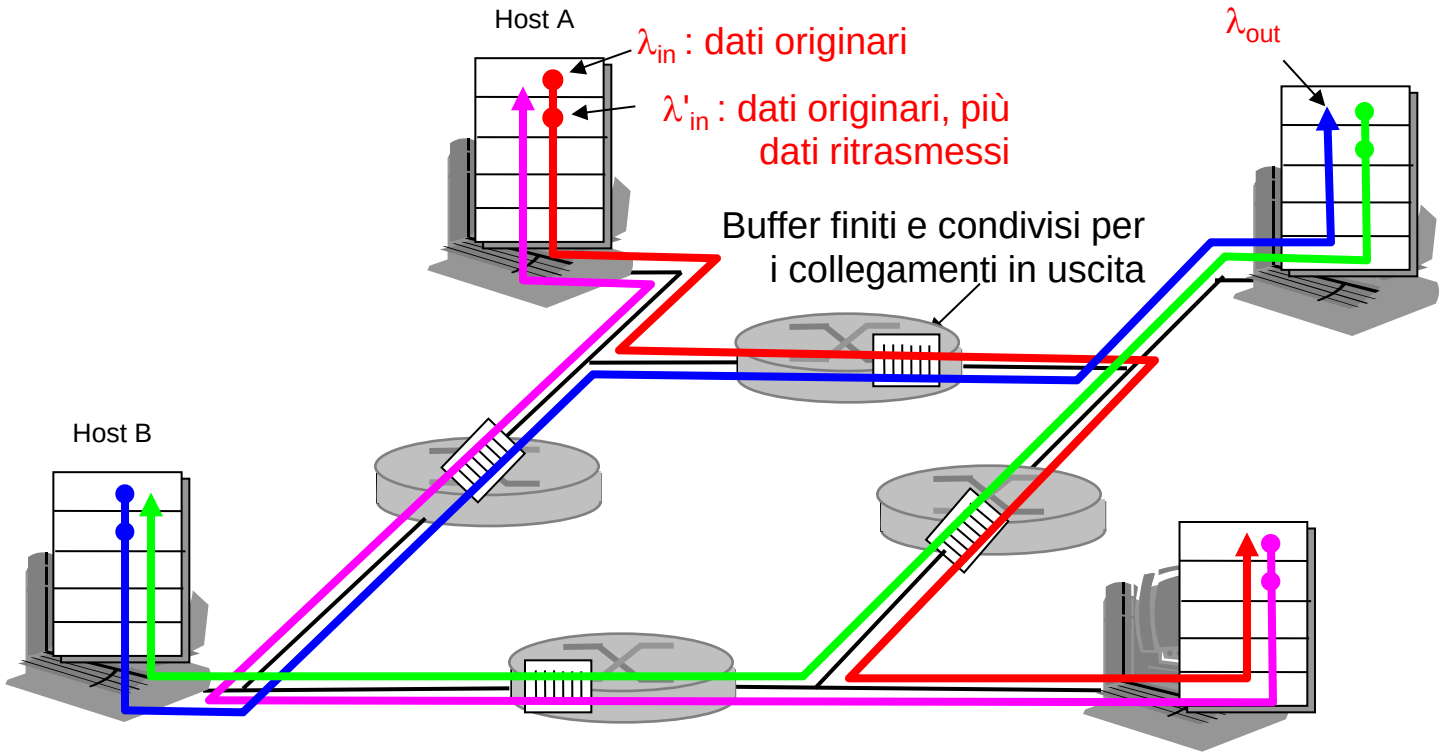
"costi" della congestione:

- Più lavoro (ritrasmissioni) per un dato "goodput"
- Ritrasmissioni non necessarie: il collegamento trasporta più copie di uno stesso pacchetto

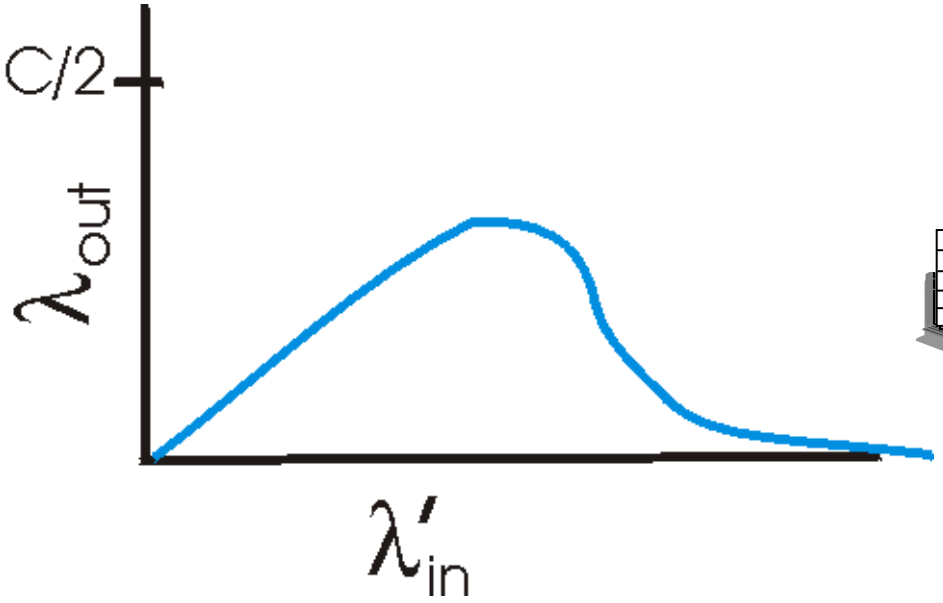
Cause/costi della congestione: scenario 3

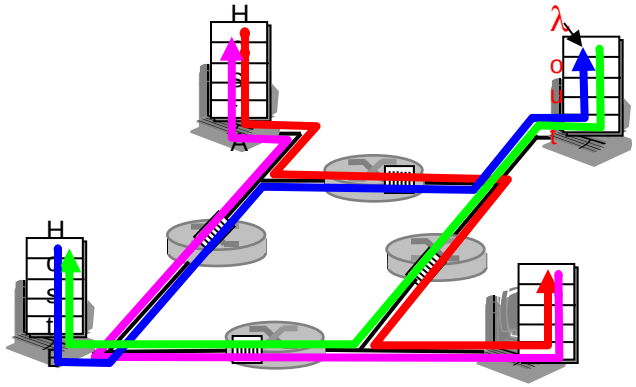
- quattro mittenti
- percorsi multihop
- timeout/ritrasmissioni

D: cosa accade quando λ_{in} e λ'_{in} aumentano?



Cause/costi della congestione: scenario 3





Altro "costo" della congestione:

- Quando un pacchetto viene scartato, la capacità trasmissiva usata nei link precedenti per quello stesso pacchetto viene sprecata

Due approcci al controllo di congestione

Controllo di congestione end-end:

- nessun supporto esplicito dalla rete
- la congestione è dedotta osservando le perdite e i ritardi nei sistemi terminali
- approccio adottato da TCP

Controllo di congestione assistito dalla rete :

- i router forniscono un feedback ai sistemi terminali
  - un singolo bit per indicare la congestione (SNA, DECbit, TCP/IP ECN, ATM)
  - notifica esplicita della velocità a cui il mittente dovrebbe trasmettere

Esempio: controllo di congestione ATM ABR

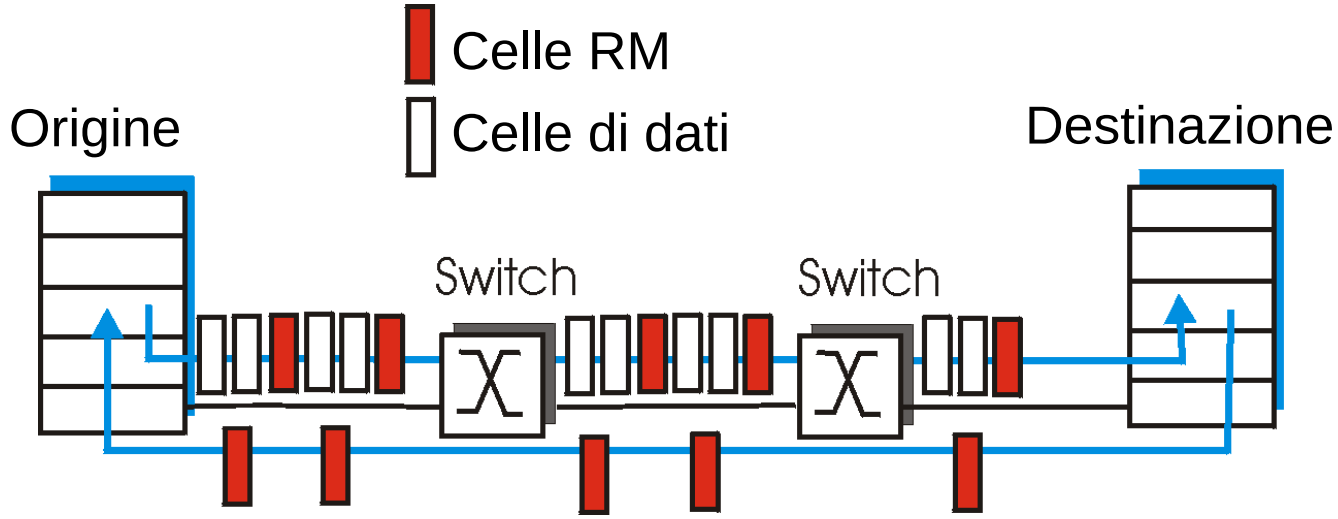
ABR: available bit rate:

- "servizio elastico"
- se il percorso del mittente è "sottoutilizzato":
  - il mittente dovrebbe utilizzare la larghezza di banda disponibile
- se il percorso del mittente è congestionato:
  - il mittente dovrebbe ridurre al minimo il tasso trasmissivo

Celle RM (resource management):

- inviate dal mittente, inframmezzate alle celle di dati
- i bit in una cella RM sono impostati dagli switch ("*assistenza dalla rete*")
  - bit NI: nessun aumento del tasso trasmissivo (congestione moderata)
  - bit CI: indicazione di congestione (traffico intenso)
- il destinatario restituisce le celle RM al mittente con i bit intatti

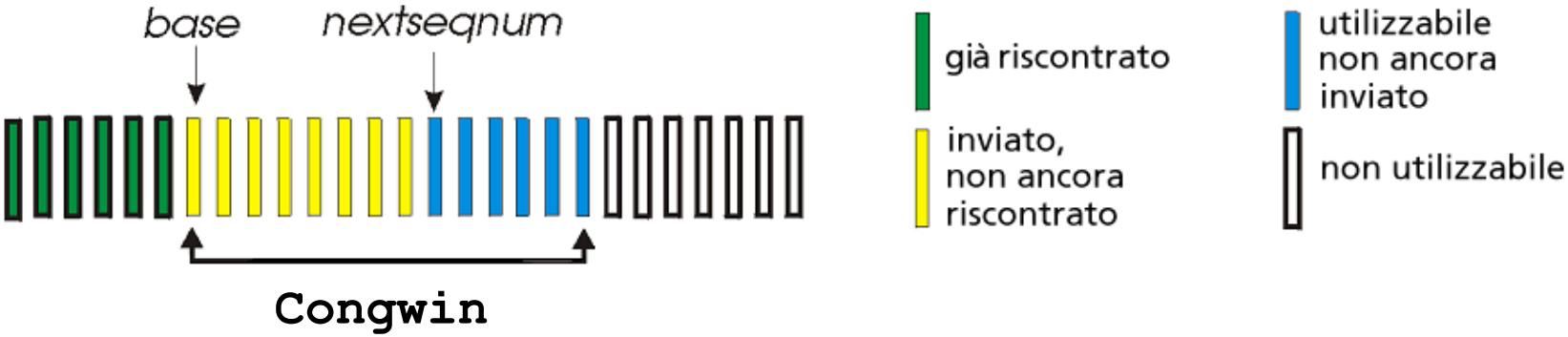
Esempio: controllo di congestione ATM ABR



- Campo esplicito per la velocità (ER, explicit rate) in ogni cella RM
  - lo switch congestionato può diminuire il valore del campo ER
  - in questo modo, il campo ER sarà impostato alla velocità minima supportabile da tutti gli switch sul percorso globale
- Ogni cella di dati contiene un bit EFCI, impostato a zero all'origine: modificato in 1 da uno switch congestionato
  - Se vale 1, viene lasciato così da tutti gli altri switch attraversati
  - La destinazione può reagire al bit EFCI per implementare meccanismi di controllo del flusso di origine, impostando il bit CI nella cella RM restituita

Controllo di congestione in TCP

- Approccio end-end (no assistenza dalla rete)
- Velocità di trasmissione limitata dalla congestion window size, `Congwin`, indicata nei segmenti:



Controllo di congestione in TCP

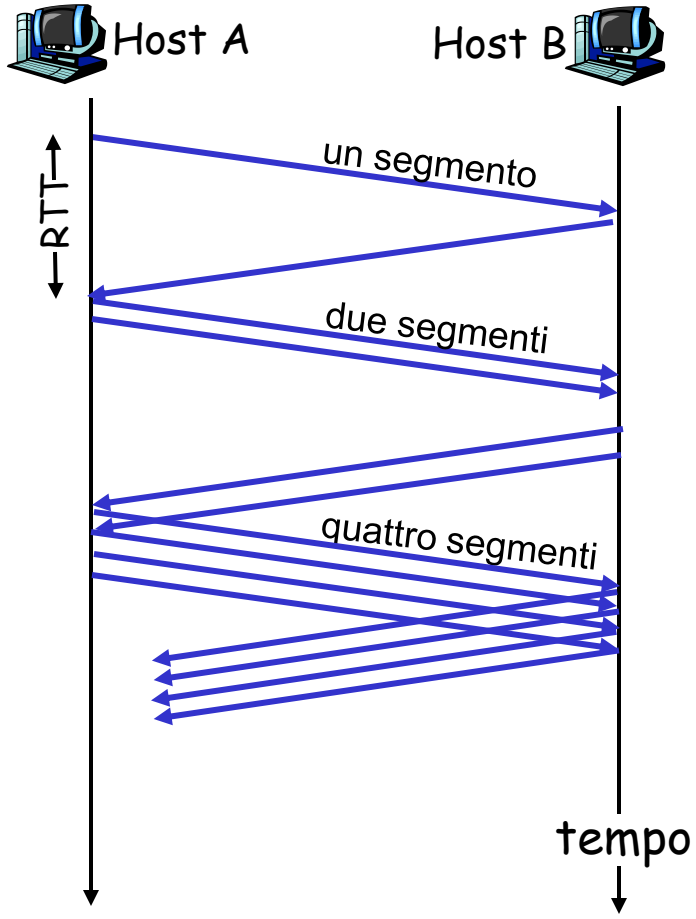
- "scoperta" della banda disponibile:
  - idealmente: trasmissione alla massima velocità possibile (in base a **Congwin**) senza creare perdite
  - *Si aumenta* **Congwin** fino a generare una perdita (per congestione)
  - Dopo la perdita: si *diminuisce* **Congwin**, poi la si aumenta di nuovo (nuova fase di scoperta)
- due "fasi"
  - slowstart
  - congestion avoidance
- variabili importanti:
  - **Congwin**
  - **threshold:** definisce la soglia tra la fase di slowstart e quella di congestion avoidance

TCP Slowstart

Algoritmo Slowstart

initialize: Congwin = 1
for (per ogni segmento ACKed)
    Congwin++
until (evento di perdita OR
    CongWin > threshold)

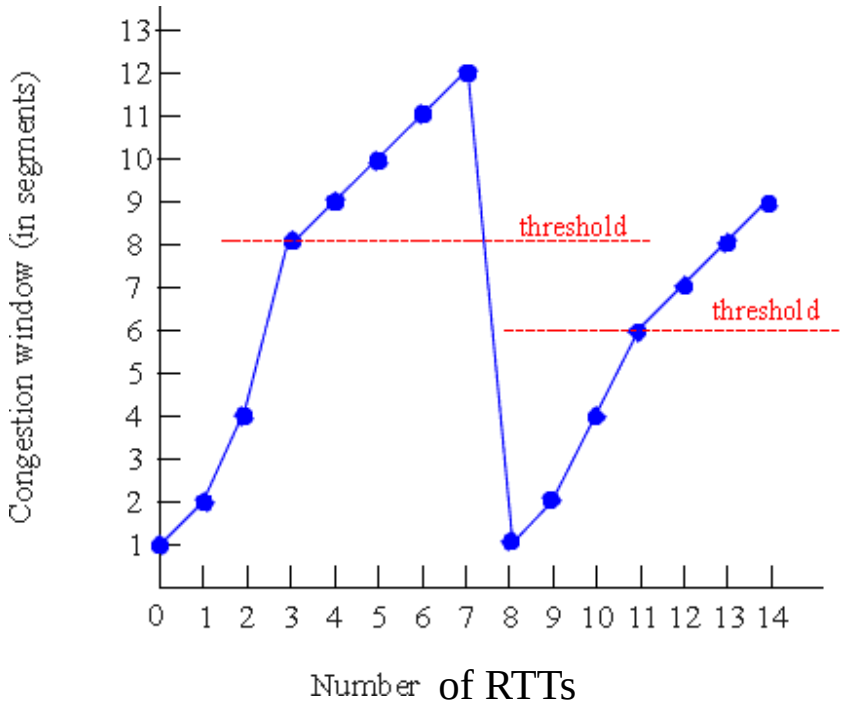
- crescita esponenziale (per RTT) della finestra (non così slow!)
- Evento di perdita: timeout (Tahoe TCP) e/o tre ACKs duplicati (Reno TCP)



TCP Congestion Avoidance: Tahoe

TCP Tahoe Congestion avoidance

```
/* (finita la fase slowstart) */
/* Congwin > threshold */
Until (evento di perdita) {
  ogni Congwin segmenti
ACKed:      Congwin++
  }
threshold = Congwin/2
Congwin = 1
esegui slowstart
```

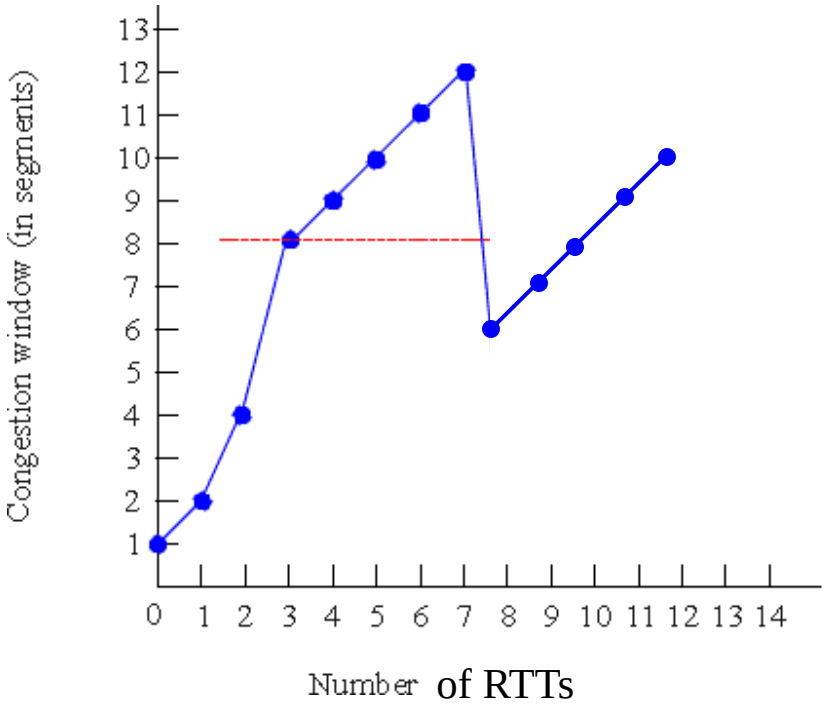


Numerosi miglioramenti: TCP Reno, TCP SACK

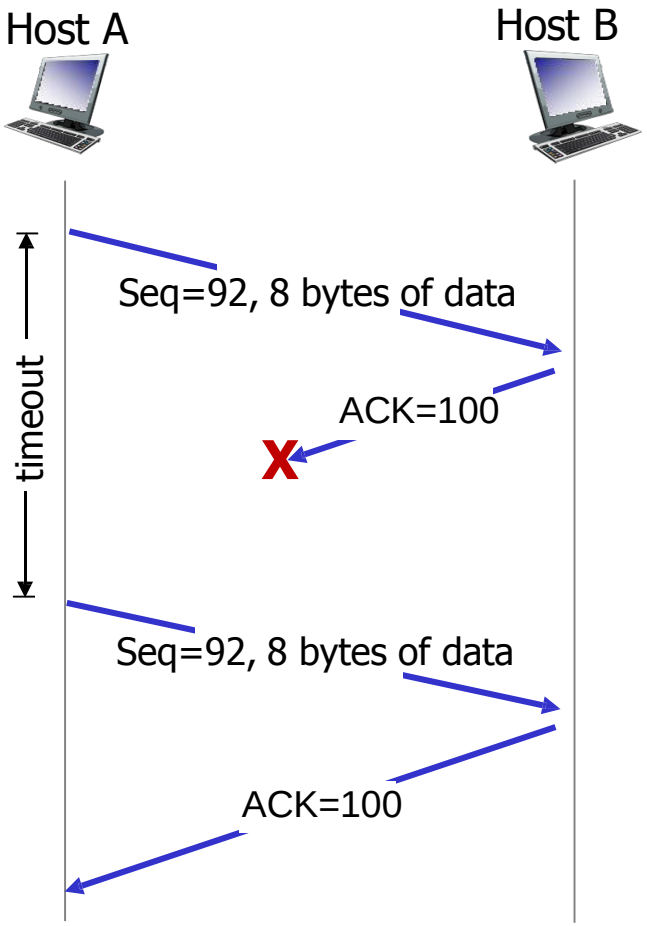
TCP Congestion Avoidance: Reno

TCP Reno Congestion avoidance

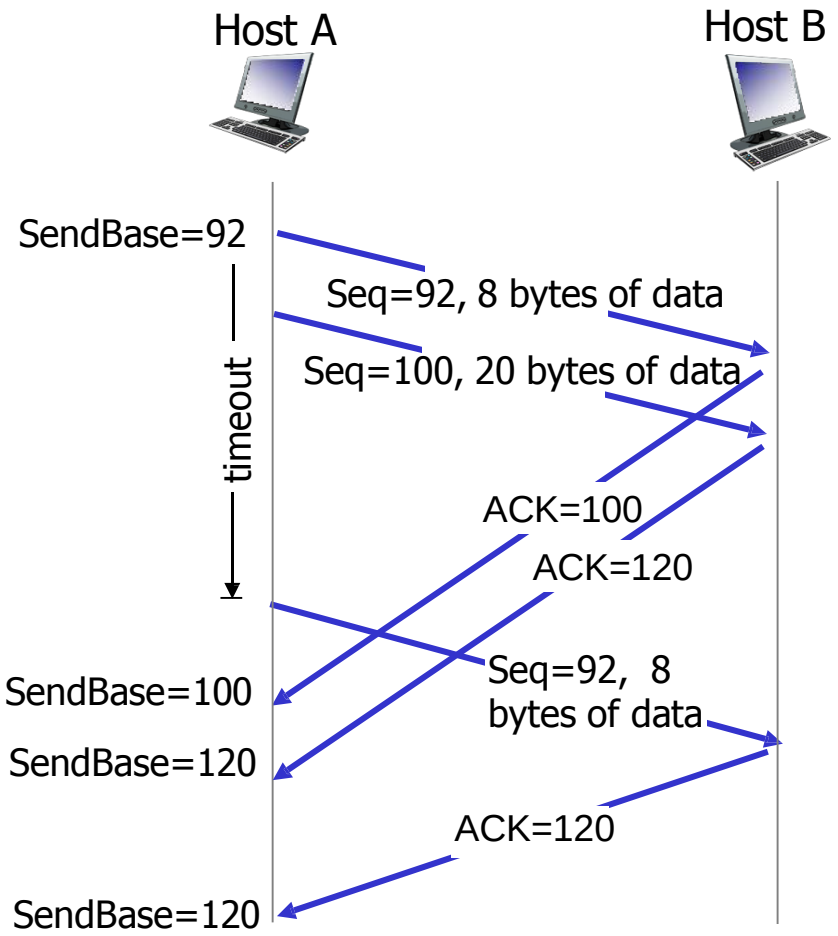
```
/* (finita la fase slowstart */
/* Congwin > threshold */
Repeat
Until (evento di perdita) {
  per ogni ACK
  Congwin = Congwin
          + 1/ Congwin
  }
Congwin = Congwin/2
```



TCP: scenari di ritrasmissione

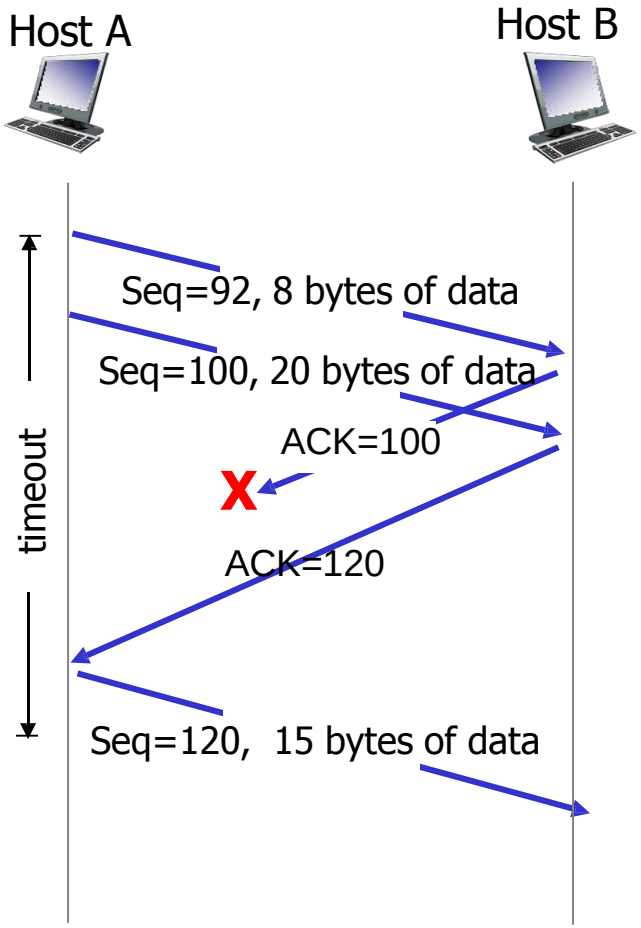


lost ACK scenario



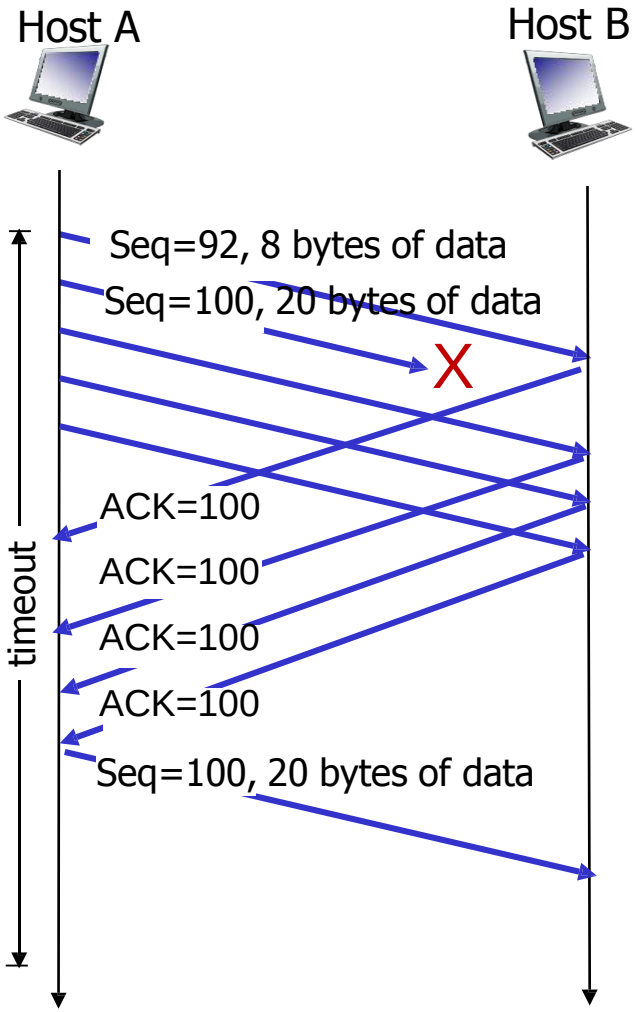
premature timeout

TCP: scenari di ritrasmissione



cumulative ACK

TCP fast retransmit



fast retransmit after sender receipt of triple duplicate ACK

Derivazione della formula $p^{-1/2}$

- Analisi di equilibrio rispetto alla finestra media E[W]
- p – probabilità di perdita
- R - round trip time medio

drift down = drift up

$$p\,E[W]/2 = (1-p) \times 1/E[W]$$

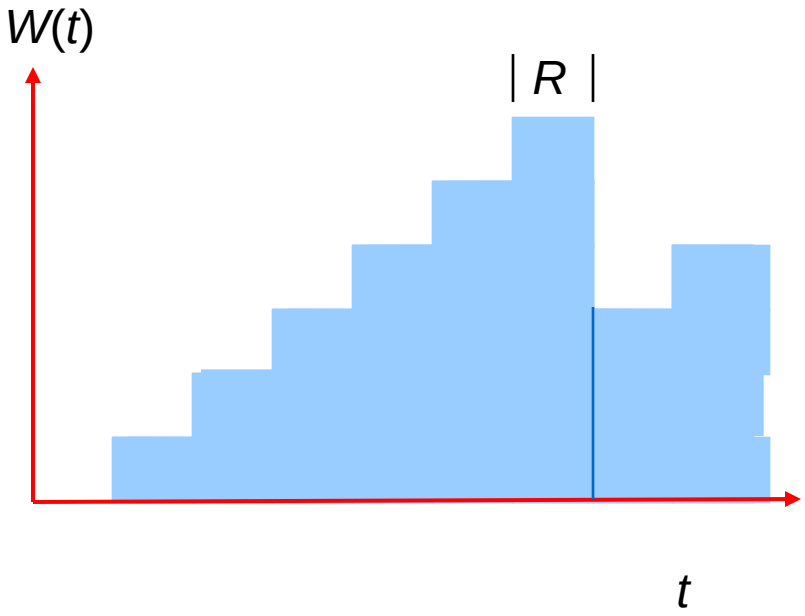
da cui:

$$E[W] = \sqrt{2(1-p)/p}$$

$$\approx \sqrt{2/p} \quad \textit{per p piccolo}$$

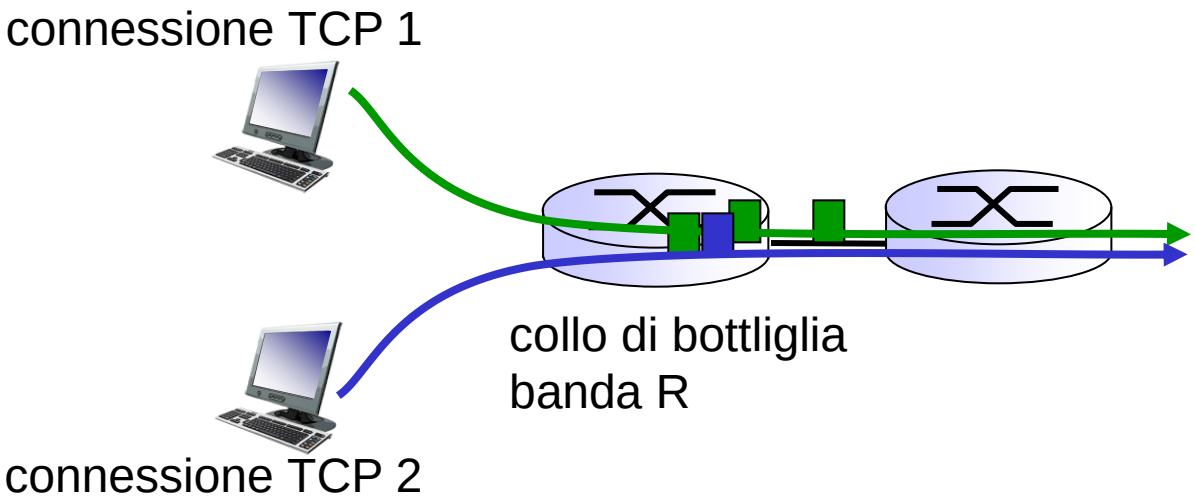
Dunque

$$B = \sqrt{2}/R\sqrt{p}$$



TCP Fairness

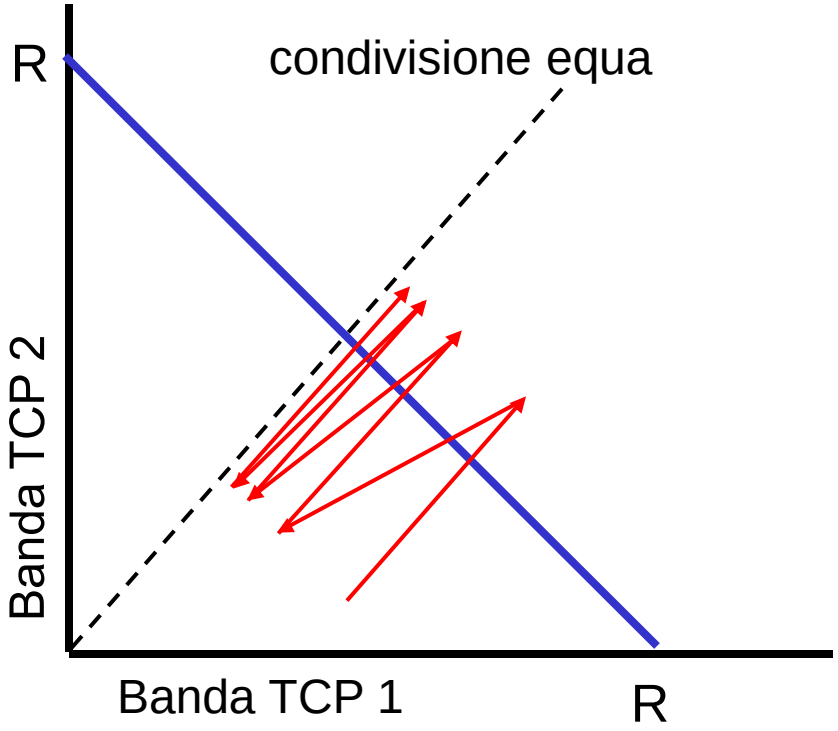
criterio di equità: se K connessioni TCP condividono lo stesso collo di bottiglia con banda totale R, ciascuna dovrebbe ottenere banda media R/K



Perchè TCP è equo?

caso di due connessioni:

- "additive increase" causa aumento lineare della banda
- "multiplicative decrease" dimezza la banda



Ripasso sulle reti

Obiettivi:

- Richiamare concetti chiave del corso introduttivo sulle reti di calcolatori
  - Rinfrescare la memoria su idee fondamentali
  - Creare una base di partenza comune
  - Identificare possibili lacune e lavoro di ripasso
  - Consolidare la terminologia

Sommario:

- Panoramica ad alto-livello
- Controllo di errore
- Controllo di flusso
- Controllo di congestione
- Indirizzamento
- Livello rete
- Livello link
- Controllo

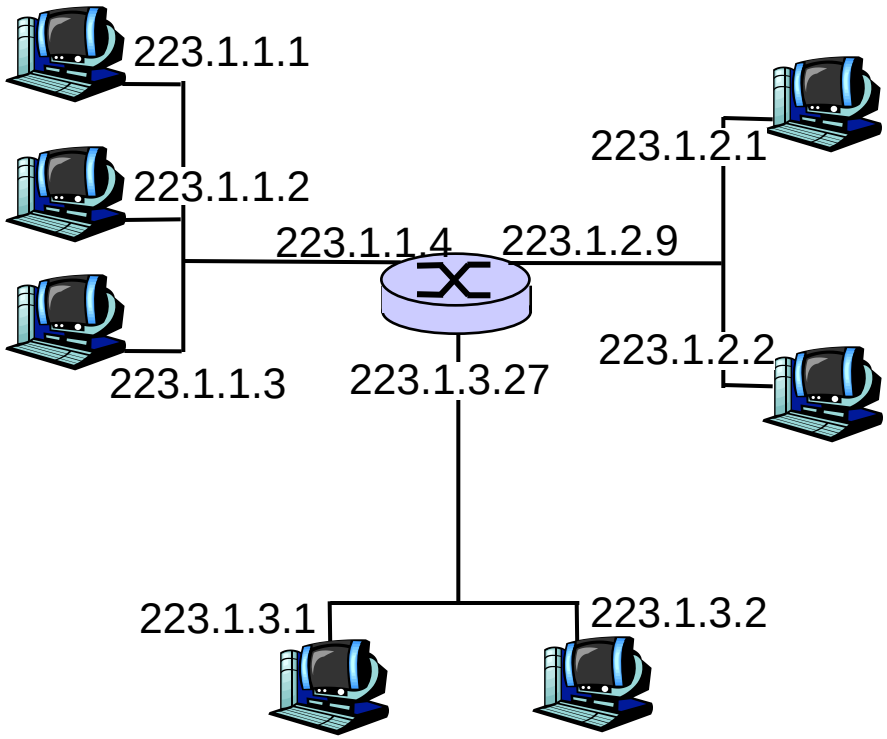
Indirizzamento

nomi vs indirizzi: domande

- per una persona fisica: che differenza c'è tra il nome e l'indirizzo?
- in Internet, qual'è il tuo *nome*, e quale è il tuo (attuale) *indirizzo*?
  - i nomi sono dipendenti dalla applicazione?
  - è possibile avere molteplici indirizzi?
- Come fa un host ad acquisire un indirizzo?

Indirizzamento a livello rete

- indirizzo IP: identificativo di 32 bit associato a un host, o all'*interfaccia* di un router
- *interfaccia:* scheda che connette un host, router a un link fisico
  - i router hanno tipicamente molte interfacce
  - gli host possono avere più di una interfaccia
  - gli indirizzi IP sono associati alle interfacce, non agli host, router

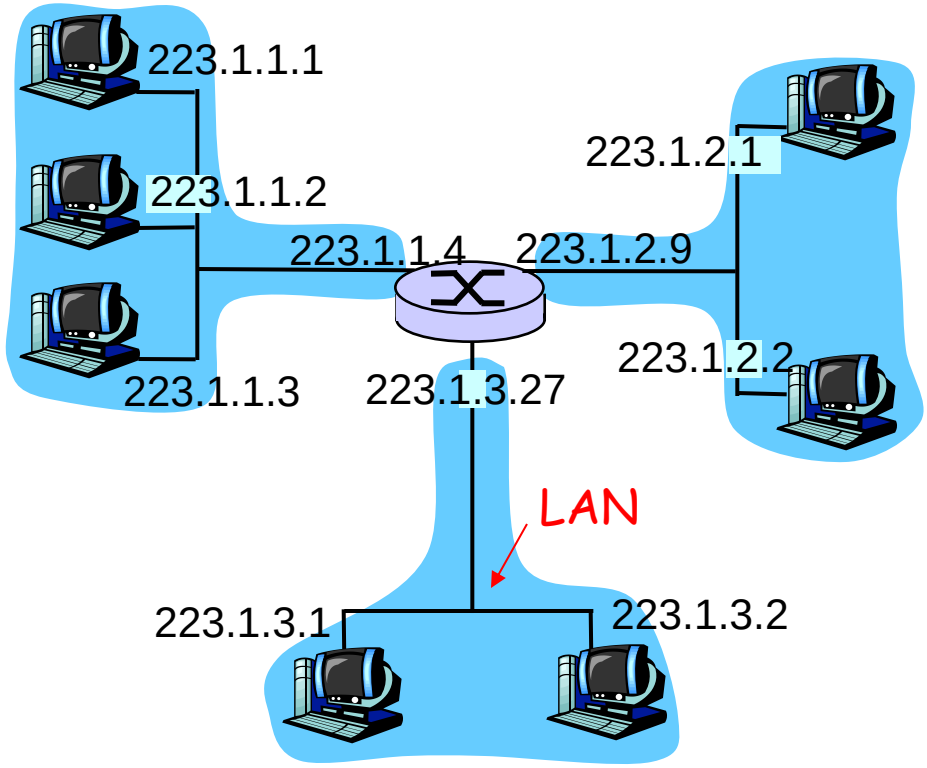


223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

Indirizzamento IP

- indirizzo IP:
  - una parte identifica la rete (bit di ordine più alto)
  - una parte identifica l'host (bit di ordine più basso)
- *che cos'è una rete ?* (dal punto di vista dell'indirizzamento IP)
  - interfacce di dispositivi con la stessa parte di rete dell'indirizzo IP
  - possono comunicare tra di loro senza l'intervento di un router

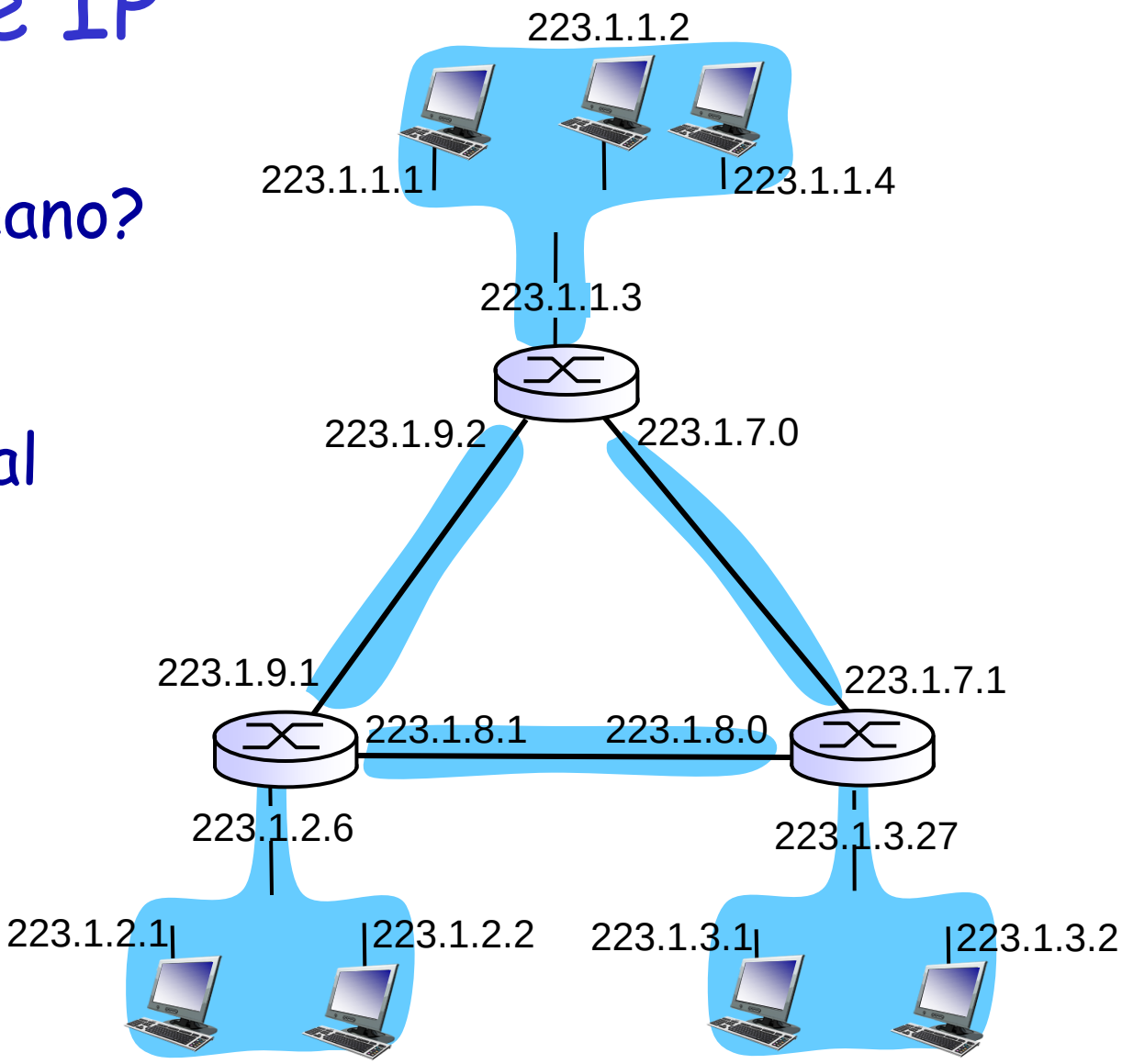


rete costituita da 3 distinte reti IP (per indirizzi IP che cominciano con 223, in cui i primi 24 bits sono la parte di rete)

Reti logiche IP

come si identificano?

immaginare di staccare l'interfaccia dal nodo



Indirizzamento IP: CIDR

CIDR: Classless InterDomain Routing

- parte rete di un indirizzo può avere lunghezza arbitraria (in numero di bits)
- formato di un indirizzo: a.b.c.d/x, dove x è il numero di bits della parte rete.
- esempio:

11001000 00010111 00010000 00000000

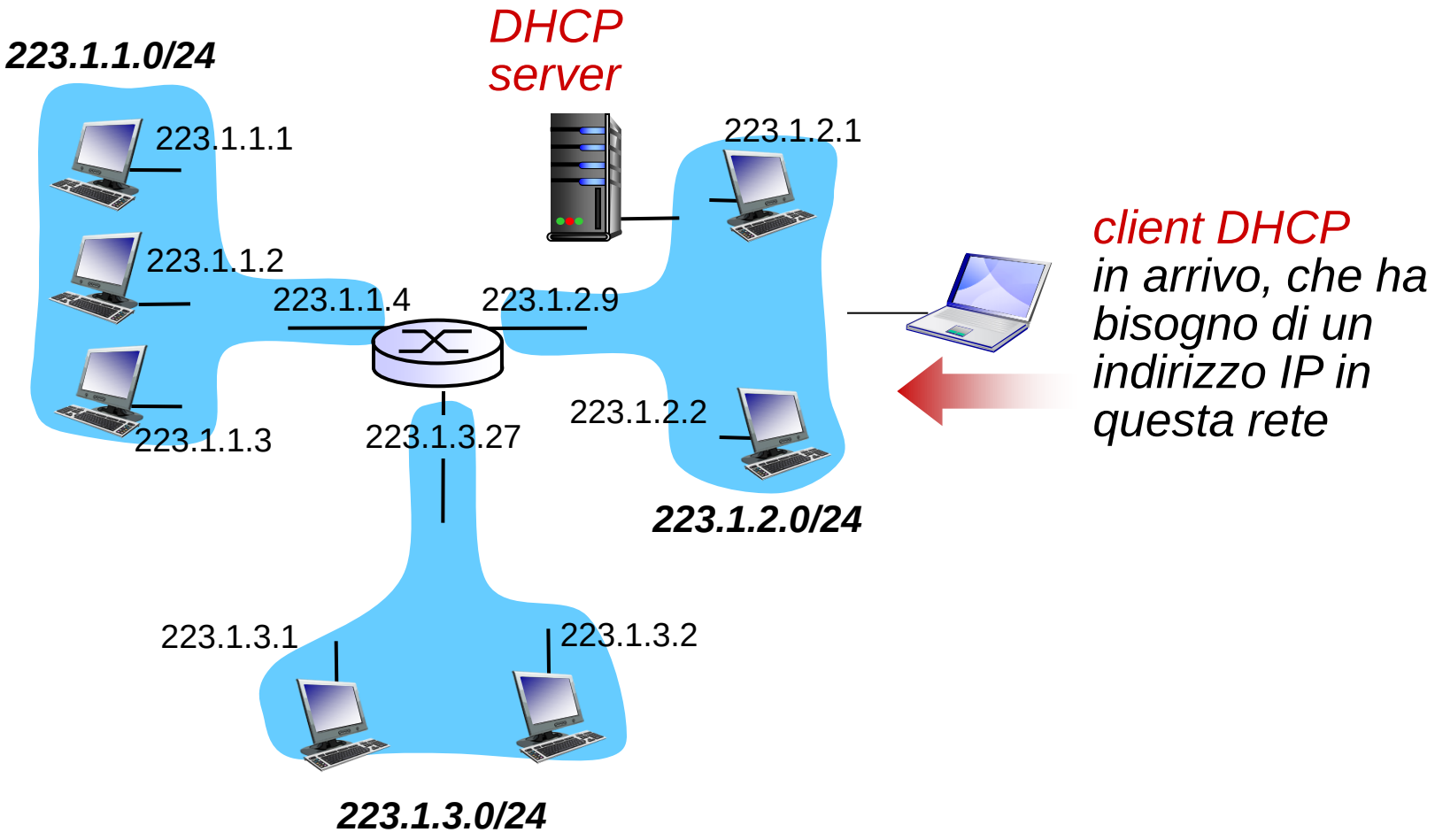
200.23.16.0/23

Indirizzi IP: come acquisirne uno?

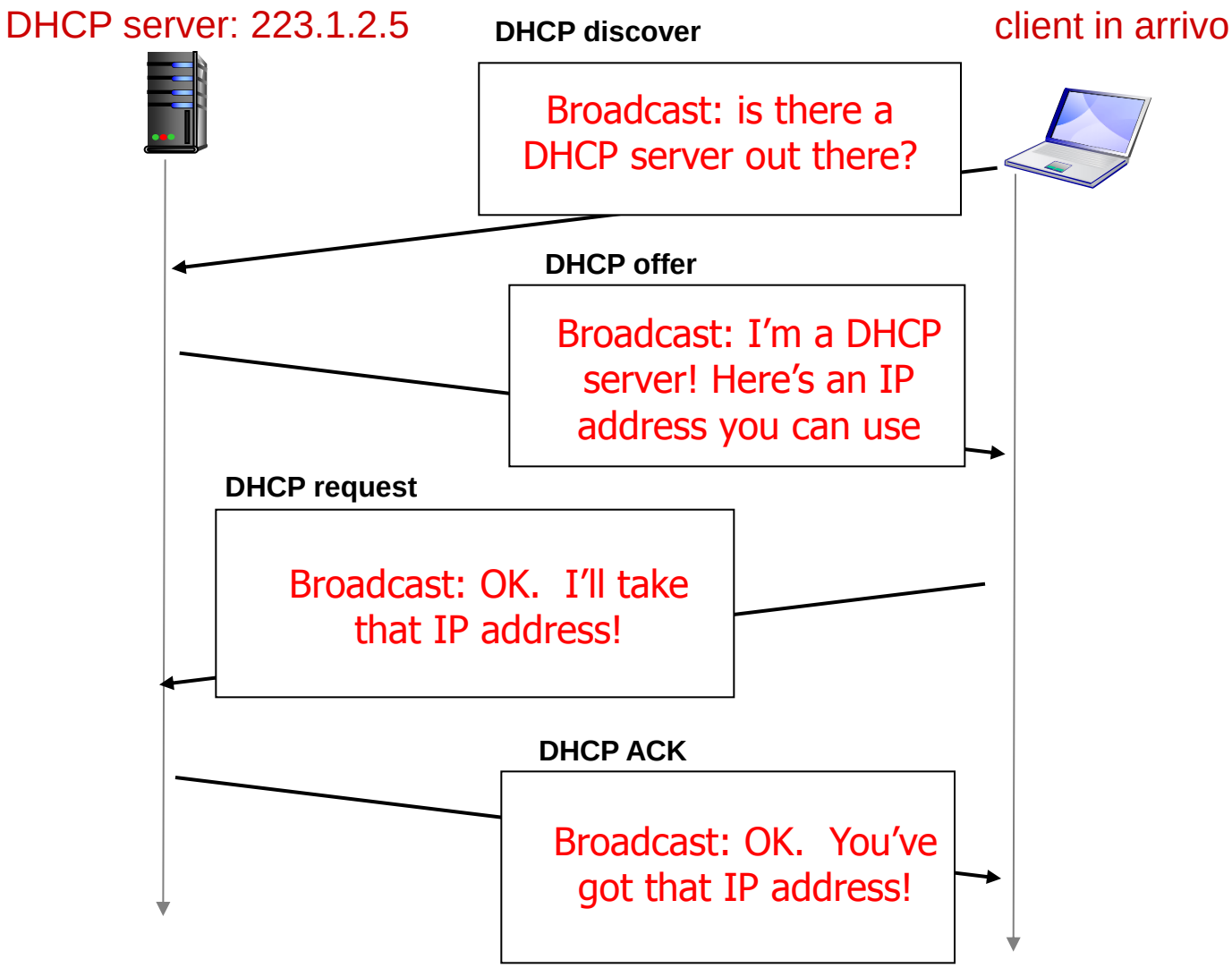
D: Come fa un host ad avere un indirizzo ?

- assegnato staticamente dall'amministratore della rete e mantenuto in memoria nell'host
  - Wintel: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config
- DHCP: Dynamic Host Configuration Protocol: indirizzo ottenuto dinamicamente: "plug-and-play"
  - host manda in broadcast "DHCP discover" msg (opz.)
  - DHCP server risponde con "DHCP offer" msg (opz.)
  - host richiede indirizzo IP "DHCP request" msg
  - DHCP server invia indirizzo: "DHCP ack" msg

Scenario DHCP client-server



Scenario DHCP client-server

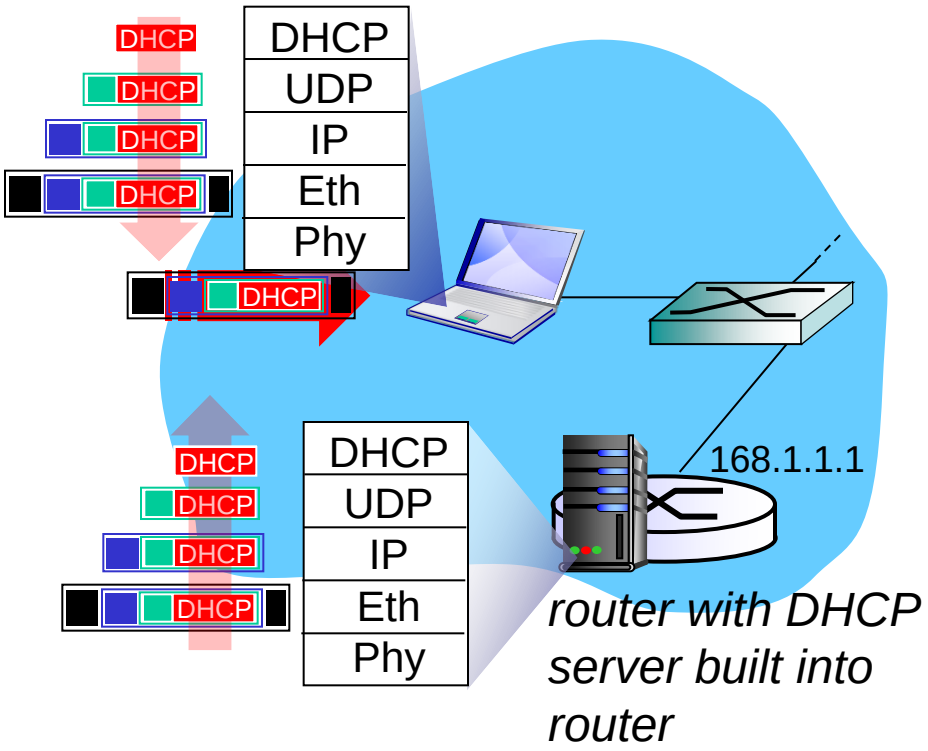


DHCP: non solo indirizzi IP

DHCP fornisce al client ulteriori informazioni:

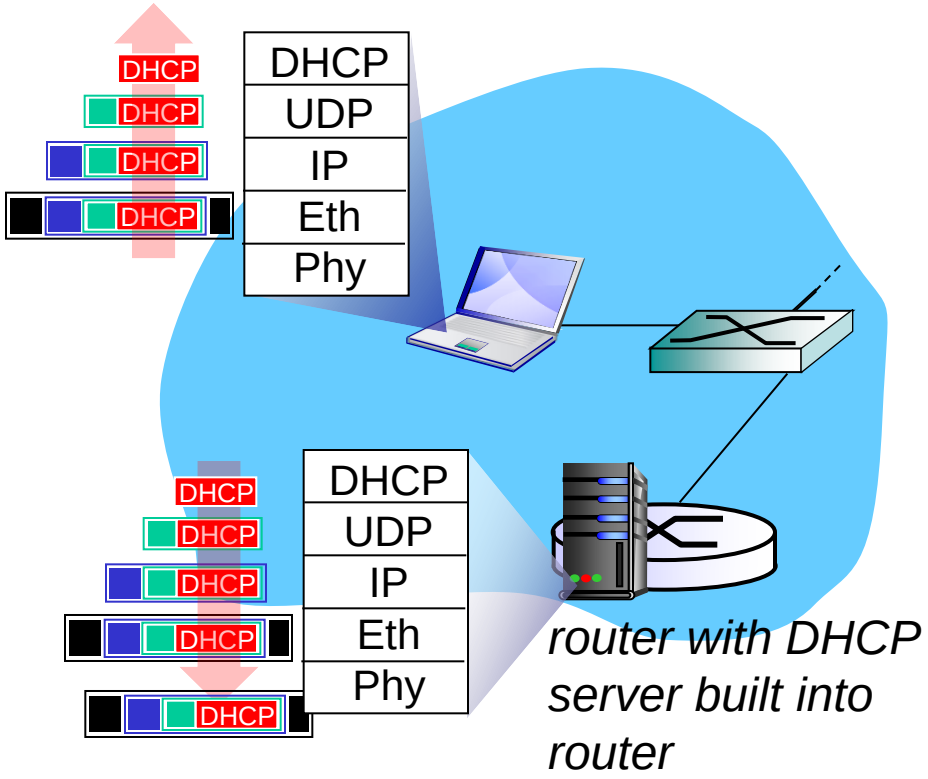
- indirizzo IP del router "first-hop"
- nome e indirizzo IP del server DNS
- maschera della rete (che distingue porzione rete e porzione host dell'indirizzo fornito)

DHCP: esempio



- connecting laptop needs its IP address, addr of first-hop router, addr of DNS server: use DHCP
- DHCP request encapsulated in UDP, encapsulated in IP, encapsulated in 802.1 Ethernet
- Ethernet frame broadcast (dest: FFFFFFFFFFFF) on LAN, received at router running DHCP server
- Ethernet demuxed to IP demuxed, UDP demuxed to DHCP

DHCP: example



- DCP server formulates DHCP ACK containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server
- encapsulation of DHCP server, frame forwarded to client, demuxing up to DHCP at client
- client now knows its IP address, name and IP address of DSN server, IP address of its first-hop router

DHCP: Wireshark output (home LAN)

request

Message type: **Boot Request (1)**
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) **DHCP Message Type = DHCP Request**
Option: (61) Client identifier
    Length: 7; Value: 010016D323688A;
    Hardware type: Ethernet
    Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Option: (t=50,l=4) Requested IP Address = 192.168.1.101
Option: (t=12,l=5) Host Name = "nomad"
Option: (55) Parameter Request List
    Length: 11; Value: 010F03062C2E2F1F21F92B
    1 = Subnet Mask; 15 = Domain Name
    3 = Router; 6 = Domain Name Server
    44 = NetBIOS over TCP/IP Name Server
    ……

reply

Message type: **Boot Reply (2)**
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 192.168.1.101 (192.168.1.101)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 192.168.1.1 (192.168.1.1)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP ACK
Option: (t=54,l=4) Server Identifier = 192.168.1.1
Option: (t=1,l=4) Subnet Mask = 255.255.255.0
Option: (t=3,l=4) Router = 192.168.1.1
Option: (6) Domain Name Server
    Length: 12; Value: 445747E2445749F244574092;
    IP Address: 68.87.71.226;
    IP Address: 68.87.73.242;
    IP Address: 68.87.64.146
Option: (t=15,l=20) Domain Name = "hsd1.ma.comcast.net."

Indirizzamento IP

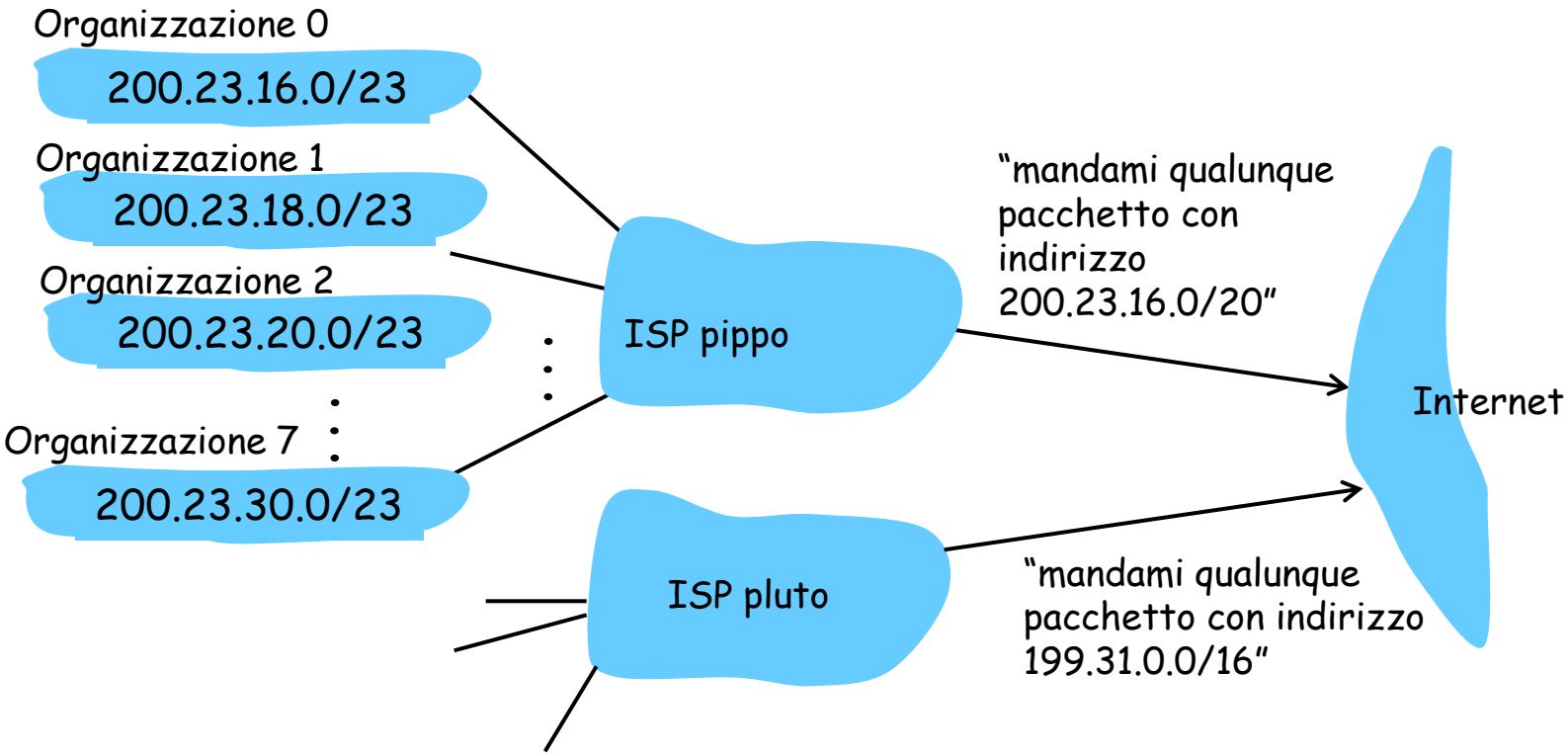
D: come fa una rete a ottenere la "parte rete" dell'indirizzo IP?

R: ottiene una porzione dello spazio IP allocato al suo ISP

allocazione ISP	11001000 00010111 00010000	00000000	200.23.16.0/20
Organizzazione 0	11001000 00010111 00010000	00000000	200.23.16.0/23
Organizzazione 1	11001000 00010111 00010010	00000000	200.23.18.0/23
Organizzazione 2	11001000 00010111 00010100	00000000	200.23.20.0/23
...	.....	....	....
Organizzazione 7	11001000 00010111 00011110	00000000	200.23.30.0/23

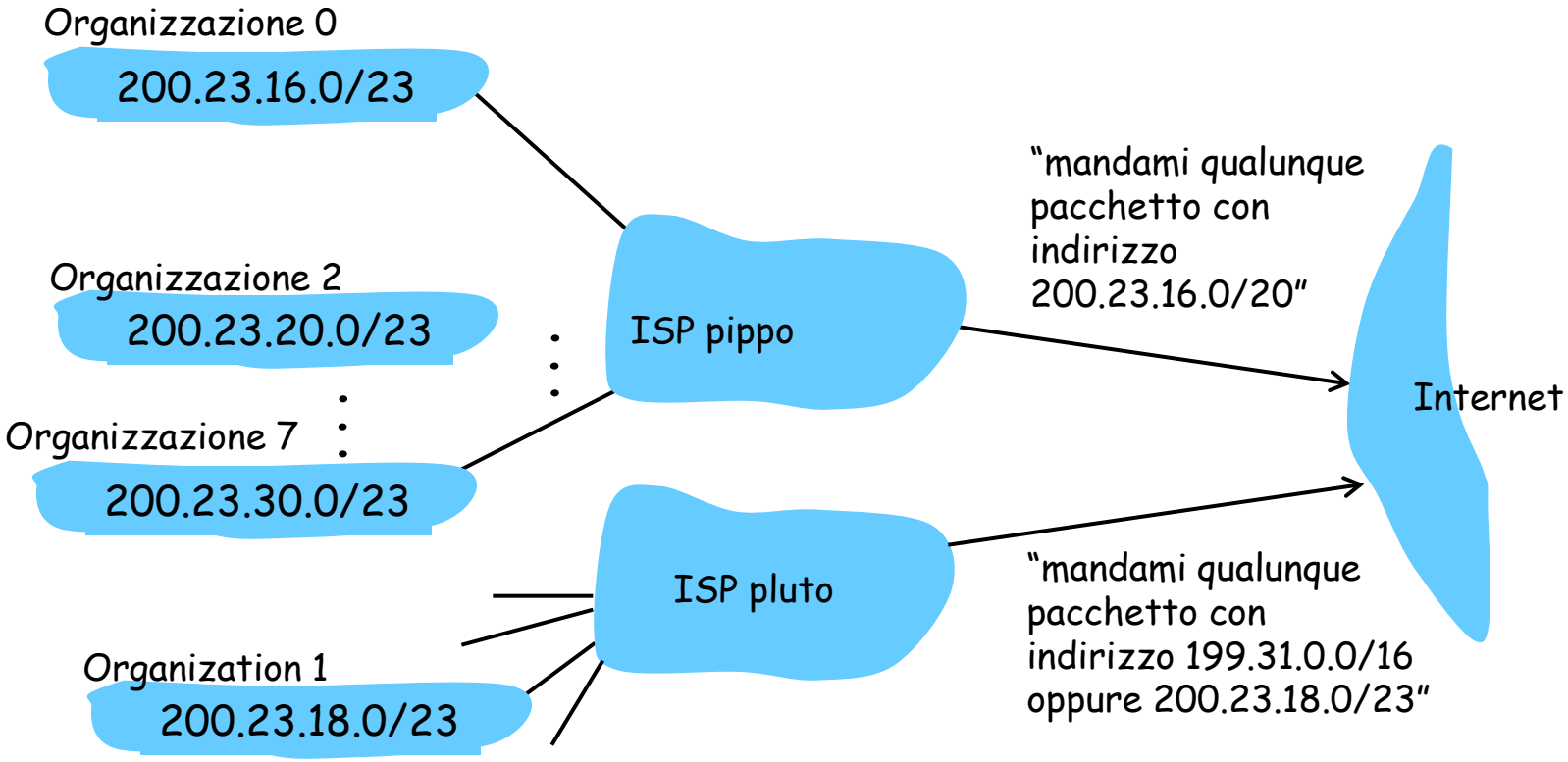
Indirizzamento gerarchico: aggregazione di percorsi

L'indirizzamento gerarchico permette una distribuzione efficiente (compatta) di informazioni di instradamento:



Indirizzamento gerarchico: regola del longest prefix match

ISP pluto ha una strada più specifica verso l'organizzazione 1



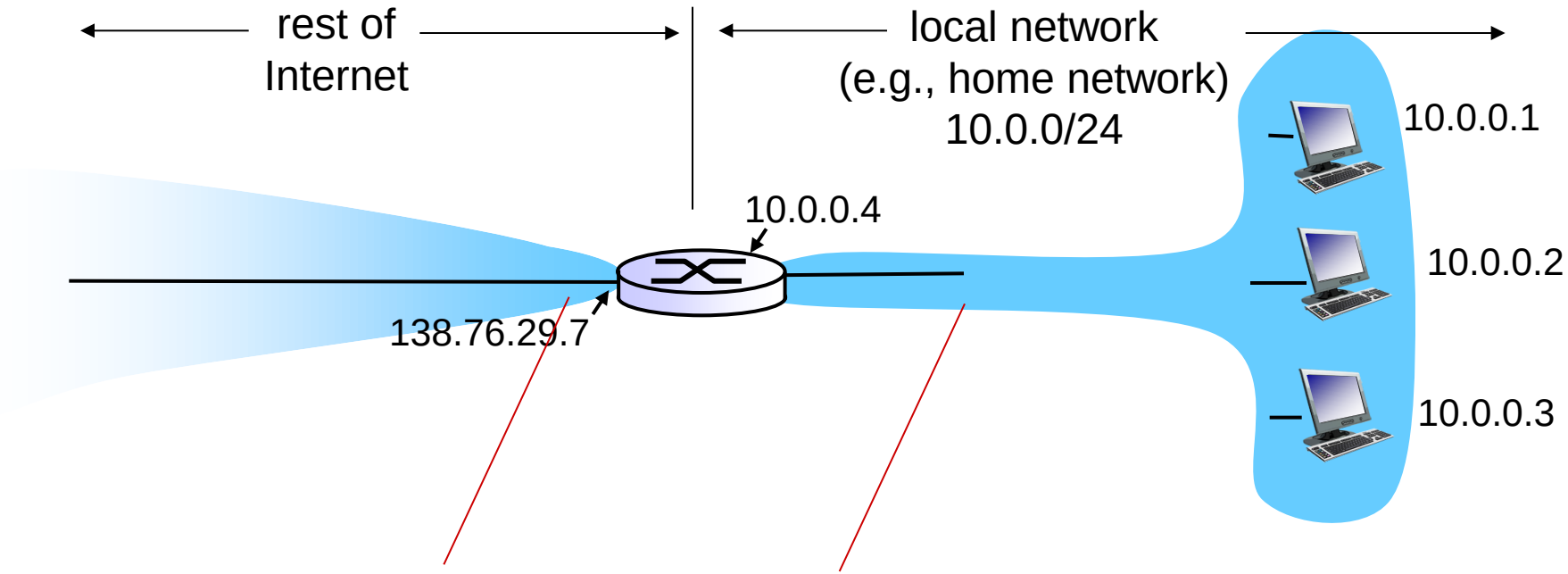
Indirizzamento IP: infine...

D: come fa un ISP a ottenere un blocco di indirizzi IP?

R: ICANN: Internet Corporation for Assigned Names and Numbers http://www.icann.org/

- alloca indirizzi
- gestisce DNS
- assegna nomi ai domini, risolve controversie

NAT: network address translation



all datagrams *leaving* local network have *same* single source NAT IP address: 138.76.29.7,different source port numbers

datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

NAT: network address translation

motivazione: una LAN usa un unico indirizzo IP per interfacciarsi con il mondo esterno:

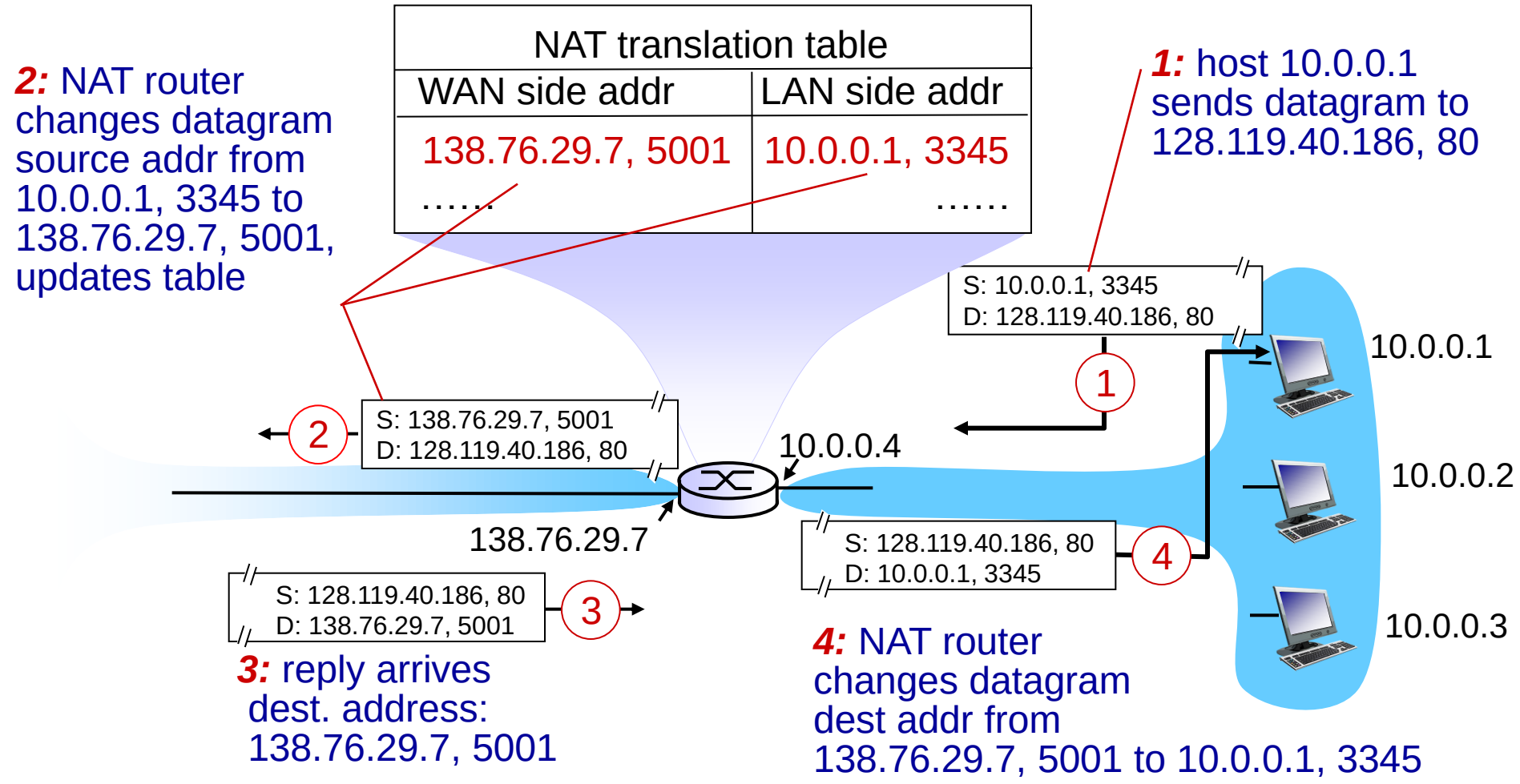
- non ha bisogno di allocare nessuno spazio condiviso dagli host: un unico indirizzo IP basta per tutti i dispositivi
- l'indirizzo interno degli host può cambiare senza bisogno di notificare il mondo esterno
- host possono cambiare gateway senza cambiare indirizzo IP privato che gli è stato assegnato
- host interni non sono direttamente raggiungibili nè visibili dal mondo esterno (sicurezza)

NAT: network address translation

implementazione: un router NAT deve fare:

- *outgoing datagrams: replace* (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
  . . . remote clients/servers will respond using (NAT IP address, new port #) as destination addr
- *remember (in NAT translation table)* every (source IP address, port #) to (NAT IP address, new port #) translation pair
- *incoming datagrams: replace* (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

NAT: network address translation



NAT: network address translation

- Il campo porta è lungo 16 bit:
  - 60,000 connessioni simultanee con un solo indirizzo IP lato-LAN!
- l'uso di NAT è controverso:
  - routers dovrebbero solo processare pacchetti fino al livello 3
  - viola il principio end-to-end di Internet
    - la possibilità di attraversare NAT deve essere tenuto in conto nello sviluppo di app, in particolare P2P
  - mancanza di indirizzi dovrebbe essere risolta con IPv6

Ripasso sulle reti

Obiettivi:

- Richiamare concetti chiave del corso introduttivo sulle reti di calcolatori
  - Rinfrescare la memoria su idee fondamentali
  - Creare una base di partenza comune
  - Identificare possibili lacune e lavoro di ripasso
  - Consolidare la terminologia

Sommario:

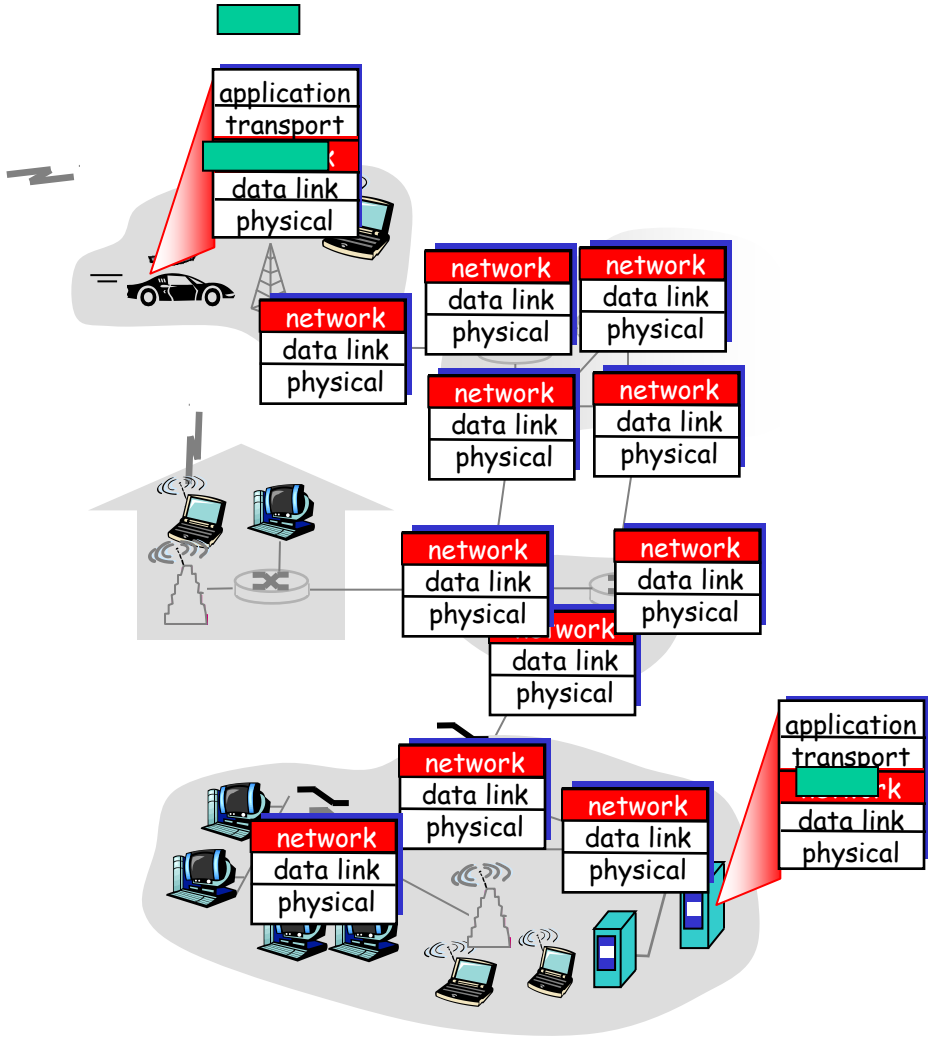
- Panoramica ad alto-livello
- Controllo di errore
- Controllo di flusso
- Controllo di congestione
- Indirizzamento
- Livello rete
- Livello link
- Controllo

Funzioni del livello di rete

- Trasportare pacchetti da host sorgente a host destinazione
- Protocolli di livello rete in *ogni* host, router

Tre funzioni importanti:

- *determinazione del cammino:* percorso seguito dai pacchetti da sorgente a destinazione. *Algoritmi di routing*
- *switching:* spostare pacchetti entro router da interfaccia di input a opportuna interfaccia di output
- *call setup:* alcune architetture di rete richiedono di impostare la connessione prima di tx dati



Modello del servizio di rete

questione CRUCIALE!

D: Qual è il *modello di servizio* per il "canale" che trasporta i datagrammi da sorgente a destinazione?

astrazione del servizio

- banda garantita?
- conservazione della distanza temporale tra i pacchetti (no jitter)?
- consegna senza perdite??
- consegna in ordine?
- feedback di congestione alla sorgente?

La più importante astrazione fornita dal livello di rete:

circuito virtuale
o
datagramma?

Esempi di modelli di servizio a livello rete

Network Architecture	Service Model	Guarantees ?				Congestion feedback
		Bandwidth	Loss	Order	Timing	
Internet	best effort	none	no	no	no	no (inferred via loss)
ATM	CBR	constant rate	yes	yes	yes	no congestion
ATM	VBR	guaranteed rate	yes	yes	yes	no congestion
ATM	ABR	guaranteed minimum	no	yes	no	yes
ATM	UBR	none	no	yes	no	no

Circuiti virtuali

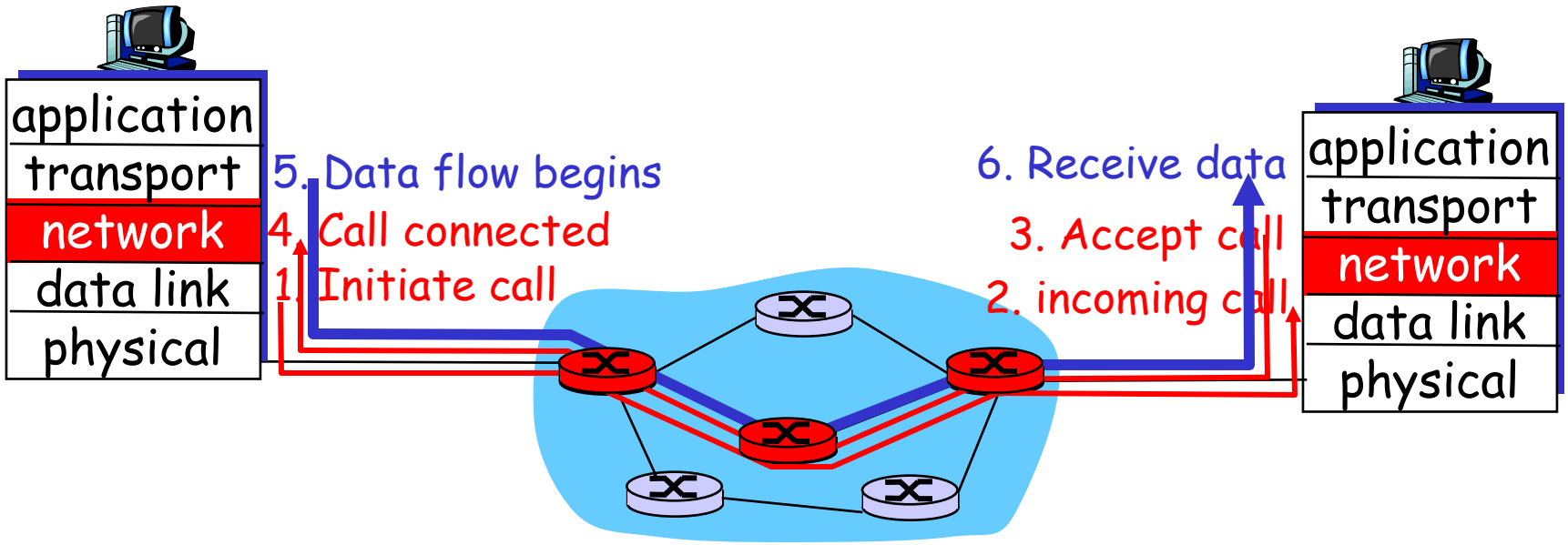
"percorsi sorgente-destinazione che si comportano come i circuiti telefonici tradizionali"

- orientati alle prestazioni
- richiedono azioni nella rete lungo il percorso sorgente-destinazione

- instaurazione e chiusura della connessione, per ogni comunicazione, in aggiunta al trasferimento di dati
- ogni pacchetto porta un identificativo VC (*non* l'ID dell'host destinazione)
- *ogni* router sul percorso sorgente-destinazione mantiene lo "stato" di ciascuna connessione che lo attraversa
  - la connessione a livello trasporto coinvolge solo i sistemi terminali
- le risorse dei link, router (banda, buffers) possono essere *allocate* ai VC
  - per ottenere prestazioni tipo circuito dedicato

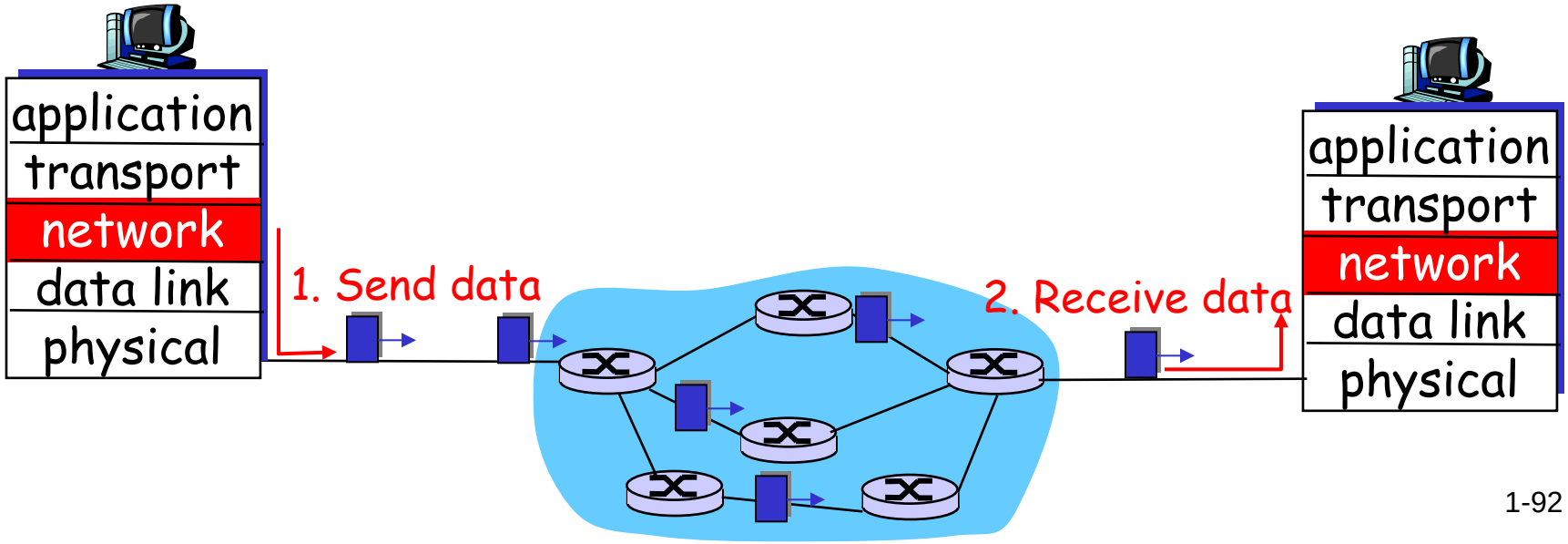
Circuiti virtuali: protocolli di segnalazione

- usati per stabilire, mantenere, abbattere un VC
- usato in ATM, frame-relay, X.25
- non usato nella Internet attuale



Reti a datagramma (architettura di Internet)

- Non c'è impostazione della chiamata a livello di rete
- router: non mantengono stato delle connessioni end-to-end
  - non esiste concetto di "connessione" a livello di rete
- pacchetti tipicamente instradati usando l'ID dell'host destinazione
  - pacchetti tra la stessa coppia sorgente-destinazione possono prendere percorsi diversi



Perché reti a circuito virtuale o a datagramma?

Internet

- Necessità di scambiare dati tra differenti calcolatori.
  - Servizi elastici, non vi sono eccessivi requisiti di tempo
- L'interconnessione è semplice (computer)
  - È adattabile, effettua controlli e recupera errori
  - Rete interna non complessa, la complessità sta agli estremi
- Svariati tipi di link
  - Caratteristiche differenti
  - Difficile uniformarne il servizio

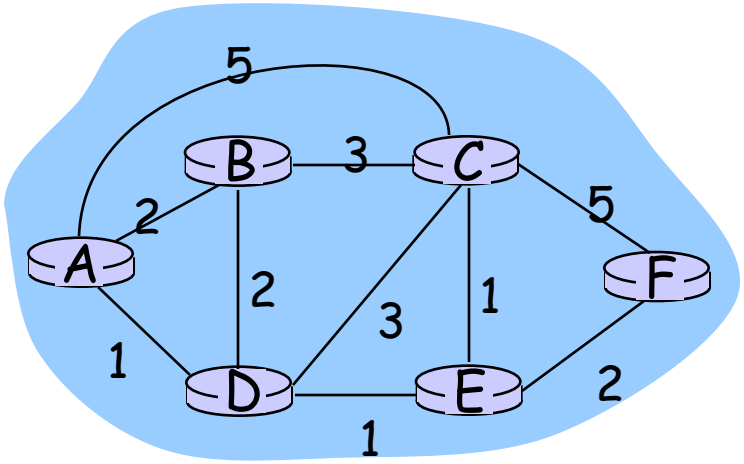
ATM

- Deriva dal mondo della telefonia.
- Conversazione telefonica:
  - Requisiti stringenti in termini di tempo e affidabilità.
  - Necessità di servizi garantiti.
- Sistemi terminali "stupidi"
  - Telefoni.
  - La complessità sta nella rete interna.

Instradamento (routing)

Protocollo di routing

Obiettivo: determinare un "buon" percorso (sequenza di router) attraverso la rete dalla sorgente alla destinazione



Astrazione sotto forma di grafo per algoritmi di instradamento:

- i nodi del grafo sono router
- gli archi del grafo sono i canali fisici
  - peso degli archi: ritardo, costo monetario, livello di congestione

- "buon" percorso:
  - tipicamente significa cammino di costo minimo
  - altre definizioni sono possibili

Routing: i due approcci usati nella pratica

Globale:

- tutti i router conoscono la topologia completa della rete (inclusi i pesi dei link)
- algoritmi "link state": usano l'algoritmo di Dijkstra per trovare i cammini più a costo minimo da un certo router a tutte le destinazioni

Decentralizzati:

- un router conosce solo i vicini direttamente collegati a livello fisico, e il costo per raggiungerli
- processo computazionale iterativo, tramite scambio di informazione tra nodi vicini
- algoritmi "distance vector"

Algoritmi di istradamento "Distance Vector"

iterativi:

- continuano finchè termina scambio di info tra i nodi
- *auto-terminanti*: no "segnale" di stop

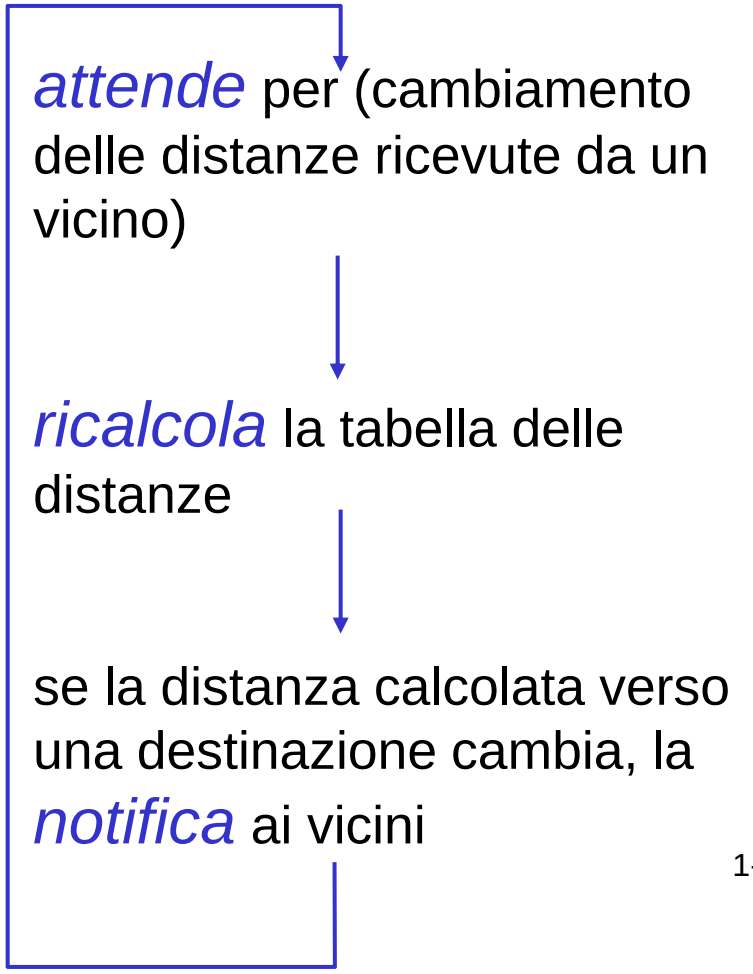
asincroni:

- non occorre che i nodi sincronizzino lo scambio di info tra loro!

distribuiti:

- ogni nodo comunica *solo* coi vicini direttamente collegati

Ogni nodo:



Instradamento gerarchico

La nostra visione finora è molto idealizzata:

- tutti i router supposti identici
- rete "piatta"

… *non* è così in pratica

scala: con un miliardo di destinazioni:

- non si possono mantenere tutte le destinazioni nelle tabelle di instradamento!
- Lo scambio delle tabelle intaserebbe i link di traffico!

autonomia amministrativa

- Internet = rete di reti
- ogni amministratore di rete può voler controllare l'instradamento dentro la sua rete

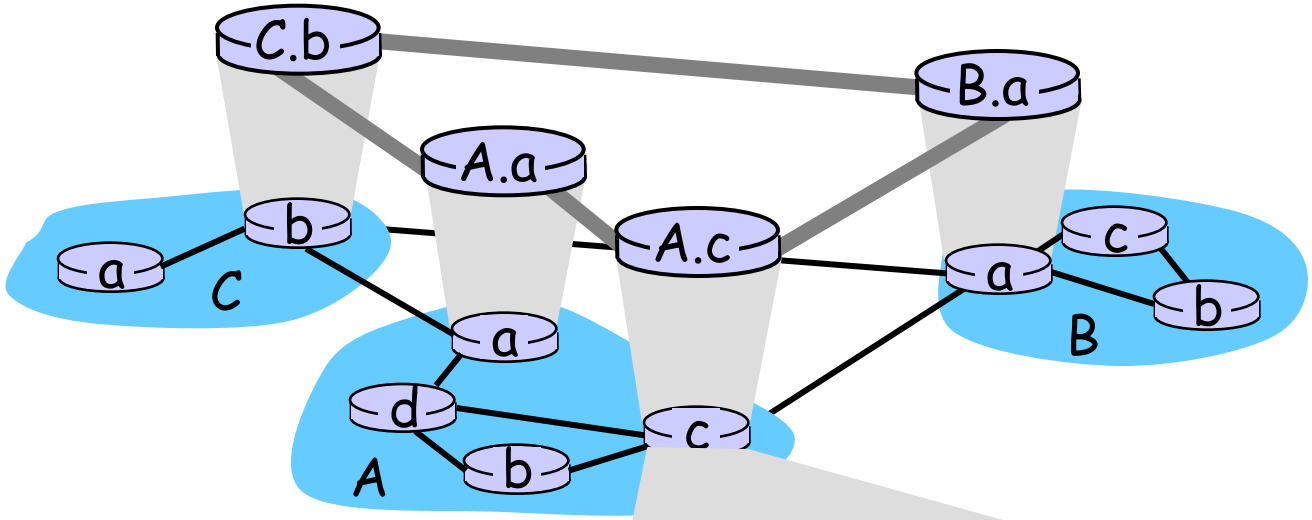
Instradamento gerarchico

- router organizzati in *sistemi autonomi (AS, autonomous system).*
- I router di uno stesso AS eseguono lo stesso algoritmo d'instradamento.
  - Protocollo d'instradamento interno al sistema autonomo (intra-AS).
  - router in diversi AS possono eseguire diversi protocolli di routing intra-AS

router gateway

- router speciali di un AS
- eseguono il protocollo di routing intra-AS alla pari degli altri router dell'AS
- *inoltre* sono responsabili dell'instradamneto verso destinazioni esterne alll'AS
  - eseguono un protocollo di routing *inter-AS* con altri gateway router

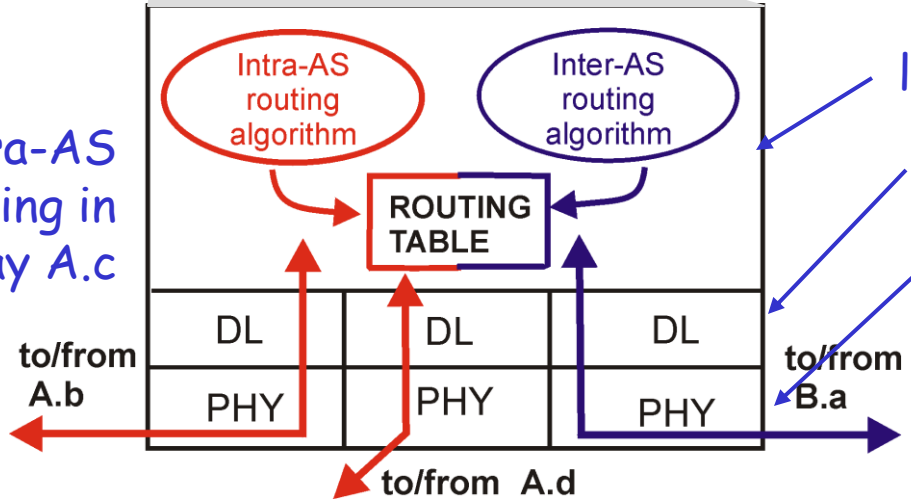
Routing Intra-AS e Inter-AS



Gateways:

- realizzano tra di loro l'instradamento inter-AS
- realizzano l'instradamento intra-AS con gli altri router del loro AS

inter-AS, intra-AS routing in gateway A.c

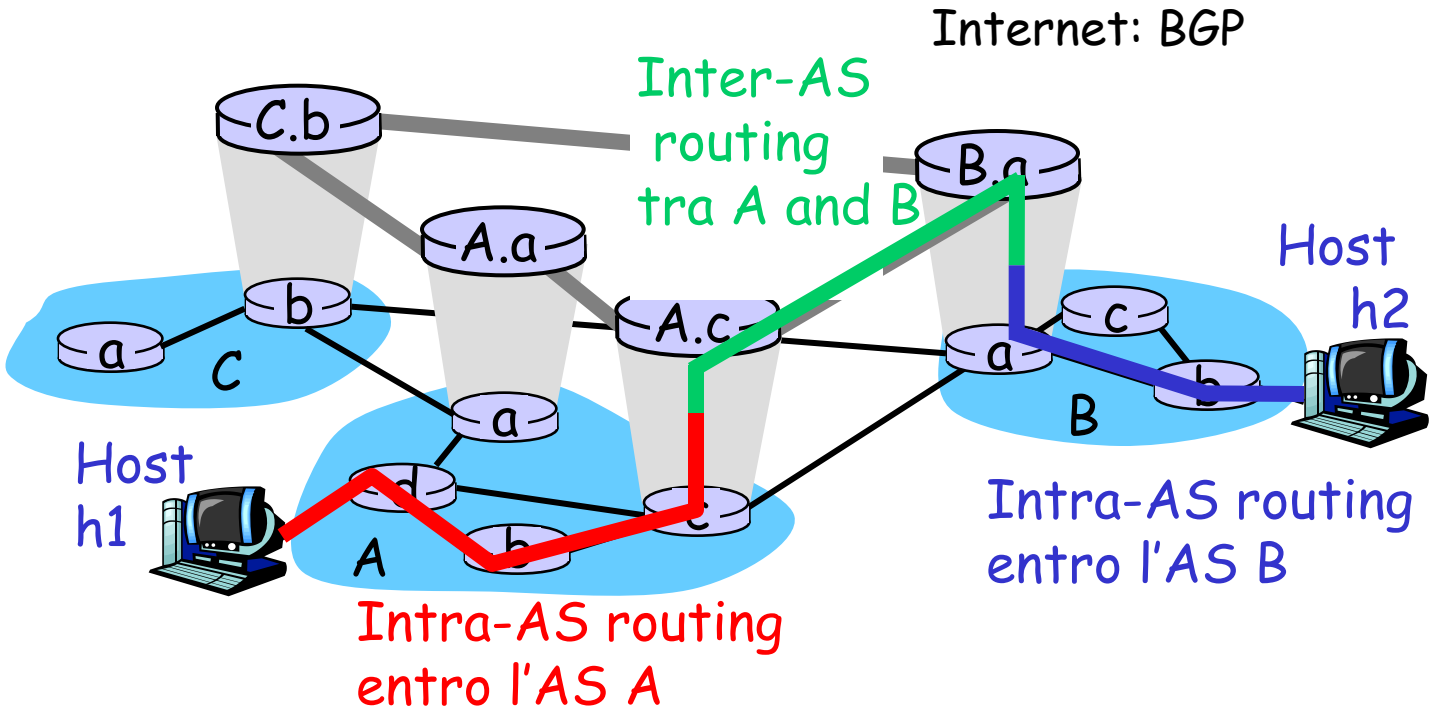


livello rete

livello link

livello fisico

Routing Intra-AS e Inter-AS



Ripasso sulle reti

Obiettivi:

- Richiamare concetti chiave del corso introduttivo sulle reti di calcolatori
  - Rinfrescare la memoria su idee fondamentali
  - Creare una base di partenza comune
  - Identificare possibili lacune e lavoro di ripasso
  - Consolidare la terminologia

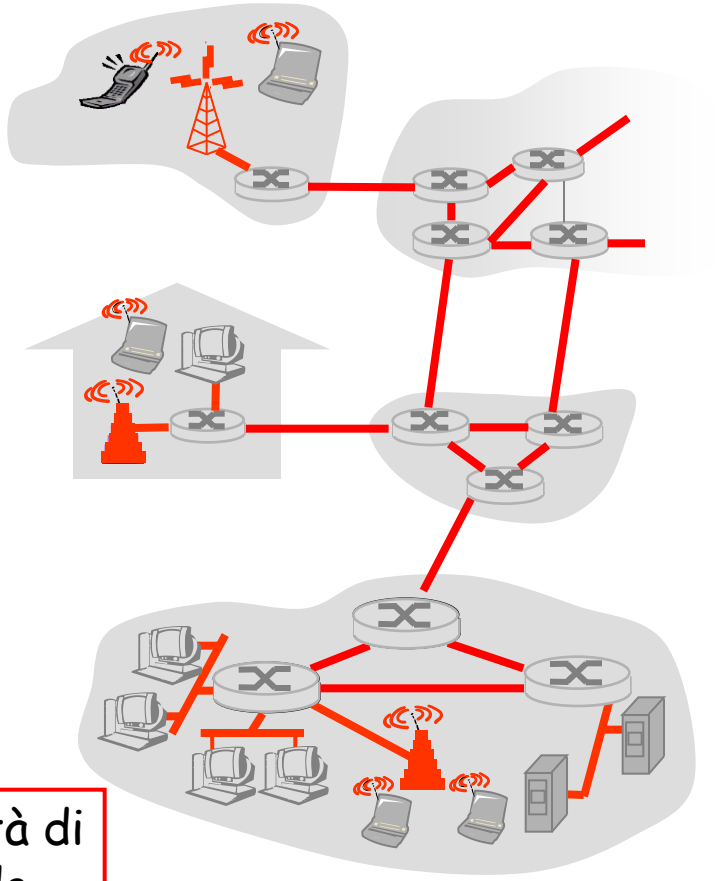
Sommario:

- Panoramica ad alto-livello
- Controllo di errore
- Controllo di flusso
- Controllo di congestione
- Indirizzamento
- Livello rete
- Livello link
- Controllo

Livello Link: Introduzione

Terminologia:

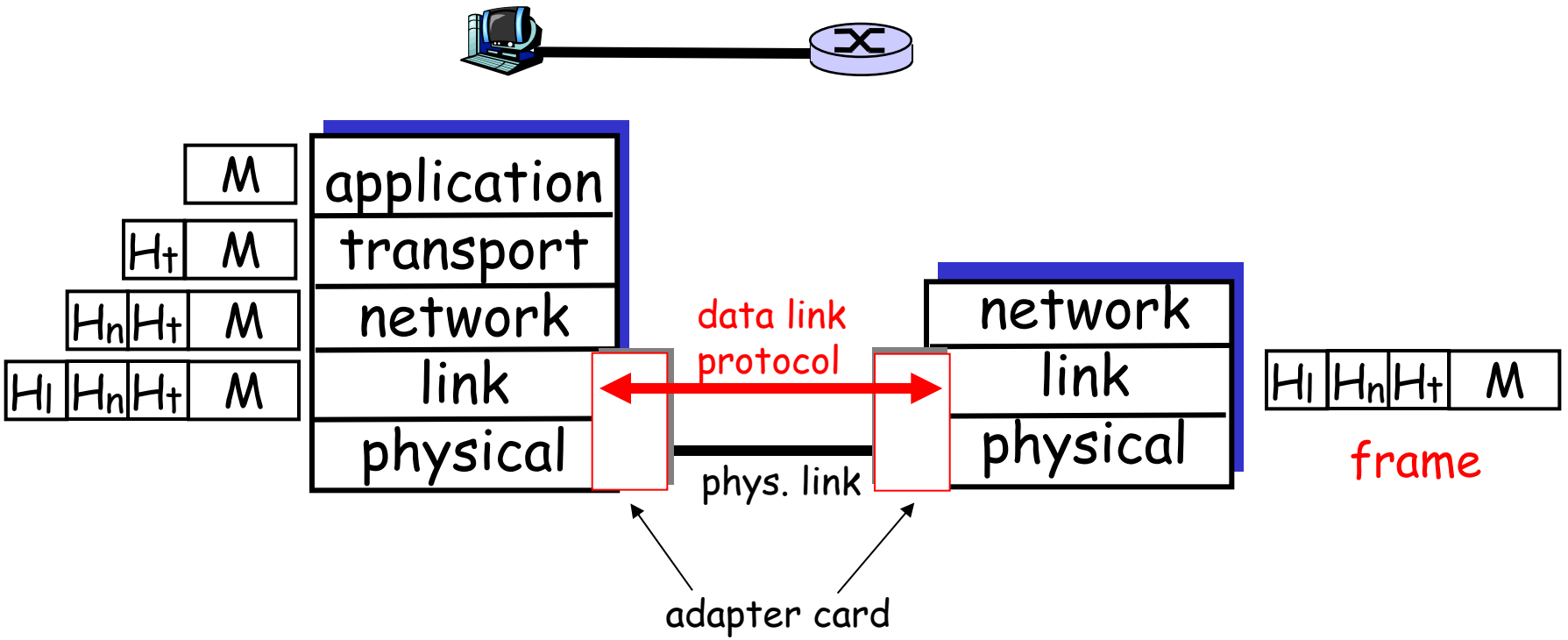
- host e routers sono **nodes**
- canali di comunicazione che connettono nodi adiacenti sono **links**
  - link cablati
  - link wireless
  - LANs
- i pacchetti di livello 2 si chiamano **frame** (incapsulano un datagramma di livello 3)



Il livello data-link layer ha la responsabilità di transferire datagrammi da un nodo a un nodo adiacente su un link

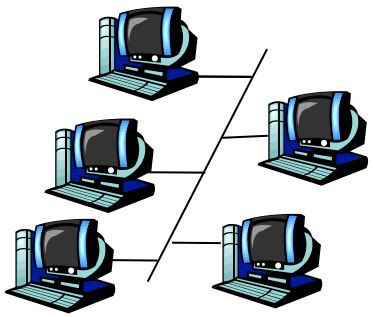
Livello Link: il contesto

- due dispositivi *fisicamente collegati*:
  - host-router, router-router, host-host
- unità di dati: *frame*

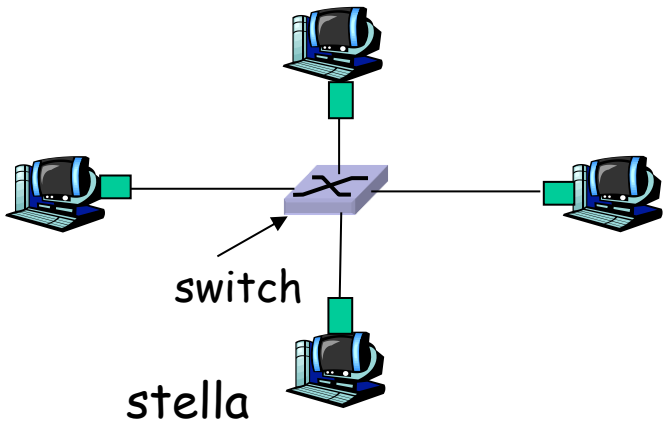


LANs

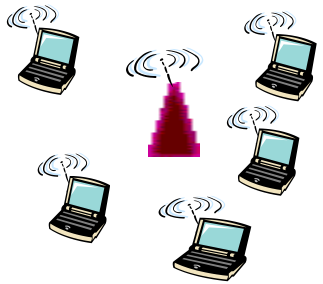
- topologia a bus, popolare fino alla metà degli anni 90
- oggi: prevale la topologia a stella
  - *switch* in mezzo, in cui ogni interfaccia esegue (separatamente) il protocollo di accesso Ethernet
- wireless LANS: 802.11



bus: cavo coassiale



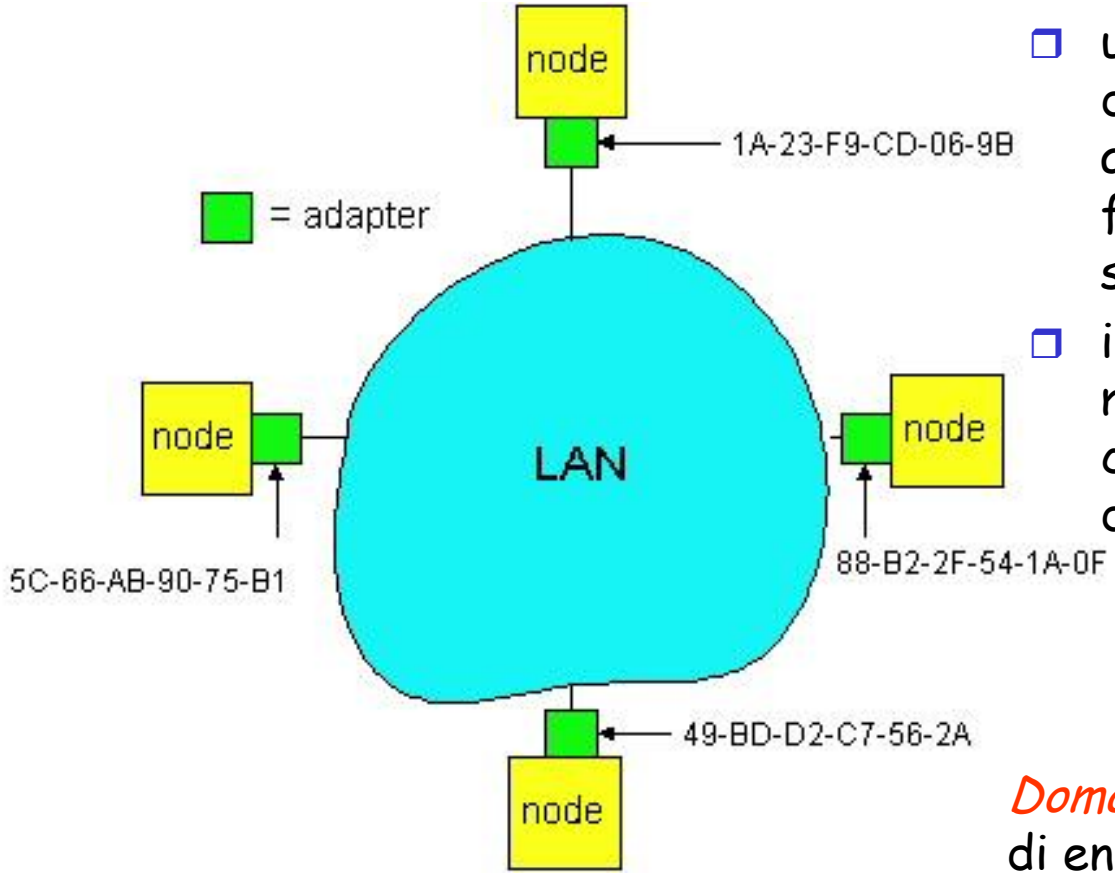
stella



canale radio condiviso
(e.s.: 802.11 WiFi)

indirizzi di LAN

Ogni adattatore LAN ha un indirizzo MAC univoco



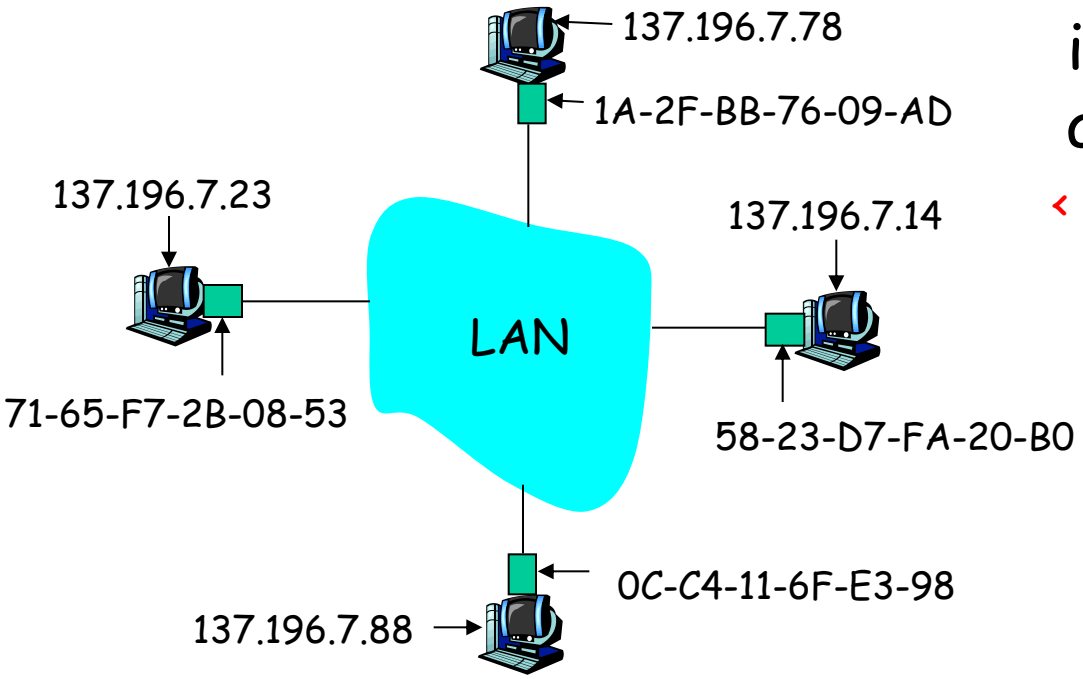
Indirizzo LAN (o MAC o fisico):

- usato per trasferire datagrammi da una interfaccia a un'altra interfaccia fisicamente connessa (sulla stessa rete)
- indirizzo MAC a 48 bit (nella maggior parte dei casi) codificato sulla ROM dell'adattatore

Domanda: che bisogno c'è di entrambi gli indirizzi MAC e IP?

ARP: Address Resolution Protocol

Domanda: come determinare l'indirizzo MAC di B a partire dall'indirizzo IP di B?



- ogni nodo IP (host, router) su una LAN mantiene una tabella ARP
- tabella ARP: mappa indirizzi IP/MAC per alcuni nodi della LAN

  < IP address; MAC address; TTL>

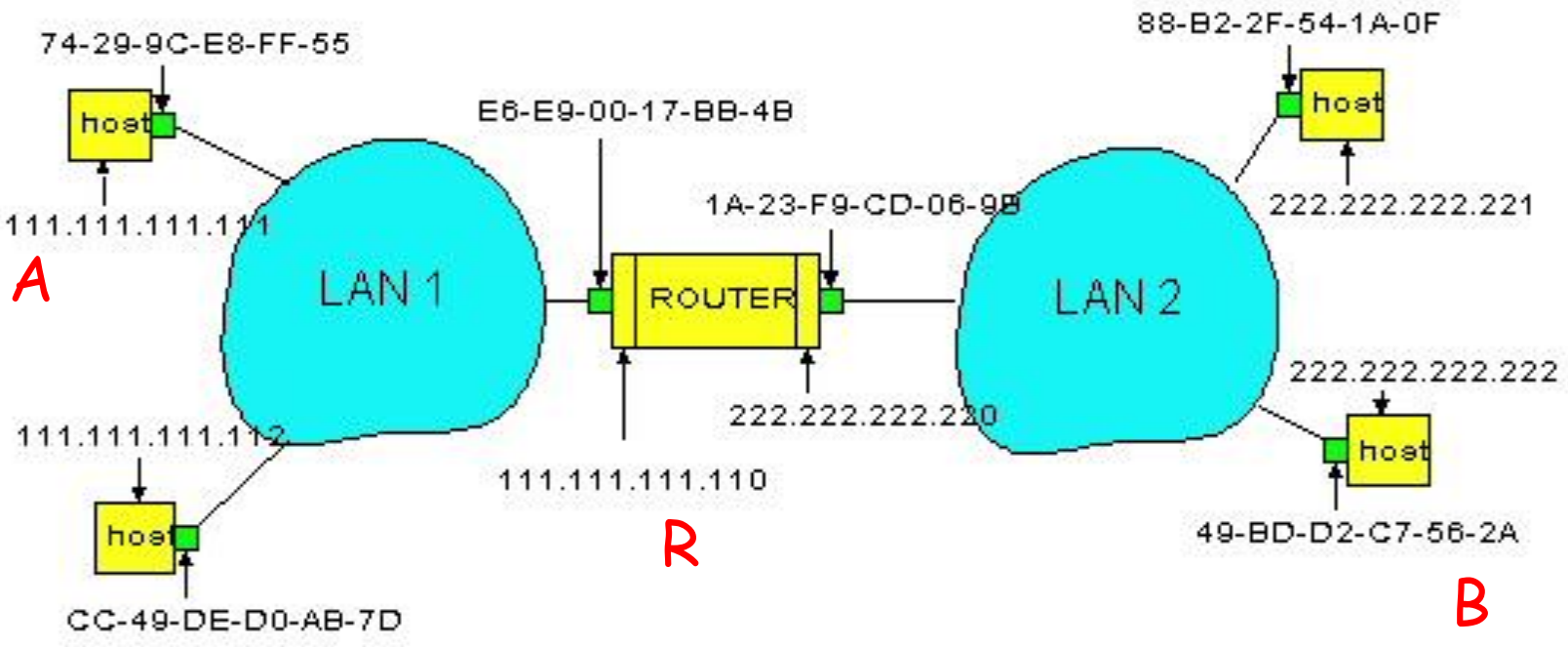
  - TTL (Time To Live): tempo dopo il quale il mappaggio viene dimenticato (tipicamente 20 min)

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.

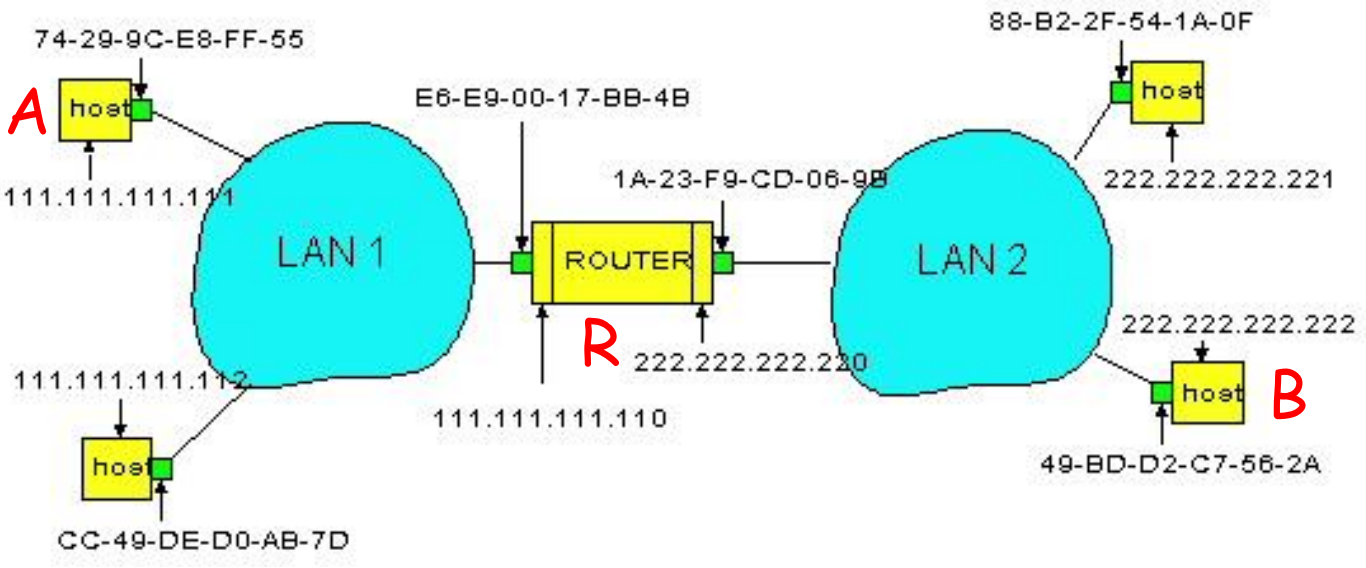
Invio verso un nodo esterno alla sottorete

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).

- A crea un datagramma con origine A, e destinazione B.
- A usa ARP per ottenere l'indirizzo MAC di R.
- A crea un collegamento a livello di rete con l'indirizzo MAC di destinazione di R, il frame contiene il datagramma IP da A a B.
- L'adattatore di A invia il datagramma.
- L'adattatore di R riceve il datagramma.
- R rimuove il datagramma IP dal frame Ethernet, e vede che la sua destinazione è B.
- R usa ARP per ottenere l'indirizzo MAC di B.
- R crea un frame contenente il datagramma IP da A a B IP e lo invia a B.

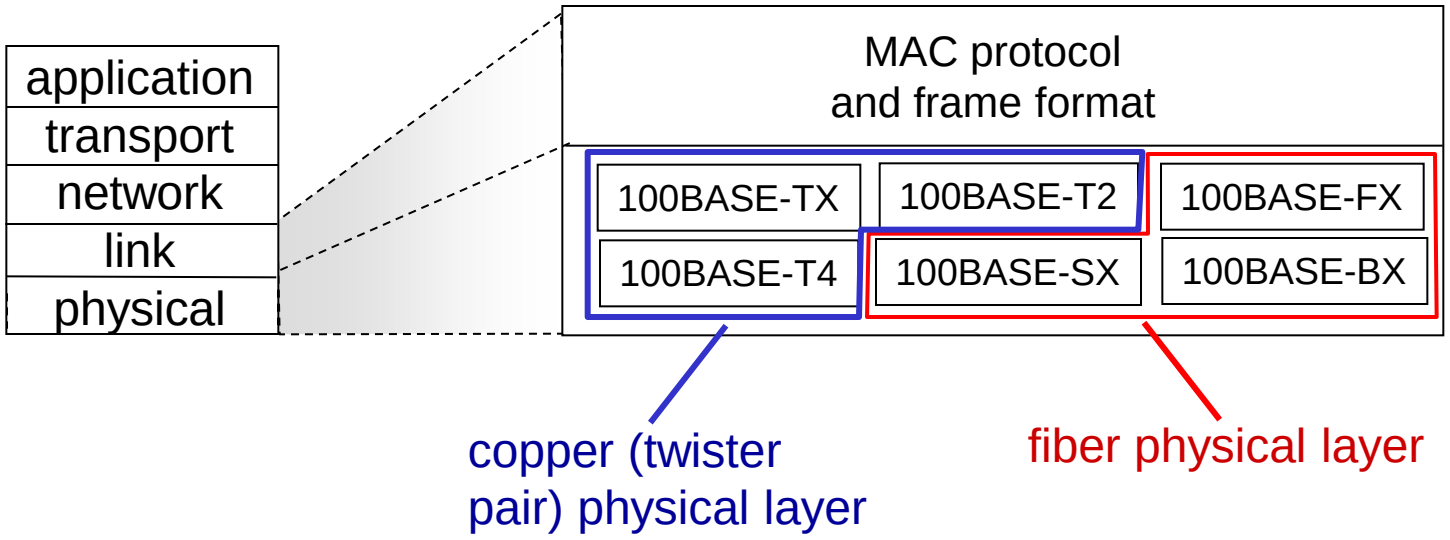


Ethernet: connectionless, non affidabile

- *connectionless:* no handshaking tra NIC mittente e NIC ricevente
- *non affidabile:* NIC ricevente non invia acks o nack al NIC mittente
  - dati in frame scartati sono recuperati solo se nodo mittente recupera errori a livello superiore (es: TCP)
- Protocollo MAC di Ethernet: *CSMA/CD non slottizzato con backoff binario esponenziale*

802.3 Ethernet standards: livelli link e fisico

- *molti* diversi standard Ethernet
  - comune formato del protocollo e dei frame
  - differenti velocità: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10G bps
  - differenti mezzi fisici di transmissione: fibra, cavo, doppino



Ethernet switch

- link-layer device: takes an *active* role
  - store, forward Ethernet frames
  - examine incoming frame's MAC address, selectively forward frame to one-or-more outgoing links when frame is to be forwarded on segment, uses CSMA/CD to access segment
- *transparent*
  - hosts are unaware of presence of switches
- *plug-and-play, self-learning*
  - switches do not need to be configured

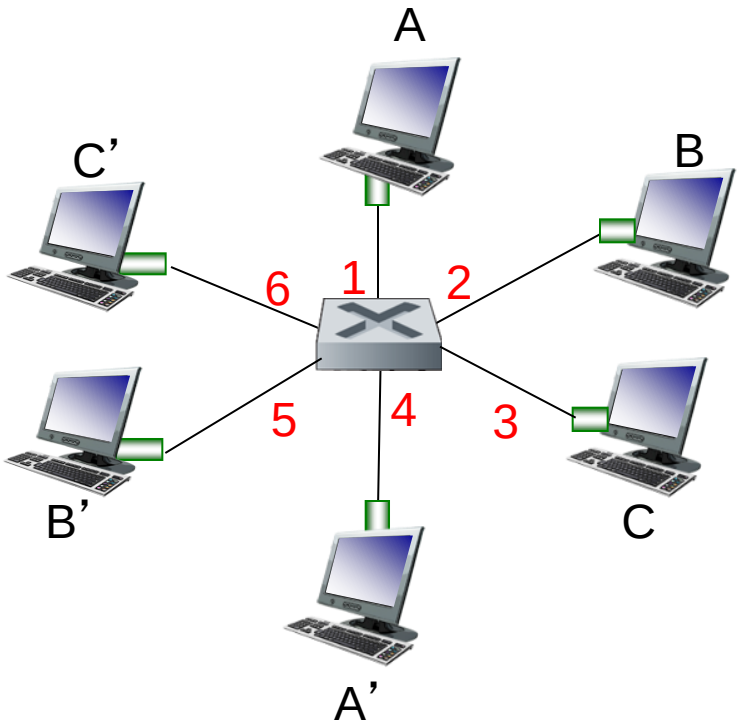
Switch forwarding table

Q: how does switch know A' reachable via interface 4, B' reachable via interface 5?

- *A: each switch has a switch table, each entry:*
  - *(MAC address of host, interface to reach host, time stamp)*
  - *looks like a routing table!*

Q: how are entries created, maintained in switch table?

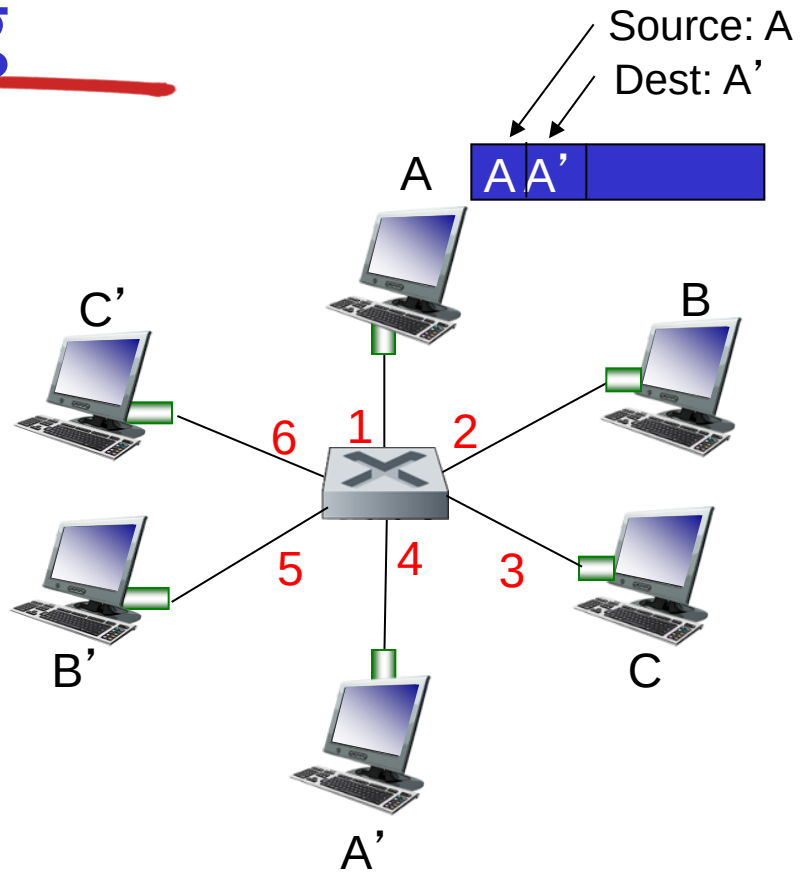
- *something like a routing protocol?*



switch with six interfaces (1,2,3,4,5,6)

Switch: self-learning

- switch *learns* which hosts can be reached through which interfaces
  - when frame received, switch "learns" location of sender: incoming LAN segment
  - records sender/location pair in switch table

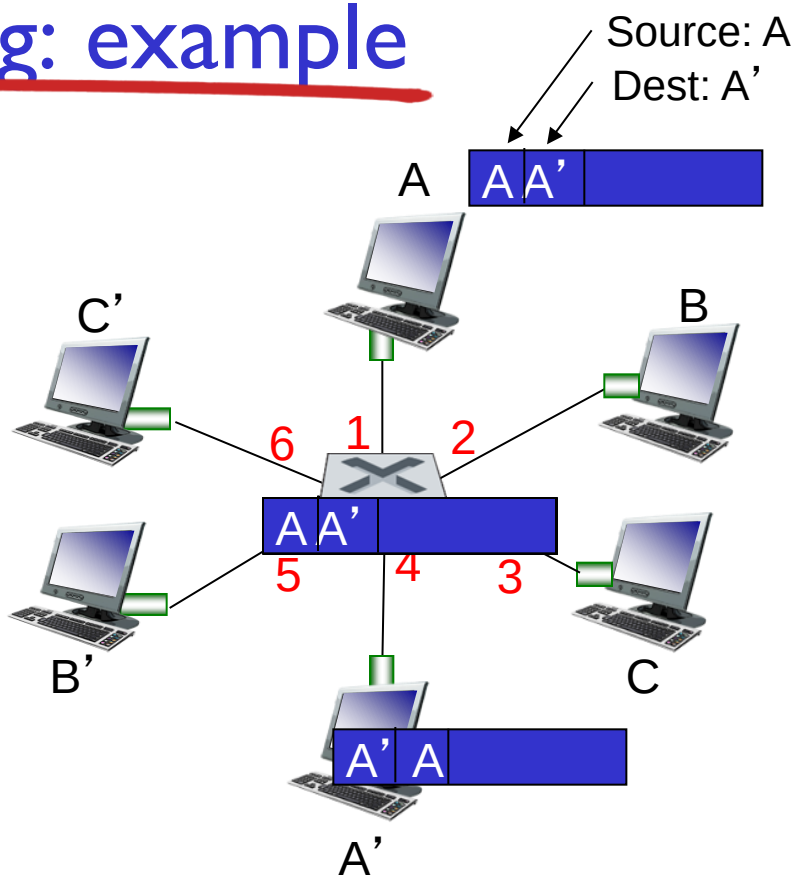


MAC addr	interface	TTL
A	*1*	*60*

Switch table (initially empty)

Self-learning, forwarding: example

- frame destination, A', location unknown: *flood*
- destination A location known: *selectively send on just one link*

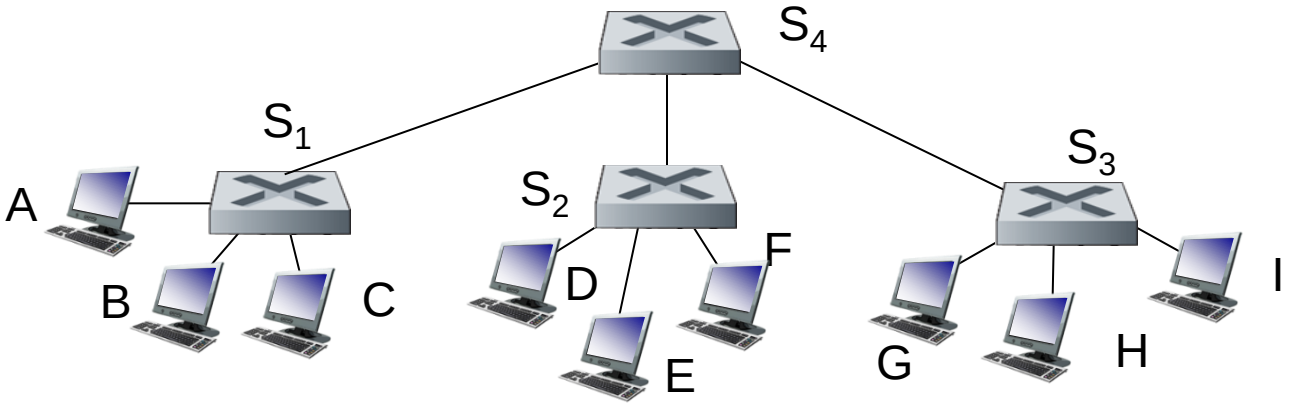


MAC addr	interface	TTL
A	*1*	*60*
A'	*4*	*60*

switch table (initially empty)

Interconnecting switches

- switches can be connected together

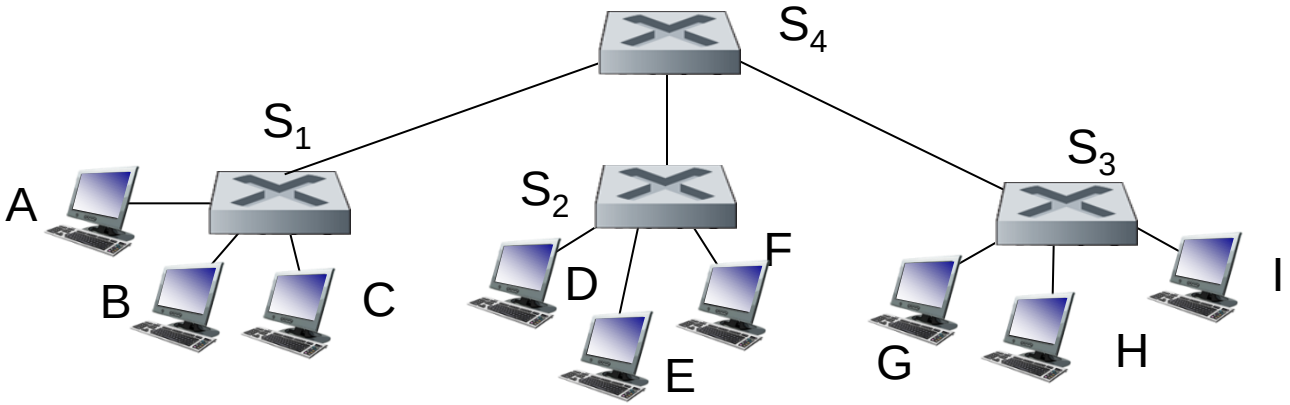


Q: sending from A to G - how does S_1 know to forward frame destined to G via S_4 and S_3?

- A: self learning! (works *exactly* the same as in single-switch case!)

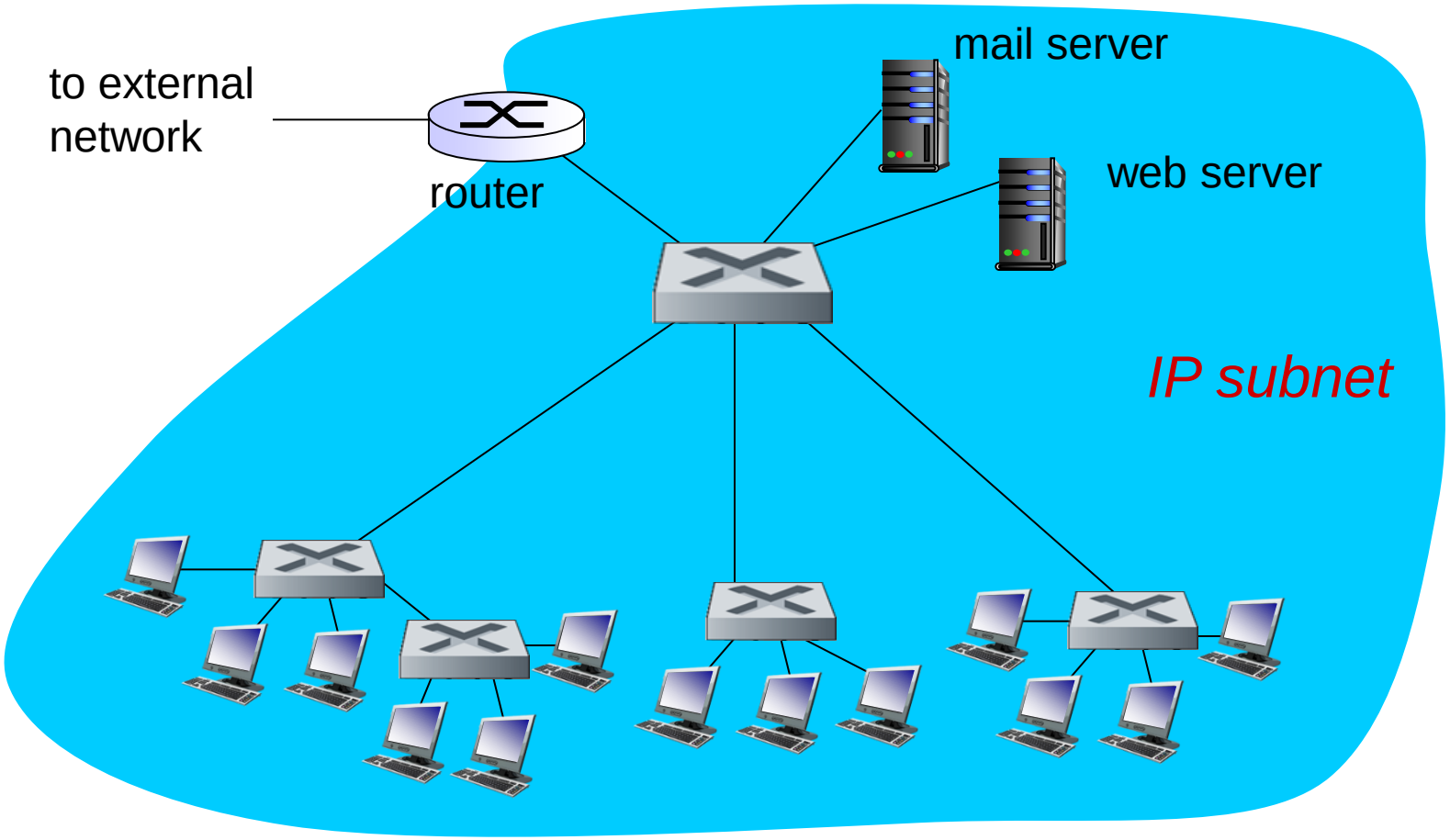
Self-learning multi-switch example

Suppose C sends frame to I, I responds to C



- *Q:* show switch tables and packet forwarding in S_1, S_2, S_3, S_4

Institutional network



Ripasso sulle reti

Obiettivi:

- Richiamare concetti chiave del corso introduttivo sulle reti di calcolatori
  - Rinfrescare la memoria su idee fondamentali
  - Creare una base di partenza comune
  - Identificare possibili lacune e lavoro di ripasso
  - Consolidare la terminologia

Sommario:

- Panoramica ad alto-livello
- Controllo di errore
- Controllo di flusso
- Controllo di congestione
- Indirizzamento
- Livello rete
- Livello link
- Controllo

Quali sono le diverse scale temporali di controllo in una rete?

- livello trasporto
- livello rete
- livello link
- altre importanti scale temporali?

A seguire: funzioni di rete/protocollo comuni

Obiettivi:

- identificare, studiare componenti architetturali, meccanismi protocollari comuni
- *in sintesi:* panoramica ad alto livello
- *in dettaglio:* argomenti importanti non trattati in un corso introduttivo

Sommario:

- segnalazione: rete telefonica, Internet, reti ATM
- gestione dello stato (nella segnalazione)
- randomizzazione
- indirezione
- virtualizzazione
- multiplazione
- progettazione per scalabilità