

Università degli Studi di Roma "Tor Vergata"
Laurea in Informatica

Sistemi Operativi e Reti
(modulo Reti)
a.a. 2024/2025

Livello di rete: piano dei dati (parte1)

dr. Manuel Fiorelli

manuel.fiorelli@uniroma2.it

<https://art.uniroma2.it/fiorelli>

Basate sulle slide del libro di testo:

https://gaia.cs.umass.edu/kurose_ross/ppt.php

Introduction: 1-1

Livello di rete: i nostri obiettivi

- Capire i principi che stanno dietro i servizi del livello di rete, focalizzandosi sul piano dei dati:
 - modelli di servizio del livello di rete
 - funzioni di inoltro e di instradamento
 - come funziona un router
 - indirizzamento
 - inoltro generalizzato
 - architettura di Internet
- Implementazione in Internet
 - protocollo IP
 - NAT, middlebox

Livello di rete: tabella di marcia sul “piano dei dati”

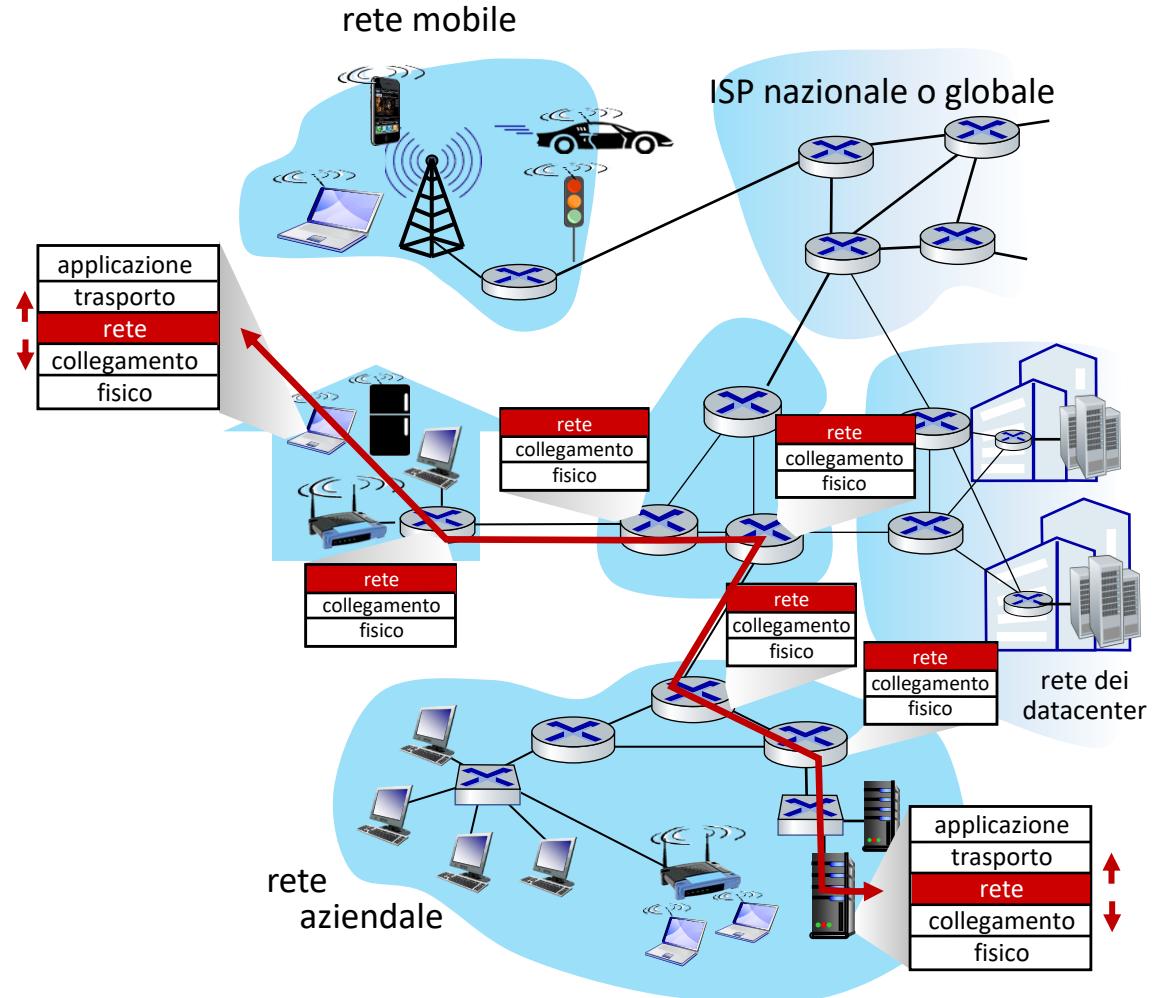
- Livello di rete: panoramica
 - piano dei dati
 - piano di controllo
- Cosa c'è dentro un router
 - porte di ingresso, struttura di commutazione, porte di uscita
 - buffer management, scheduling
- IP: il Protocollo Internet
 - formato dei datagrammi
 - indirizzamento
 - traduzione degli indirizzi di rete
 - IPv6



- inoltro generalizzato, SDN
 - Match+action
 - OpenFlow: match+action in azione
- middlebox

Servizi e protocolli del livello di rete

- trasporta i segmenti dall'host mittente all'host destinatario
 - **mittente:** incapsula i segmenti dentro ai datagrammi che passa al livello di collegamento
 - **destinatario:** consegna i segmenti al protocollo del livello di trasporto
- i protocolli di livello di rete sono implementati da *tutti i dispositivi in Internet*: host, router
- **router:**
 - esamina i campi dell'intestazione di tutti i datagrammi IP che lo attraversano
 - sposta i datagrammi dalle porte di ingresso alla porta di uscita per trasferire il datagramma lungo il percorso dall'host di origine a quello di destinazione



Due funzioni chiave del livello di rete

funzioni del livello di rete

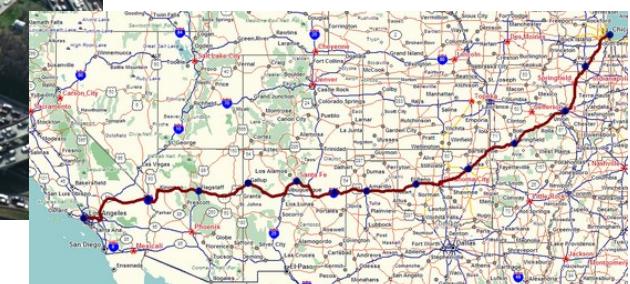
- *inoltro (forwarding)*: trasferisce i pacchetti da un collegamento di ingresso di un router al collegamento di uscita appropriato del router
- *instradamento (routing)*: determina il percorso seguito dai pacchetti dall'origine alla destinazione
 - *algoritmi di instradamento*

analogia: fare un viaggio

- *inoltro*: attraversamento di uno svincolo seguendo le indicazioni dei cartelli
- *instradamento*: pianificazione dei percorsi verso tutte le destinazioni scegliendo tra i molteplici possibili e conseguente installazione dei cartelli



inoltro

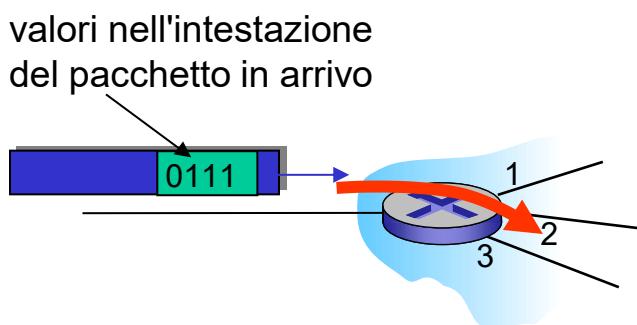


instradamento

Livello di rete: piano dei dati e piano di controllo

Piano dei dati:

- funzione *locale*, a livello di singolo router
- determina come i pacchetti in arrivo a una porta di ingresso del router sono inoltrati verso una porta di uscita del router

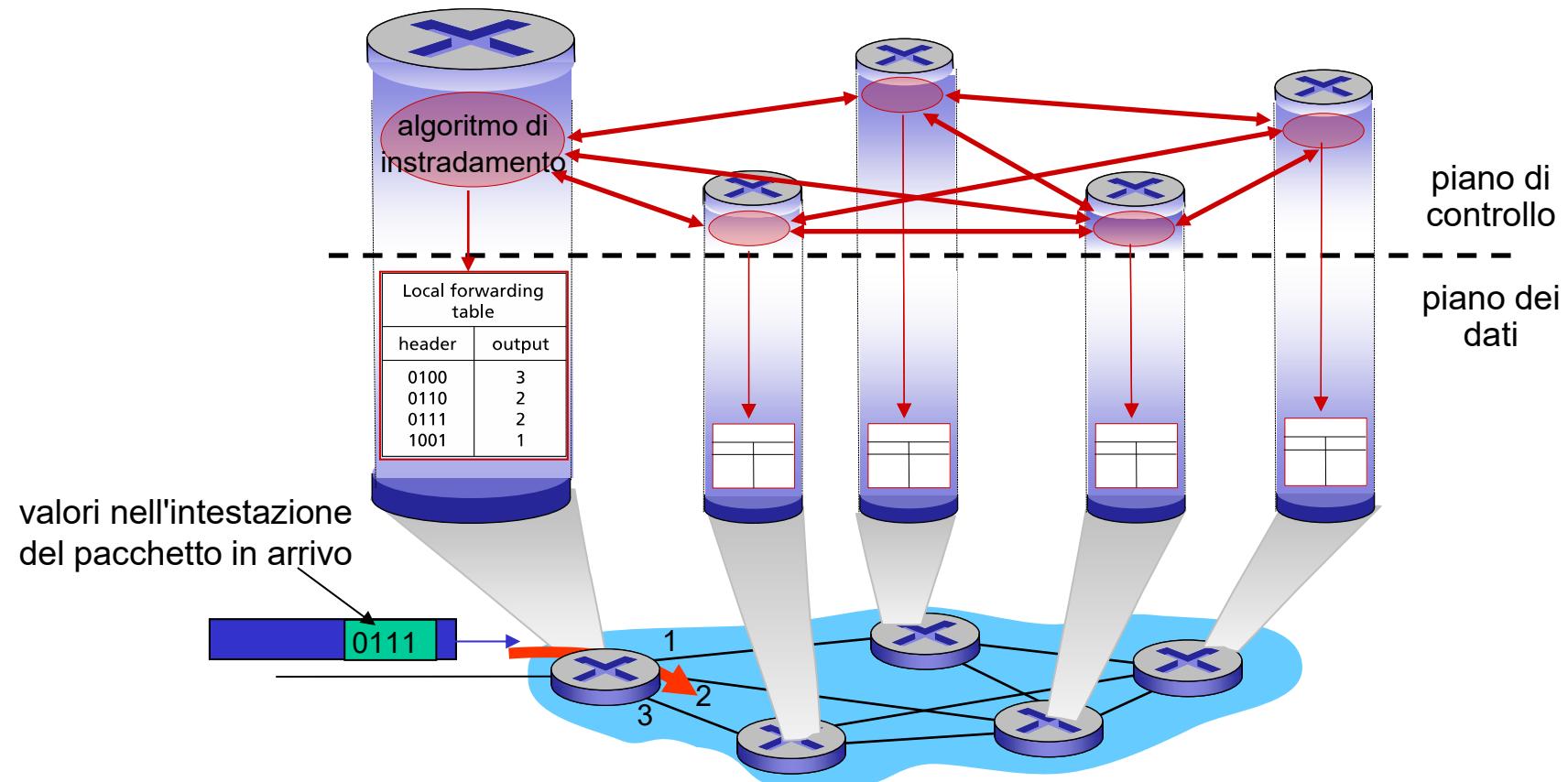


Piano di controllo

- *logica di rete*
- determina come i pacchetti sono instradati tra i router lungo un percorso dall'host di origine all'host di destinazione
- due approcci per il piano di controllo:
 - *algoritmi di instradamento tradizionali*: implementati nei router
 - *software-defined networking (SDN)*: implementato nei server (remoti)

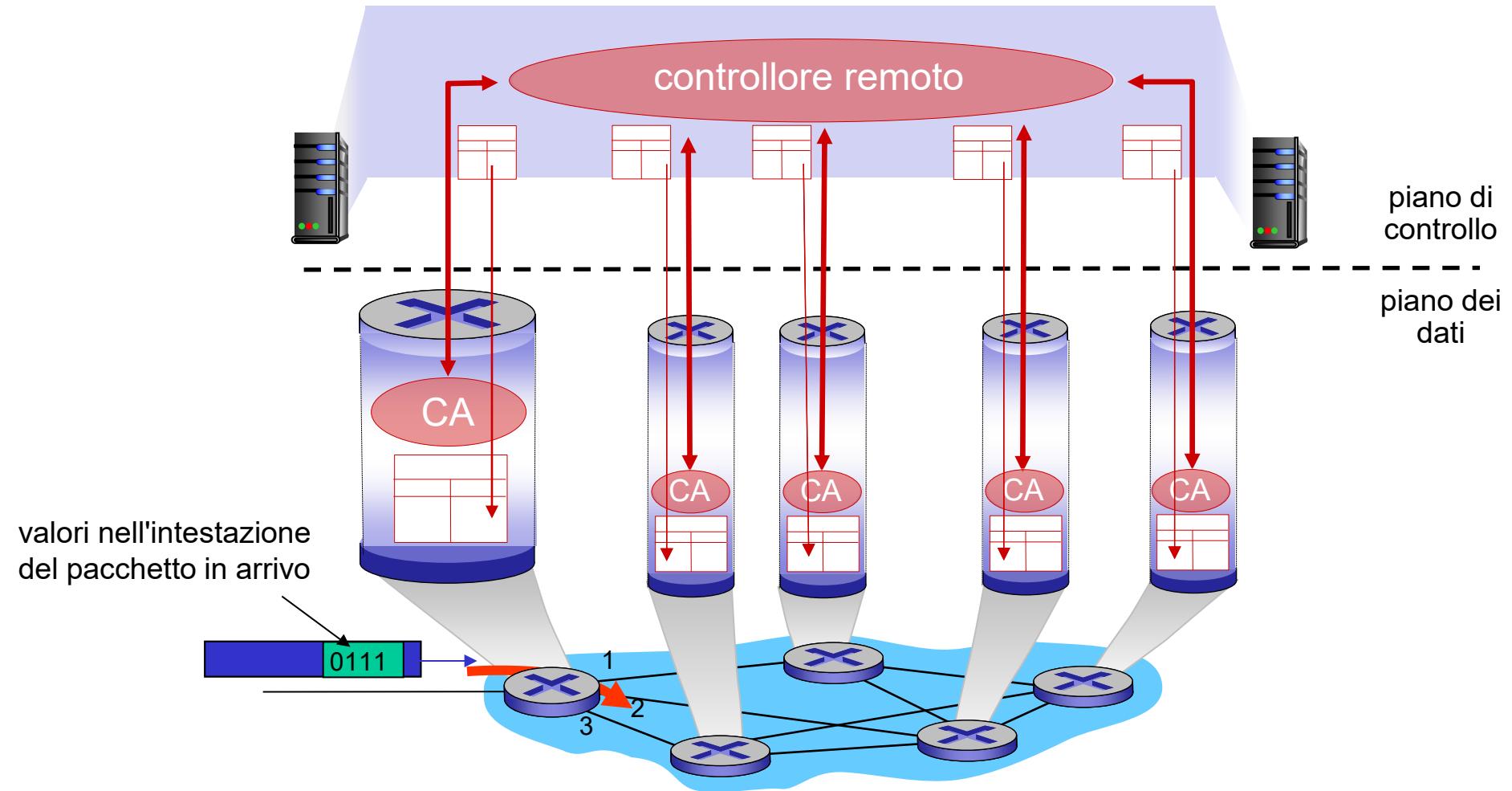
Piano di controllo per router

I singoli componenti dell'algoritmo di routing *in ogni singolo router*.



Software-Defined Networking (SDN)

Un *controllore remoto* calcola e installa le tabelle di inoltro nei router



Modello di servizio del livello di rete

D: Qual è il *modello di servizio* per il “canale” che trasporta i datagrammi dal mittente al destinatario (ovvero le sue caratteristiche)?

Esempi di servizi per un singolo datagramma

- consegna garantita
- consegna garantita con un ritardo inferiore a 40 ms

Esempi di servizi per un *flusso* di datagrammi:

- consegna in ordine
- minima ampiezza di banda garantita
- restrizioni sulle modifiche della spaziatura tra i pacchetti

Modelli di servizi del livello di rete

Architettura di rete	Modello di servizio	Garanzie di qualità del servizio, <i>quality of service (QoS)</i> ?			
		Banda	Consegna	Ordine	Temporizzazione
Internet	best effort	nessuna	no	no	no
ATM	Constant Bit Rate	tasso costante	sì	sì	sì
ATM	Available Bit Rate	min. garantita	no	sì	no
Internet	Intserv Guaranteed (RFC 1633)	sì	sì	sì	sì
Internet	Diffserv (RFC 2475)	possibile	possibilmente	possibilmente	possibilmente no

Modelli di servizi del livello di rete

Architettura di rete	Modello di servizio	Garanzie di qualità del servizio, <i>quality of service (QoS)</i> ?			
		Banda	Consegna	Ordine	Temporizzazione
Internet	best effort	nessuna	no	no	no

Modello di servizio "best effort" di Internet

Nessuna garanzia circa:

- i. consegna del datagramma alla destinazione con successo
- ii. tempi o ordine di consegna
- iii. larghezza di banda disponibile per il flusso da un capo all'altro

Riflessioni sul servizio best effort

- la semplicità del meccanismo ha consentito l'ampia diffusione di internet
- una dotazione sufficiente di larghezza di banda e protocolli in grado di adattarsi alla banda disponibile consentono alle prestazioni delle applicazioni in tempo reale (ad esempio, voce interattiva, video) di essere "sufficientemente buone" per la "maggior parte del tempo"
- servizi replicati e distribuiti a livello applicativo (datacenter, reti di distribuzione dei contenuti) che si collegano alle reti dei clienti e consentono di fornire servizi da più luoghi
- il controllo della congestione dei servizi "elastici" aiuta

Il successo del modello di servizio "best-effort" è difficilmente contestabile

Livello di rete: tabella di marcia sul “piano dei dati”

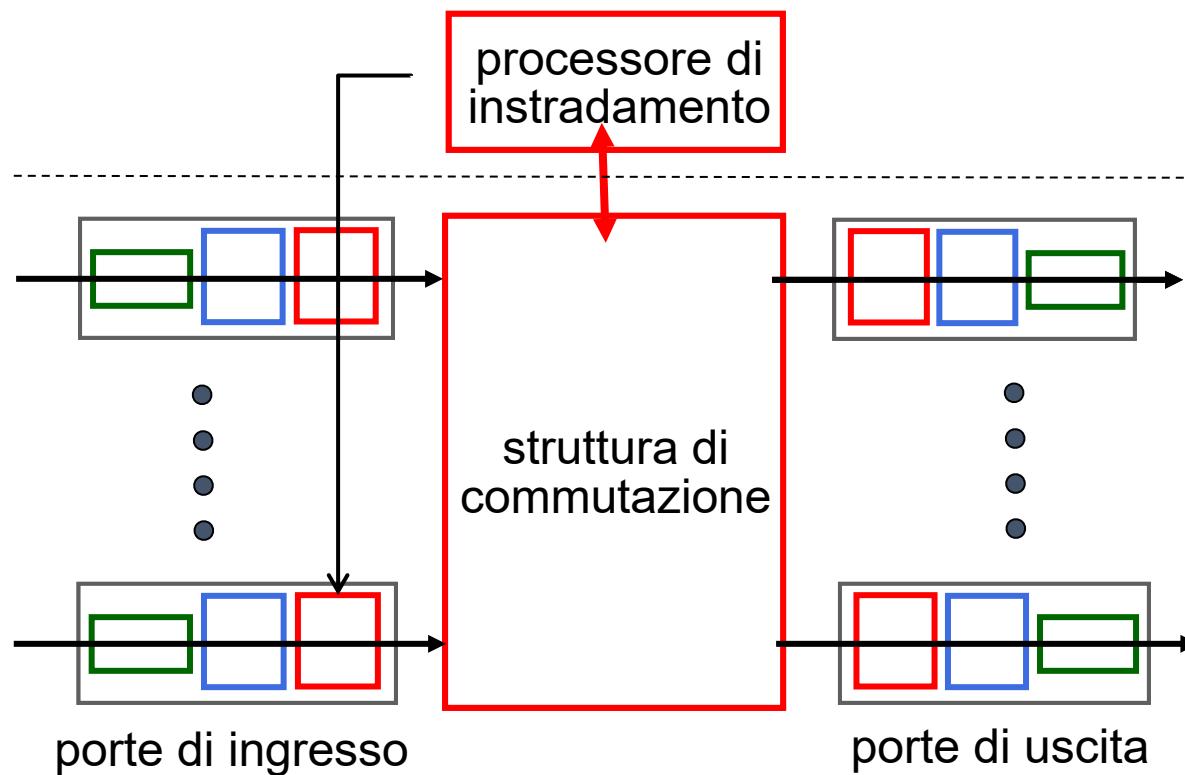
- Livello di rete: panoramica
 - piano dei dati
 - piano di controllo
- Cosa c'è dentro un router
 - Porte di ingresso, struttura di commutazione, porte di uscita
 - buffer management, scheduling
- IP: il Protocollo Internet
 - Formato dei datagrammi
 - indirizzamento
 - Traduzione degli indirizzi di rete
 - IPv6



- Inoltro generalizzato, SDN
 - Match+action
 - OpenFlow: match+action in azione
- Middlebox

Architettura del router

visione ad alto livello di una generica architettura di router:



*piano di controllo
(instradamento, risposta a
malfunzionamenti e gestione)*

(software) opera sulla scala temporale
dei millisecondi o dei secondi

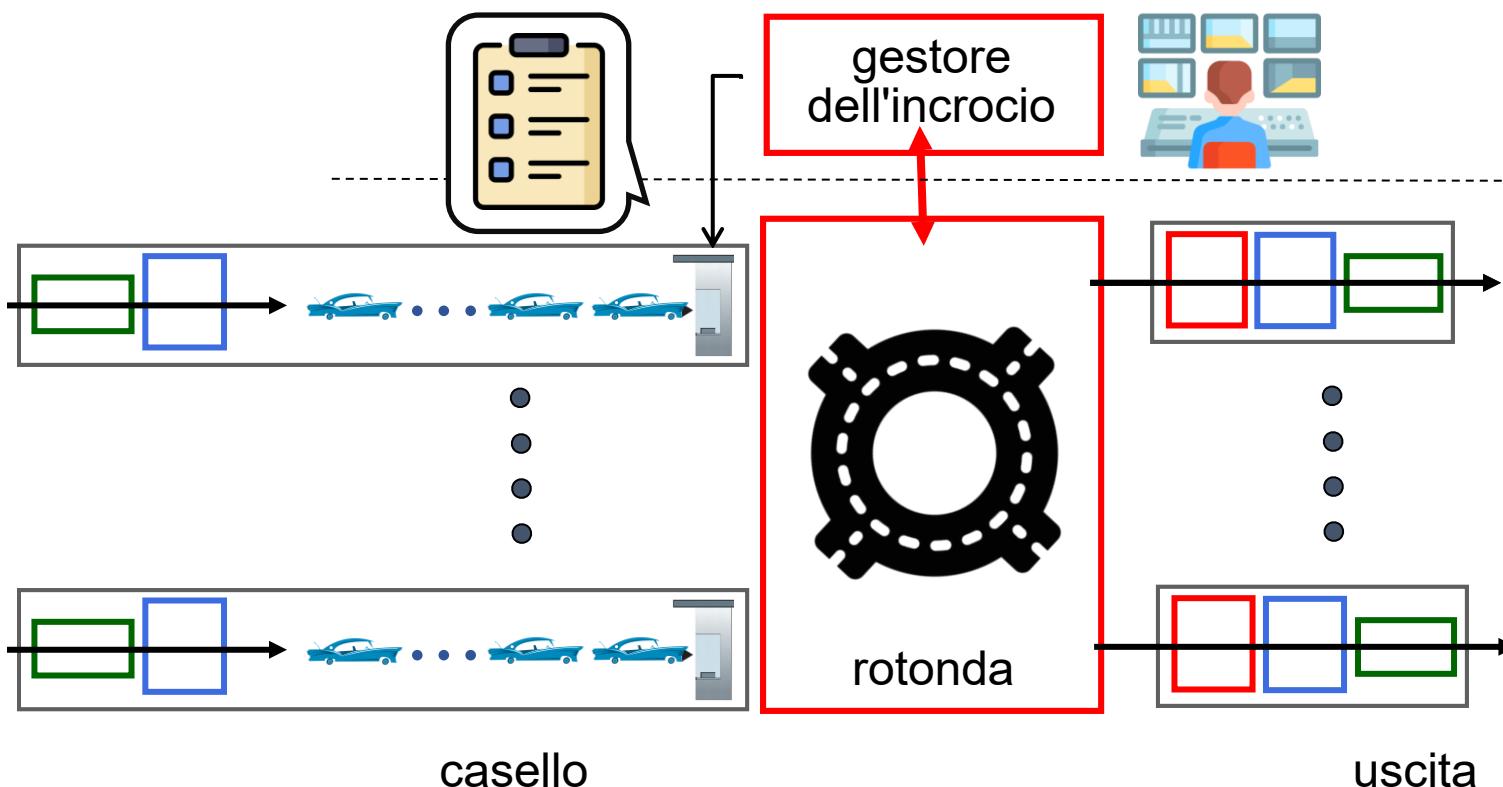
piano dei dati (inoltro)
(hardware) opera
sulla scala temporale
dei nanosecondi

Si consideri un collegamento a 100 Gbps e un datagramma da 64 byte.
Il prossimo arriverà tra:

$$\frac{64 \cdot 8 \text{ bit}}{100 \text{ Gb/s}} = \frac{512 \text{ bit}}{100 \cdot 10^9 \text{ bit/s}} = 5.12 \text{ ns}$$

Architettura del router

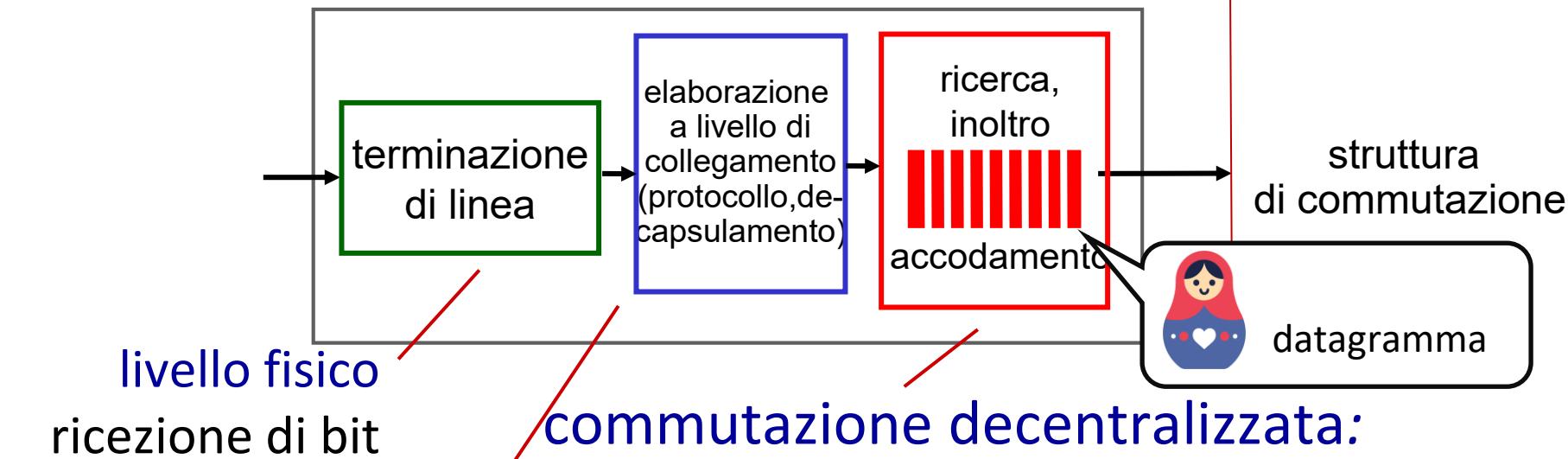
analogia per la architettura generica di router



*piano di controllo
(instradamento, risposta a
malfunzionamenti e gestione)*
(software) opera sulla scala temporale
dei millisecondi o dei secondi

piano dei dati (inoltro)
(hardware) opera
sulla scala temporale
dei nanosecondi

Funzioni delle porte di ingresso



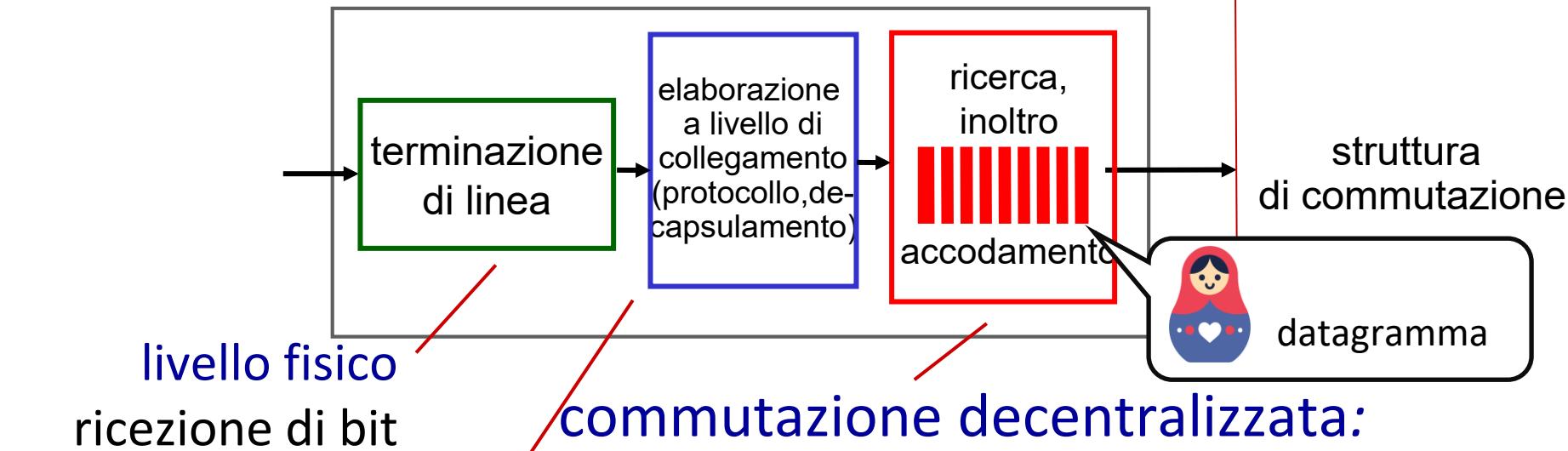
commutazione decentralizzata:

- usando i valori dei campi di intestazione, trova la porta di uscita usando la tabella di inoltro nella memoria della porta di ingresso (*"match plus action"*)
- obiettivo: completare l'elaborazione nella porta di ingresso alla "velocità della linea"
- **accodamento presso la porta di ingresso:** se i datagrammi arrivano più velocemente di quanto la struttura di commutazione possa trasferirli



Frame

Funzioni delle porte di ingresso



Livello di collegamento:
Es., Ethernet



Frame

commutazione decentralizzata:

- usando i valori dei campi di intestazione, trova la porta di uscita usando la tabella di inoltro nella memoria della porta di ingresso (*"match plus action"*)
- **inoltro basato sulla destinazione:** inoltro basato esclusivamente sull'indirizzo IP di destinazione (tradizionale)
- **inoltro generalizzato:** inoltro basato su più campi di intestazione

Destinazione basata sull'indirizzo di destinazione

forwarding table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

D: ma cosa succede se gli intervalli non si dividono così bene?

Destinazione basata sull'indirizzo di destinazione

forwarding table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00010000 00000100 through 11001000 00010111 00010000 00000111	3
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Corrispondenza a prefisso più lungo

Corrispondenza a prefisso più lungo

quando si cerca una voce della tabella di inoltro per un dato indirizzo di destinazione, si usa il prefisso di indirizzo *più lungo* che corrisponde all'indirizzo di destinazione.

Intervallo di indirizzi di destinazione	Interfaccia
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011*** *****	2
altrimenti	3

esempi:

- 11001000 00010111 00010110 10100001 quale interfaccia?
11001000 00010111 00011000 10101010 quale interfaccia?

Corrispondenza a prefisso più lungo

Corrispondenza a prefisso più lungo

quando si cerca una voce della tabella di inoltro per un dato indirizzo di destinazione, si usa il prefisso di indirizzo *più lungo* che corrisponde all'indirizzo di destinazione.

Intervallo di indirizzi di destinazione	Interfaccia
11001000 00010111 00010*****	0
11001000 00010111 00011000 *****	1
11001000 00010111 0011*** *****	2
altrimenti	3

esempi:

corrispondenza!

quale interfaccia?

quale interfaccia?

Corrispondenza a prefisso più lungo

Corrispondenza a prefisso più lungo

quando si cerca una voce della tabella di inoltro per un dato indirizzo di destinazione, si usa il prefisso di indirizzo *più lungo* che corrisponde all'indirizzo di destinazione.

Intervallo di indirizzi di destinazione	Interfaccia
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011*** *****	2
otherwise	3

corrispondenza!

e esempi:

110010uu uuuuuuu uu010110 10100001 quale interfaccia?
11001000 00010111 00011000 10101010 quale interfaccia?

Corrispondenza a prefisso più lungo

Corrispondenza a prefisso più lungo

quando si cerca una voce della tabella di inoltro per un dato indirizzo di destinazione, si usa il prefisso di indirizzo *più lungo* che corrisponde all'indirizzo di destinazione.

Intervallo di indirizzi di destinazione	Interfaccia
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011*** *****	2
altrimenti	3

corrispondenza!

e esempi:

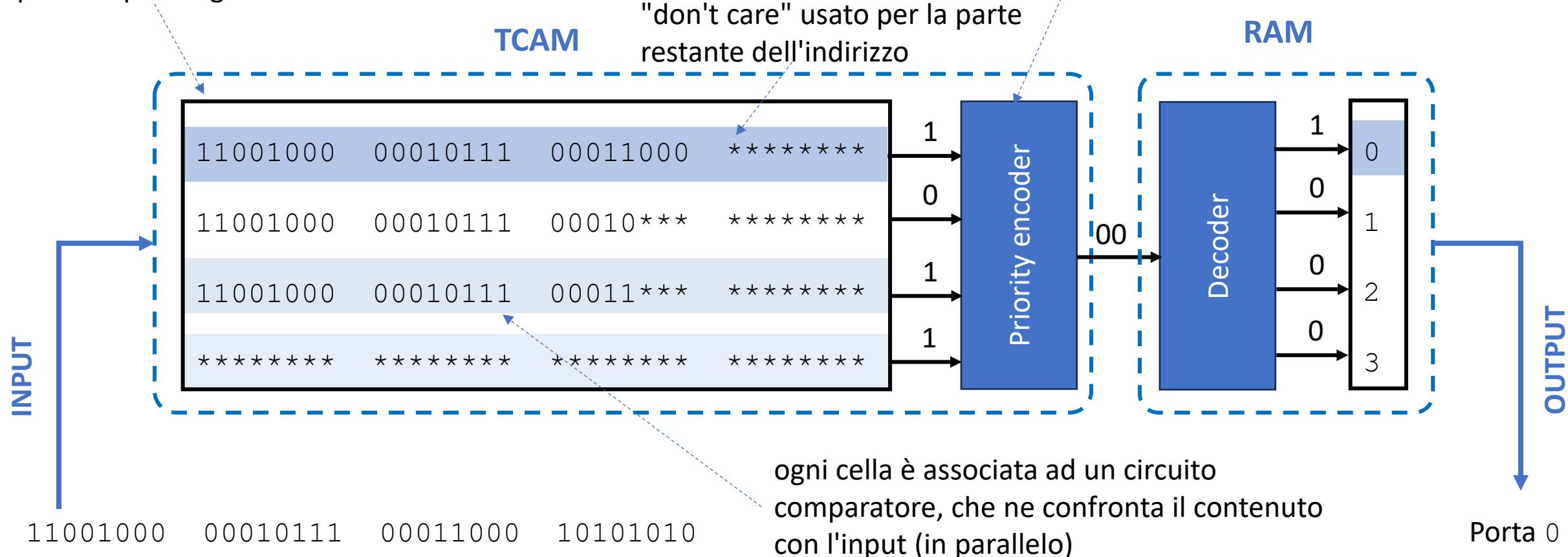
11001000 00010111 00010110	10100001 quale interfaccia?
11001000 00010111 00011000	10101010 quale interfaccia?

Corrispondenza a prefisso più lungo

- Vedremo a breve *perché* viene usata la corrispondenza a prefisso più lungo, quando studieremo l'indirizzamento
- corrispondenza a prefisso più lungo: spesso eseguito con le ternary content addressable memories (TCAMs)
 - *content addressable*: un indirizzo IP a 32 bit è passato alla memoria che restituisce il contenuto della tupla nella tabella di inoltro corrispondente a quell'indirizzo in un tempo essenzialmente costante
 - Cisco Catalyst: ~1M voci nella tabella di inoltro in TCAM

Corrispondenza a prefisso più lungo

Occorre memorizzare i prefissi dal più lungo al più corto, in modo che la prima corrispondenza selezionata dal *priority encoder* sia quella per il prefisso più lungo

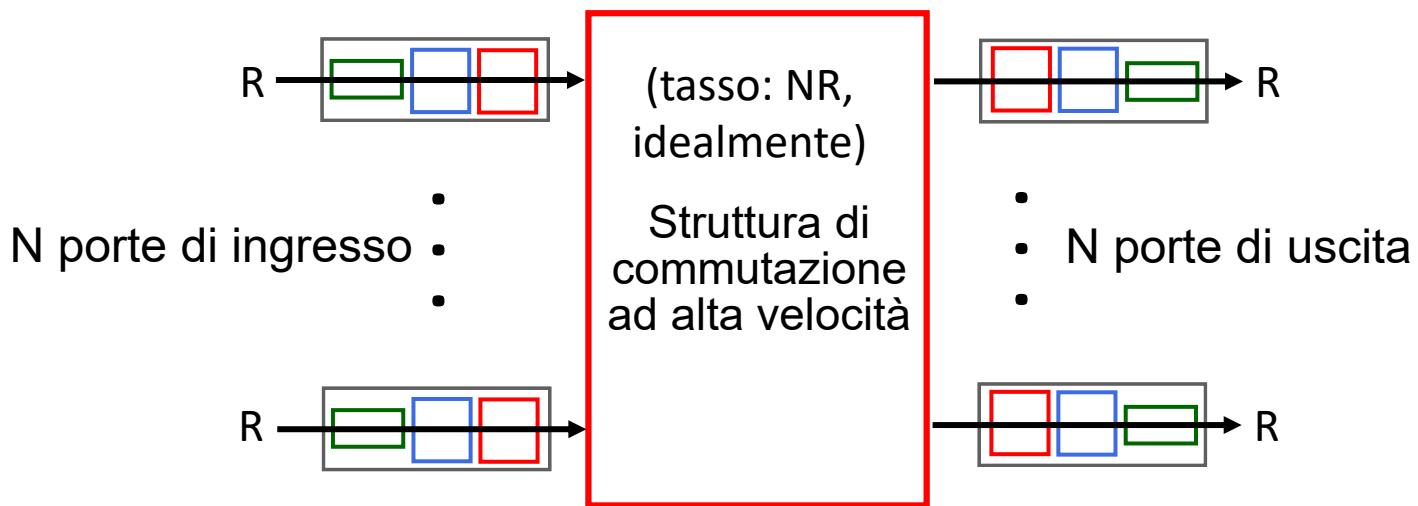


Vagamente basato su:

Irfan, Muhammad & Ullah, Dr. Zahid & Cheung, Ray C.C.. (2019). A High-performance Distributed RAM based TCAM Architecture on FPGAs. IEEE Access. PP. 10.1109/ACCESS.2019.2927108.

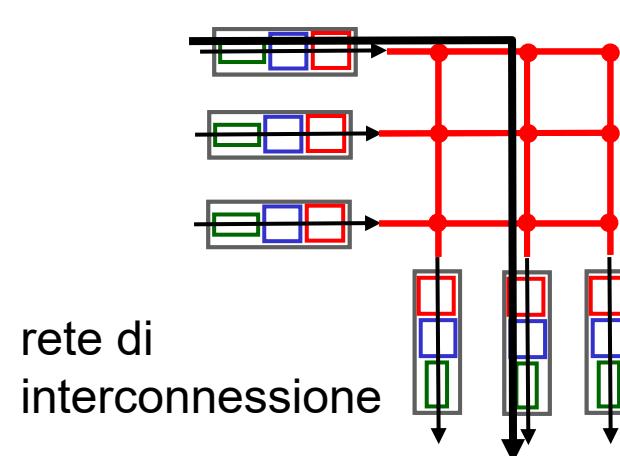
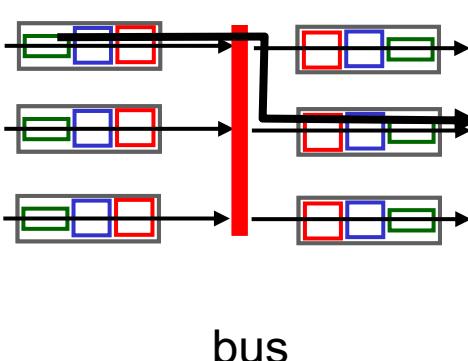
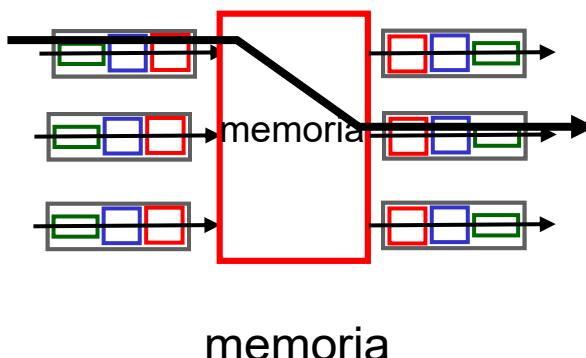
Struttura di commutazione (*switching fabric*)

- trasferisce i pacchetti dal collegamento di ingresso al collegamento di uscita appropriato
- **tasso di trasferimento:** tasso al quale i pacchetti vengono trasferiti dalla porta di input a quella di output
 - Spesso misurato come multiplo del tasso di trasmissione delle linee di input/output
 - N input: si desidera avere un tasso di trasferimento della struttura di commutazione N volte il tasso delle linee di input/output



Struttura di commutazione (*switching fabric*)

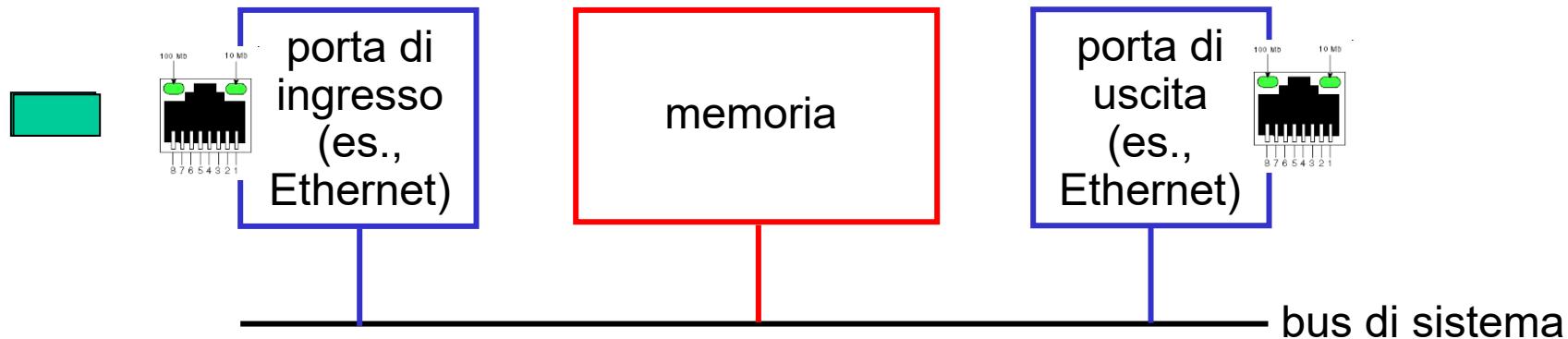
- trasferisce i pacchetti dal collegamento di ingresso al collegamento di uscita appropriato
- **tasso di trasferimento:** tasso al quale i pacchetti vengono trasferiti dalla porta di input a quella di output
 - Spesso misurato come multiplo del tasso di trasmissione delle linee di input/output
 - N input: si desidera avere un tasso di trasferimento della struttura di commutazione N volte il tasso delle linee di input/output



Commutazione in memoria

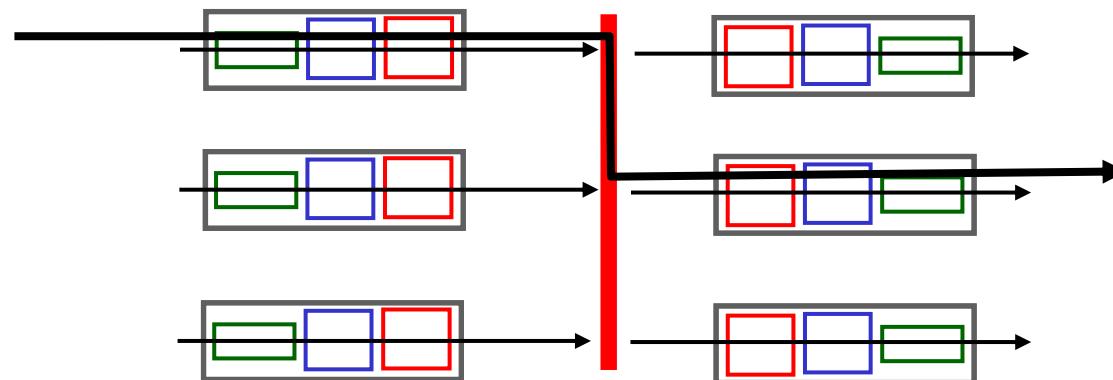
router di prima generazione:

- computer tradizionali con commutazione sotto il diretto controllo della CPU
- pacchetti copiati nella memoria del sistema
- velocità limitata dall'ampiezza di banda della memoria (2 attraversamenti del bus per datagramma)



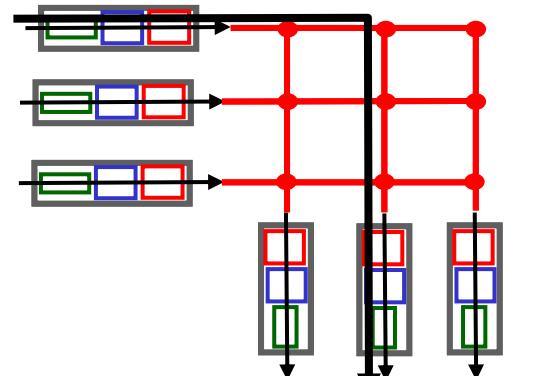
Commutazione tramite bus

- le porte di ingresso trasferiscono un pacchetto direttamente alle porte di uscita tramite un bus condiviso
- *bus contention*: velocità di commutazione limitata dalla velocità del bus
- bus a 32 Gbps, Cisco 5600: velocità sufficiente per router di accesso

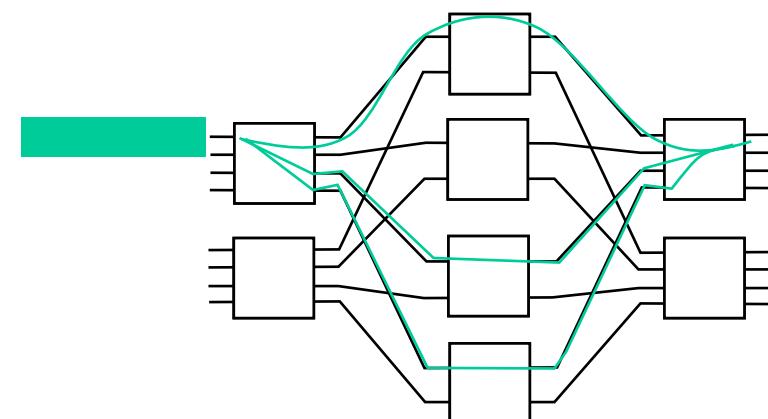


Commutazione attraverso rete di interconnessione

- Crossbar (matrice di commutazione), reti Clos, altre reti di interconnessione sviluppate originariamente per architetture multiprocessore
- multistage switch: switch $n \times n$ da più stadi di switch più piccoli
- sfruttare il parallelismo:
 - frammenta il datagramma in celle di lunghezza fissa all'ingresso
 - commutare le celle attraverso la rete di commutazione, riassemblare il datagramma in uscita



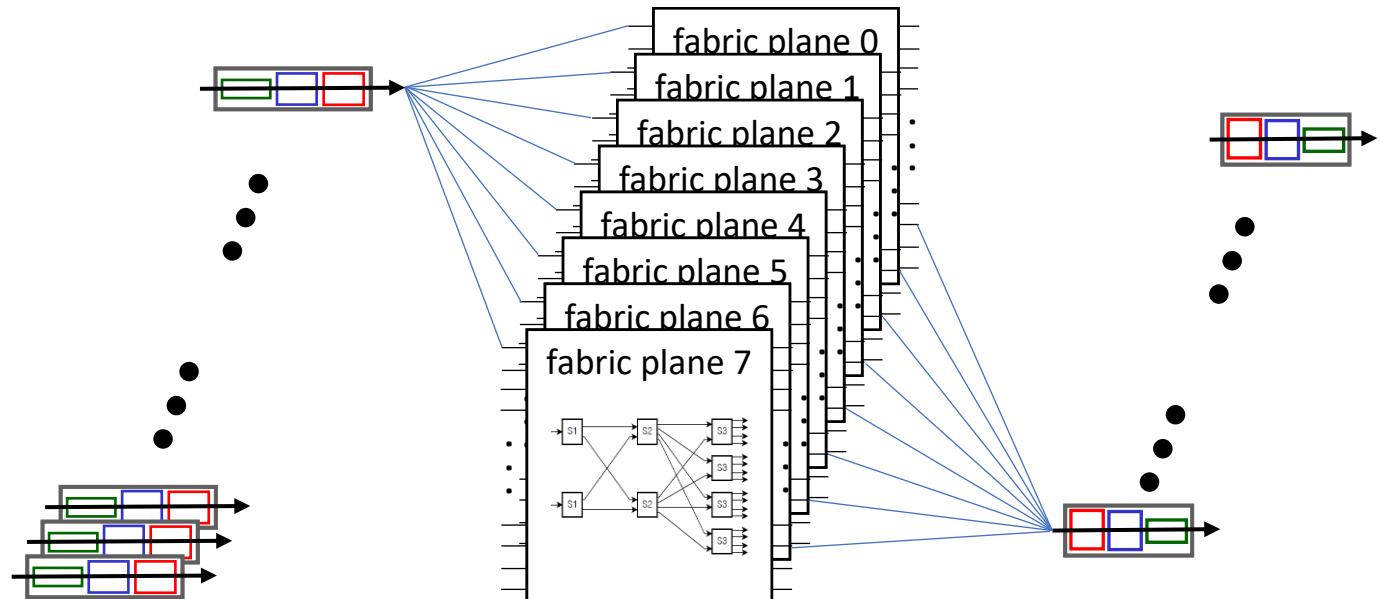
3x3 crossbar switch



8x8 multistage switch
costruito da switch più piccoli

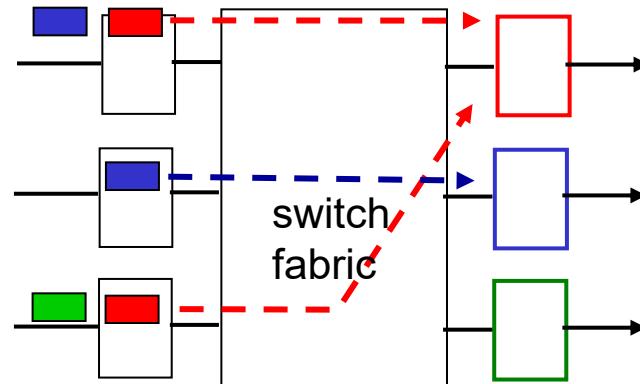
Commutazione attraverso rete di interconnessione

- scalare usando molteplici piani di commutazione in parallelo:
 - speedup, scaleup attraverso il parallelismo
- Cisco CRS router:
 - unità di base: 8 switching plane
 - ogni plane: rete di interconnessione a 3 stadi
 - Capacità di commutazione fino a centinaia di Tbps

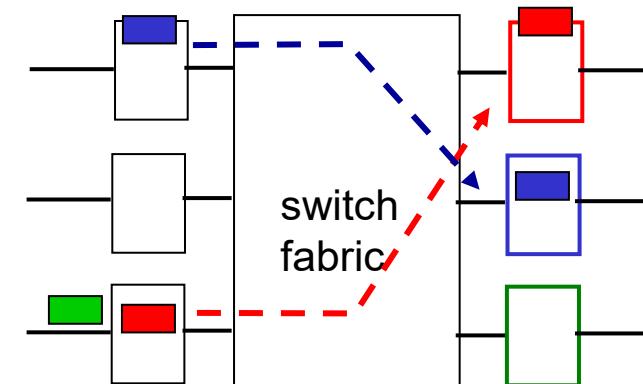


Accodamento sulle porte di ingresso

- Se la struttura di commutazione è più lenta della porta di ingresso combinate -> può verificarsi accodamento sulle porte di ingresso
 - ritardo di accodamento e perdite dovute all'overflow dei buffer di input!
- **Blocco in testa alla coda [Head-of-the-Line (HOL) blocking]**: il datagramma accodato all'inizio della coda impedisce agli altri in coda di avanzare



contesa della porta di uscita: soltanto un datagramma rosso può essere trasferito.
Il pacchetto rosso in basso è **bloccato**

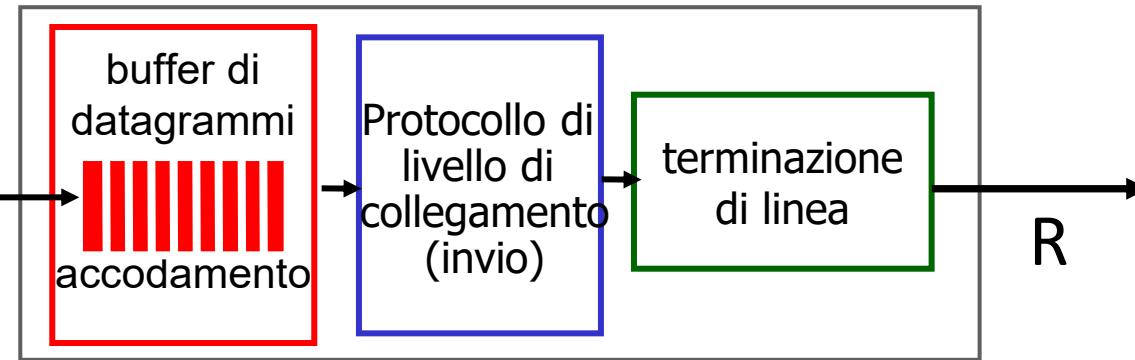


dopo il trasferimento di un pacchetto: il pacchetto verde sta sperimentando il **blocco in testa alla coda**

Accodamento in uscita



struttura di
commutazione
(tasso: NR)



questa è una slide importante

- **Buffering** richiesto quando i datagrammi arrivano dalla struttura di commutazione più velocemente del tasso di trasmissione del collegamento. **Drop policy:** quale datagramma scartare se il buffer non è sufficiente?

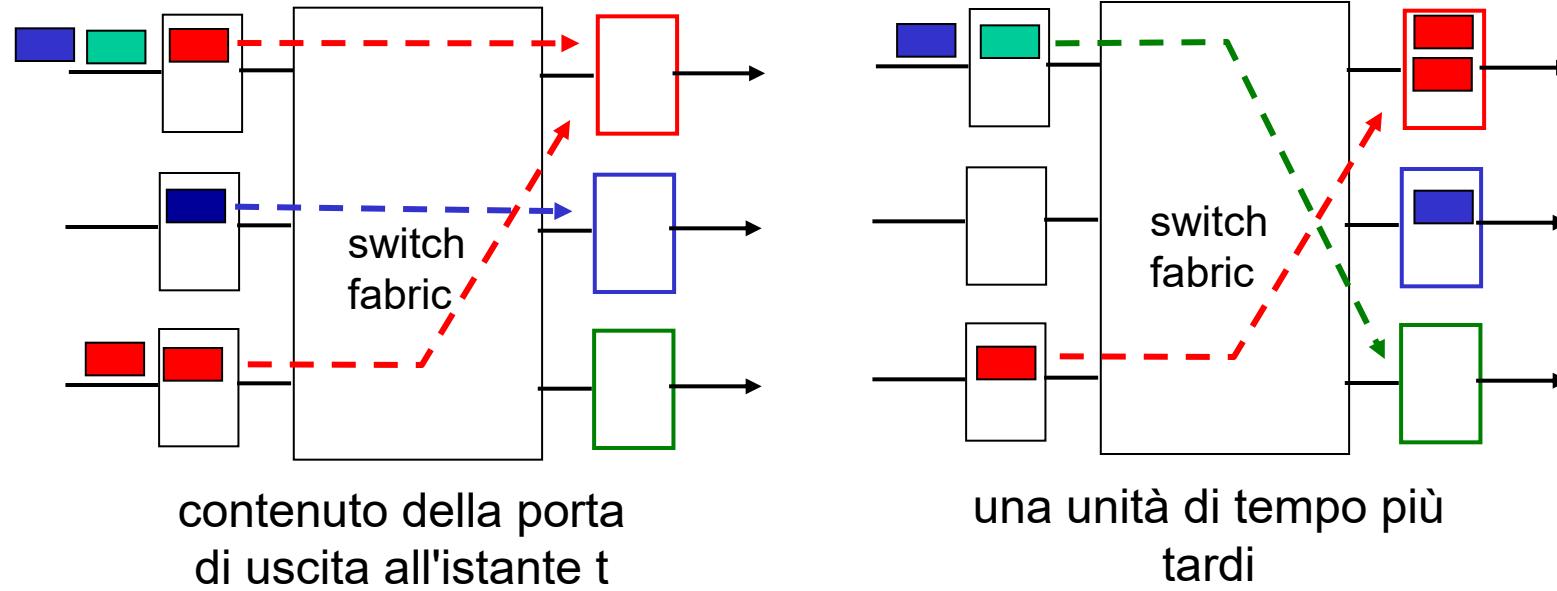
I datagrammi possono essere persi a causa di congestione, mancanza di buffer

- **Disciplina di scheduling** sceglie tra i datagrammi in coda quale trasmettere

Schedulazione con priorità

- chi ottiene le migliori prestazioni, neutralità della rete

Accodamento in uscita



- buffering quando il tasso di arrivo attraverso la struttura di commutazione supera la velocità delle linea di uscita
- *accodamento (ritardo) e perdite causata dall'overflow del buffer della porta di uscita!*

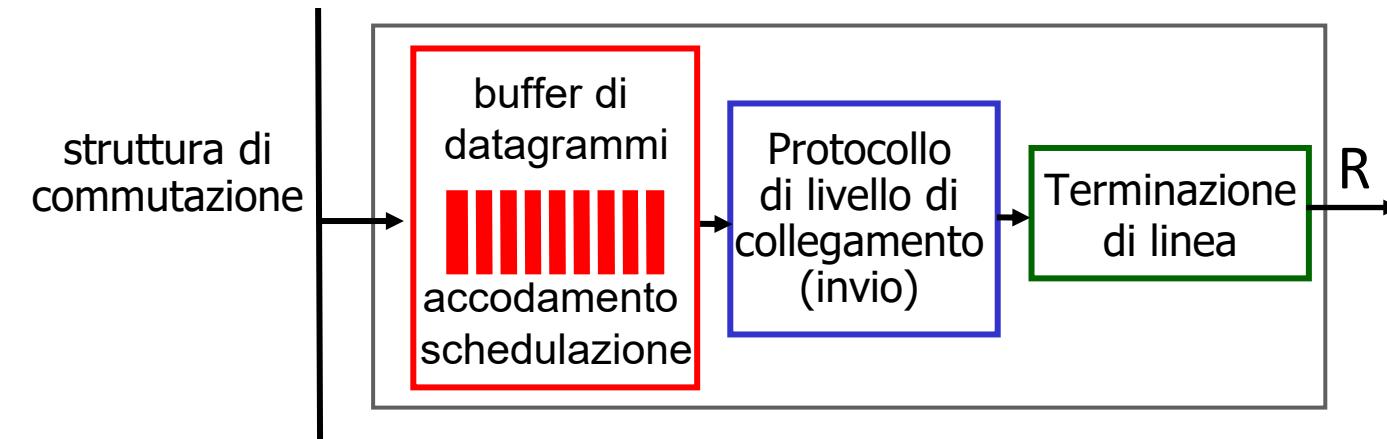
Quanta memoria di buffer è necessaria?

- RFC 3439 rule of thumb: buffering medio uguale al prodotto del RTT “tipico” (diciamo 250 ms) per la capacità del collegamento C
 - es., capacità del collegamento C = 10 Gbps: buffer di 2.5 Gbit
- raccomandazione più recente: con N flussi, dimensione del buffer

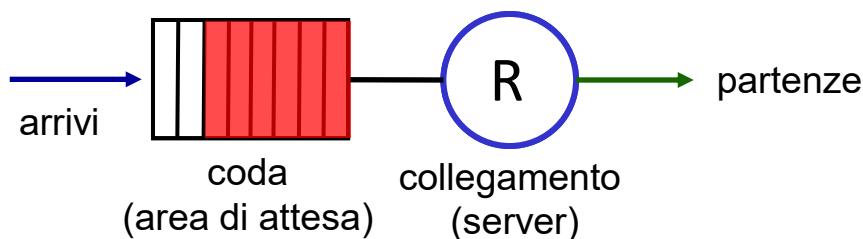
$$\frac{RTT \cdot C}{\sqrt{N}}$$

- ma *troppo* buffering può aumentare i ritardi (soprattutto nei router domestici)
 - RTT elevato: prestazioni scarse delle applicazioni real-time, mittenti TCP meno reattivi alla congestione e alla perdita dei pacchetti
 - ricordiamoci del controllo di congestione basato sul ritardo: “mantenere il collegamento collo di bottiglia sufficientemente pieno (occupato) ma non più pieno”

Gestione del buffer



Astrazione: coda



gestione del buffer:

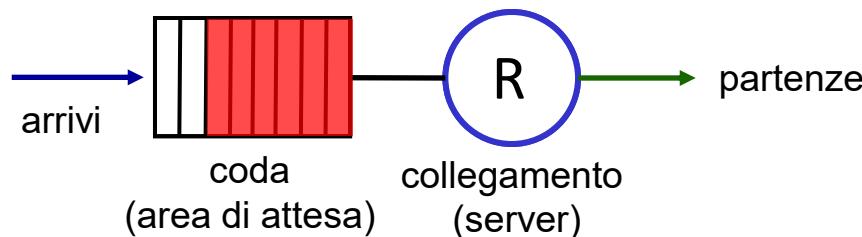
- **politica di scarto (drop):** quale pacchetto eliminare quando la coda è piena
 - **tail drop:** scarta il pacchetto in arrivo
 - **priorità:** scarta/rimuovi in base alla priorità
- **marcatura:** quali pacchetti marcare per segnalare la congestione (ECN, RED)

Schedulazione dei pacchetti: FCFS

Schedulazione dei pacchetti:
decidere quale pacchetto inviare
successivamente sul
collegamento

- first come, first served
- priority
- round robin
- weighted fair queueing

Astrazione: coda



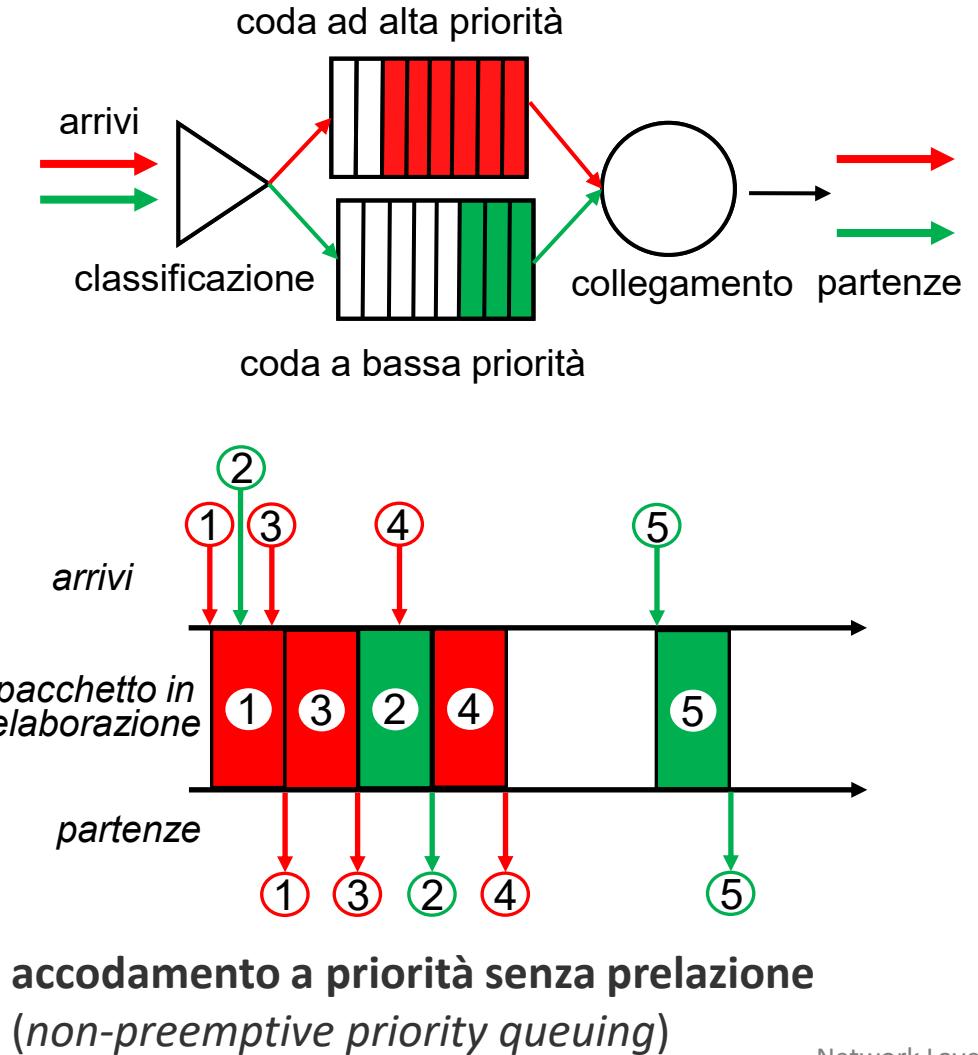
FCFS: pacchetti trasmessi in
ordine di arrivo alla porta
di uscita

- conosciuta anche come:
First-in-first-out (FIFO)
- esempi del mondo reale?

Schedulazione dei pacchetti : priority

Schedulazione con priorità:

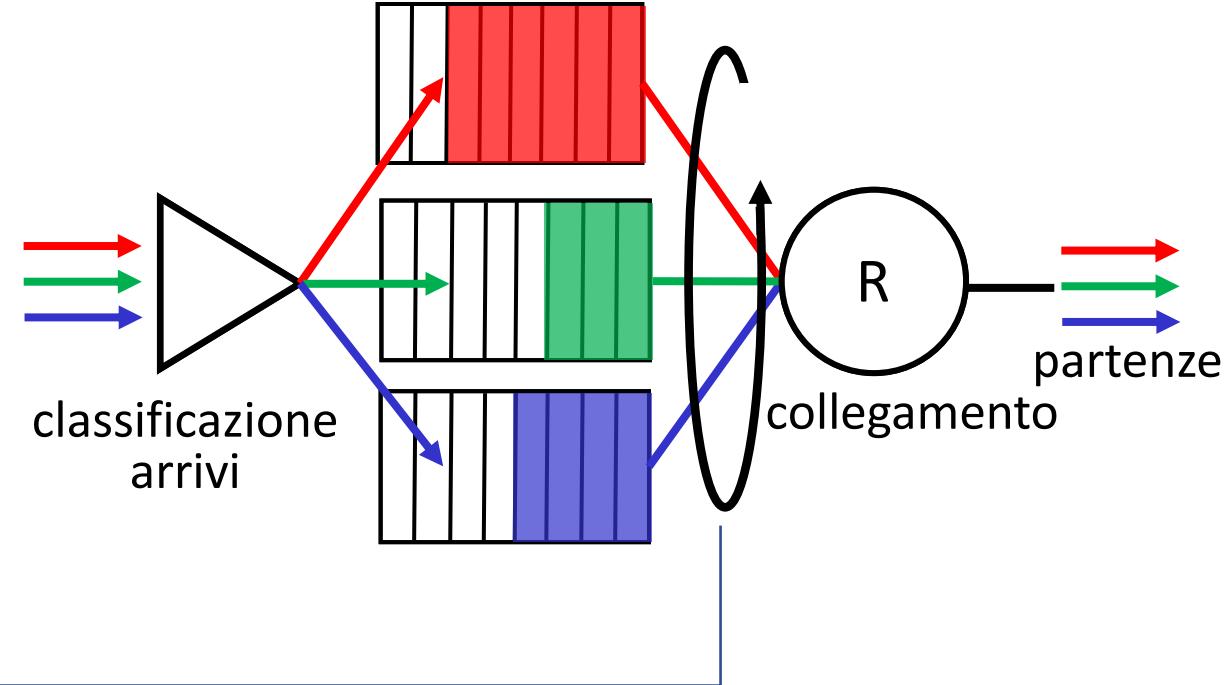
- traffico in arrivo classificato, accodato per classi
 - qualsiasi campo di intestazione può essere usato per la classificazione
- invia il pacchetto dalla coda non vuota con priorità più alta
 - FCFS all'interno di ciascuna classe
 - possibilità di *starvation*: un pacchetto può attendere indefinitivamente se continuano a arrivare pacchetti con priorità maggiore



Schedulazione dei pacchetti: round robin

Round Robin (RR) scheduling:

- Traffico in arrivo classificato, accodato per classi
 - qualsiasi campo di intestazione può essere usato per la classificazione
- Il server esegue ciclicamente e ripetutamente la scansione delle code di classe, inviando a turno un pacchetto completo di ogni classe (se disponibile).



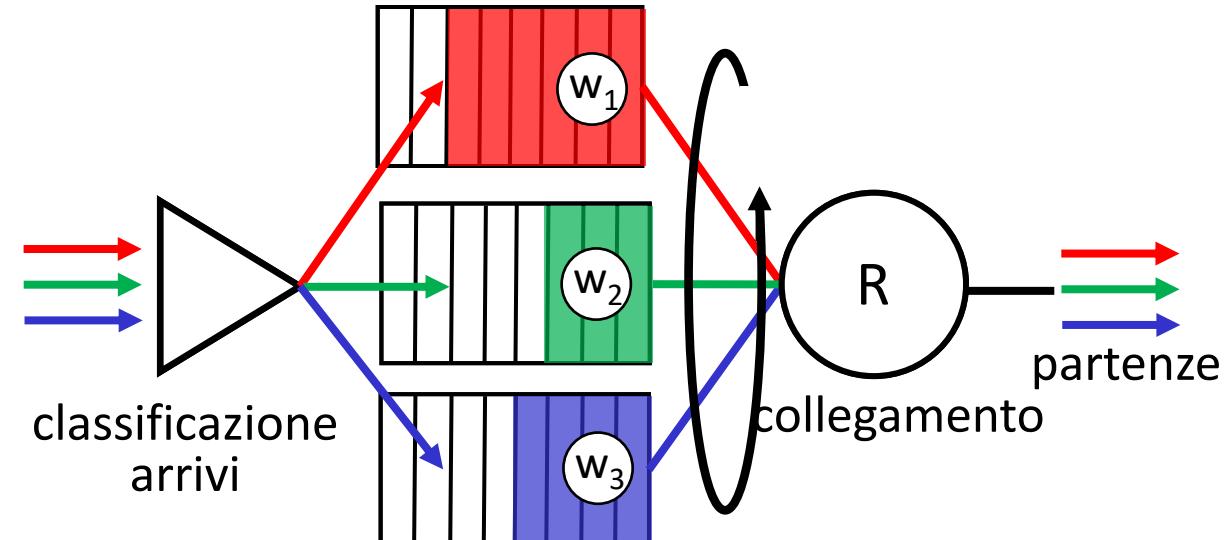
Schedulazione dei pacchetti: weighted fair queueing

Weighted Fair Queueing (WFQ):

- generalizza Round Robin
- ciascuna classe, i , ha un peso, w_i , e riceve una quantità ponderata di servizio in ogni ciclo :

$$\frac{w_i}{\sum_j w_j}$$

- garanzia di larghezza di banda minima (per classe di traffico)



Barra laterale: Neutralità della rete

Cos'è la neutralità della rete (*net neutrality*)?

- *tecnica*: come un ISP dovrebbe condividere/allocare le proprie risorse
 - la schedulazione dei pacchetti e la gestione dei buffer sono i *meccanismi*
- Principi *sociali e economici*
 - proteggere la libertà di espressione
 - Incoraggiare l'innovazione, la competizione
- Far rispettare *politiche* e *leggi*

Ogni paese ha il proprio approccio alla neutralità della rete

Barra laterale: Neutralità della rete

2015 US FCC *Order on Protecting and Promoting an Open Internet*: tre regole definite “clear, bright line”:

- **no blocking** ... “non bloccherà i contenuti, le applicazioni, i servizi o i dispositivi non dannosi leciti, fatta salva una ragionevole gestione della rete.”
- **no throttling** ... “non devono pregiudicare o degradare il traffico Internet lecito sulla base del contenuto, dell'applicazione o del servizio Internet o dell'uso di un dispositivo non dannoso, fatta salva una ragionevole gestione della rete.”
- **no paid prioritization**. ... “non deve impegnarsi nella prioritizzazione a pagamento”

Nel 2017, la *Restoring Internet Freedom Order* ha annullato questi divieti, concentrandosi invece sulla trasparenza degli ISP.

ISP: telecommunications or information service?

Un ISP è un "servizio di telecomunicazione" o un fornitore di "servizi di informazione"?

- la risposta è importante dal punto di vista normativo!

US Telecommunication Act del 1934 e 1996:

- *Titolo II*: impone “common carrier duties” ai *servizi di telecomunicazione*: tariffe ragionevoli, non discriminazione e richiede una regolamentazione
- *Titolo I*: si applica ai *servizi di informazione*:
 - no common carrier duties (*non regolamentato*)
 - ma concede alla FCC l'autorità "... necessaria per l'esecuzione delle sue funzioni "

Livello di rete: tabella di marcia sul “piano dei dati”

- Livello di rete: panoramica
 - piano dei dati
 - piano di controllo
- Cosa c'è dentro un router
 - Porte di ingresso, struttura di commutazione, porte di uscita
 - buffer management, scheduling
- IP: il Protocollo Internet
 - Formato dei datagrammi
 - indirizzamento
 - Traduzione degli indirizzi di rete
 - IPv6

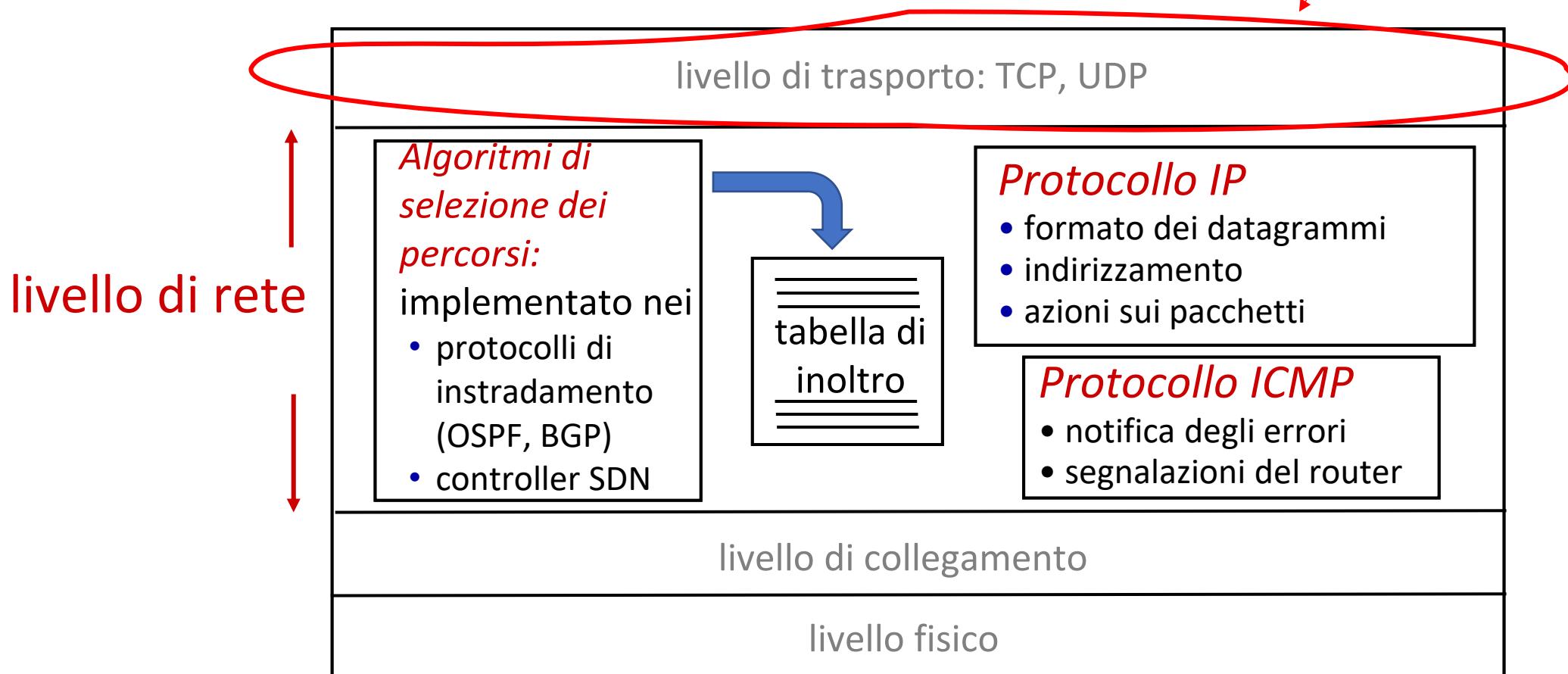


- Inoltro generalizzato, SDN
 - Match+action
 - OpenFlow: match+action in azione
- Middlebox

Livello di rete: Internet

Attenzione: i router non implementano questo livello!

Uno sguardo a livello di rete Internet



Formato dei datagrammi IP

numero di versione del protocollo IP

lunghezza della intestazione
(multipli di 32 bit)

Type of service (ToS):
▪ DiffServ (0:5)
▪ ECN (6:7)

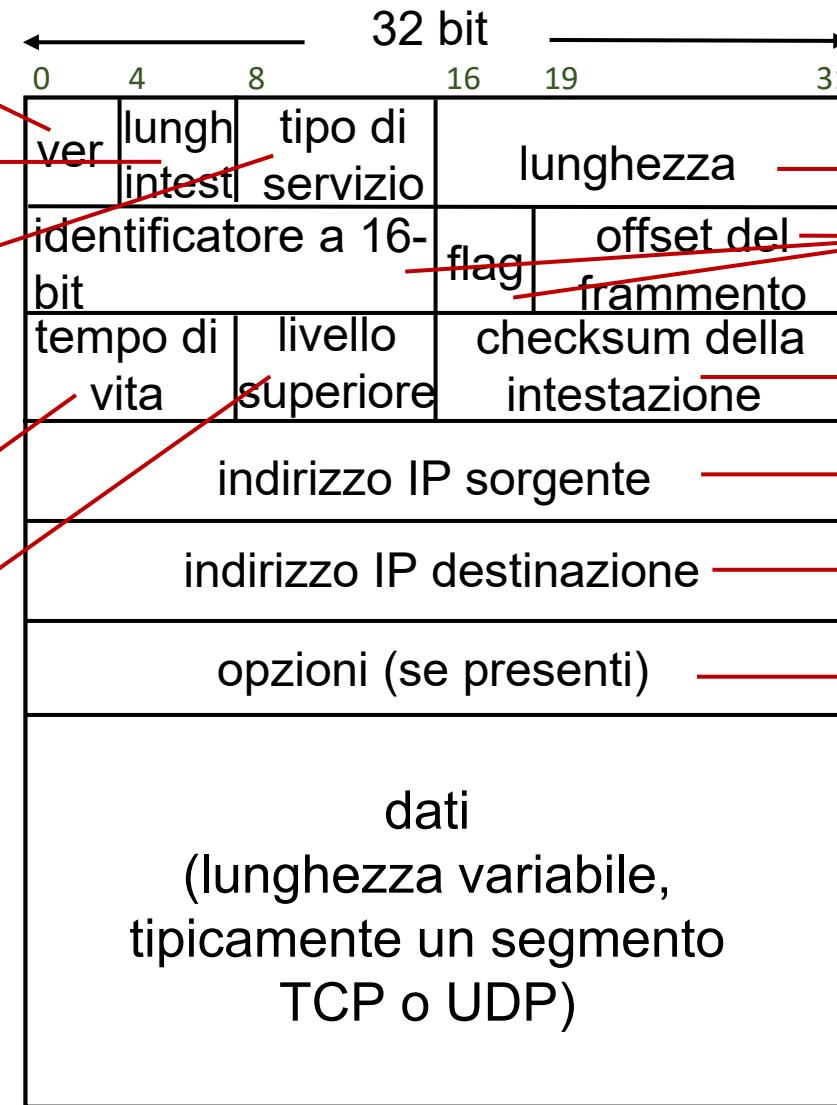
TTL: numero massimo di hop
rimanenti

(decrementato a ogni router)

protocollo di livello superiore (es., TCP
= 6 o UDP = 17) cui consegnare i dati

overhead

- 20 byte di TCP
- 20 byte di IP
- = 40 byte + overhead
di livello applicazione
per TCP+IP



Lunghezza massima: 65535 byte
Tipicamente: 1500 byte o meno

lunghezza totale del
datagramma (byte)

frammentazione/
riassemblaggio

checksum intestazione

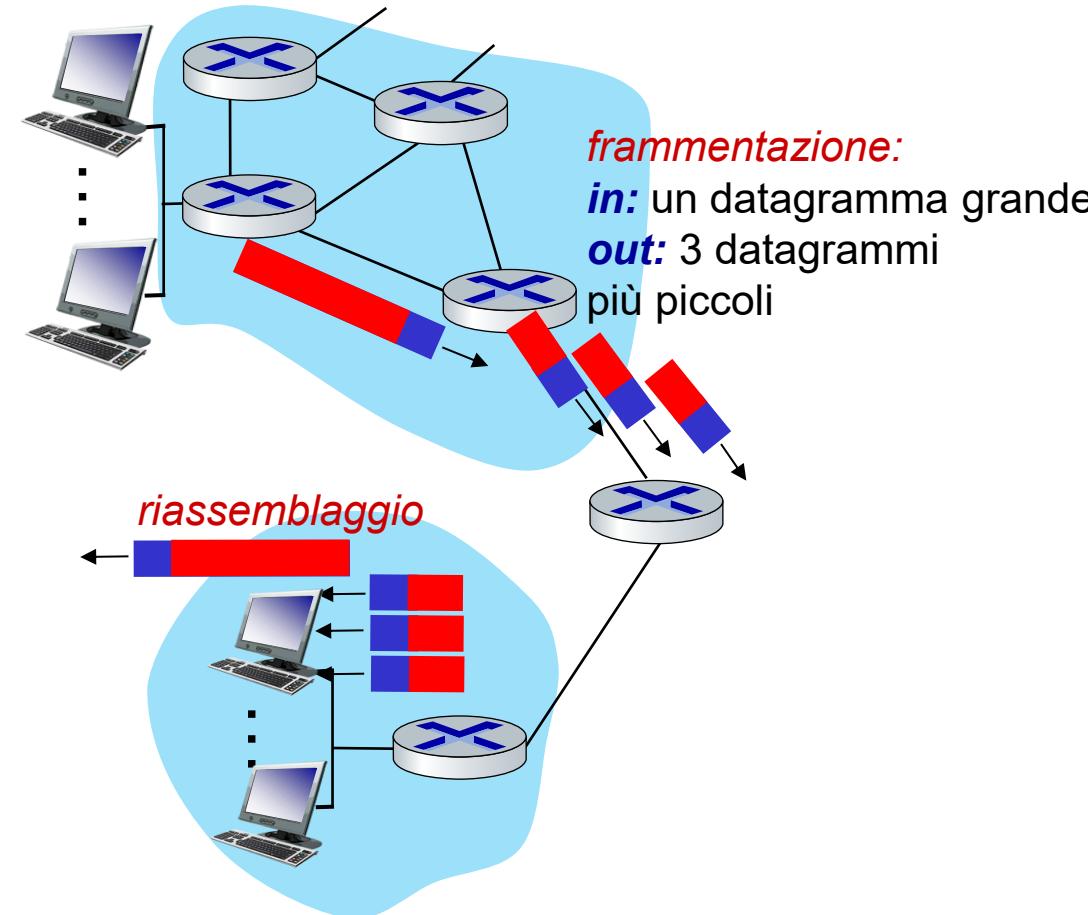
indirizzo IP sorgente a 32 bit

indirizzo IP destinazione a 32 bit

es., registrazione del
percorso di un
datagramma o di
timestamp,
determinazione del
percorso (o almeno parte
di esso) da parte del
mittente

Frammentazione dei datagrammi IP

- L'unità massima di trasmissione (MTU) è la massima quantità di dati che un frame a livello di collegamento può trasportare
 - Differenti tipi di collegamento, differenti MTU
- Datagrammi IP grandi vengono suddivisi ("frammentati") in datagrammi IP più piccoli
 - un datagramma viene frammentato
 - i frammenti saranno "riassemblati" solo una volta raggiunta la *destinazione*
 - i bit dell'intestazione IP sono usati per identificare e ordinare i frammenti



Frammentazione e riassemblaggio IP

- bit 2: Riservato; deve essere 0
- bit 1: Don't Fragment (DF)
- bit 0: More Fragments (MF)

Esempio:

- Datagramma di 4000 byte
- MTU = 1500 byte

lunghez.	ID	fragflag	offset	
=4000	=x	=0	=0	

Un datagramma IP grande viene frammentato in datagrammi IP più piccoli

1480 byte nel campo dati

La lunghezza dei dati in ciascun frammento tranne l'ultimo deve essere multiplo di 8

offset =
 $1480/8$

	length	ID	fragflag	offset	
	=1500	=x	=1	=0	

	length	ID	fragflag	offset	
	=1500	=x	=1	=185	

	length	ID	fragflag	offset	
	=1040	=x	=0	=370	

unico per una combinazione di indirizzo IP sorgente e destinazione e protocollo

Frammentazione e riassemblaggio IP

- Deprecato, rimosso in IPv6
- *Path MTU Discovery*
 - invio di pacchetti con bit (DF) Don't Fragment impostato a 1
 - se il router non può inoltrare il datagramma perché eccede la MTU, scarta il pacchetto e invia al mittente un messaggio ICMP "Destination Unreachable: Fragmentation Required"

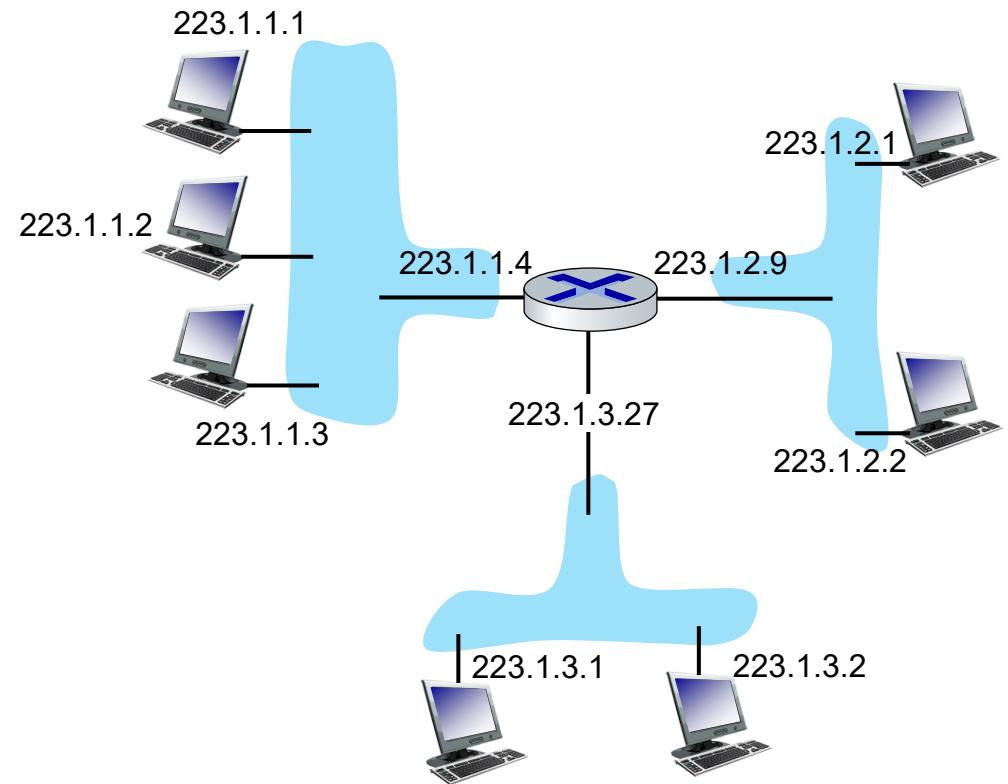
Il problema è che questi messaggi ICMP possono essere bloccati (per motivi di sicurezza): in questi casi, per esempio, un mittente TCP rischia addirittura di ritrasmettere inutilmente lo pacchetto più volte! Inoltre, il percorso e quindi la MTU possono cambiare!

Sono stati proposti approcci alternativi più robusti.

Tra le alternative: manipolazione di segmenti SYN in fase di instaurazione di una connessione TCP, cambiando l'opzione relativa al MSS.

Indirizzamento IP: introduzione

- **Indirizzo IP:** identificatore a 32 bit associato a ciascuna *interfaccia* di host e router
- **interfaccia:** connessione tra host/router e collegamento fisico
 - i router hanno tipicamente più interfacce
 - gli host hanno tipicamente una o due interfacce (es., Ethernet cablata, 802.11 wireless)



notazione decimale puntata (*dotted-decimal notation*):

223.1.1.1 =

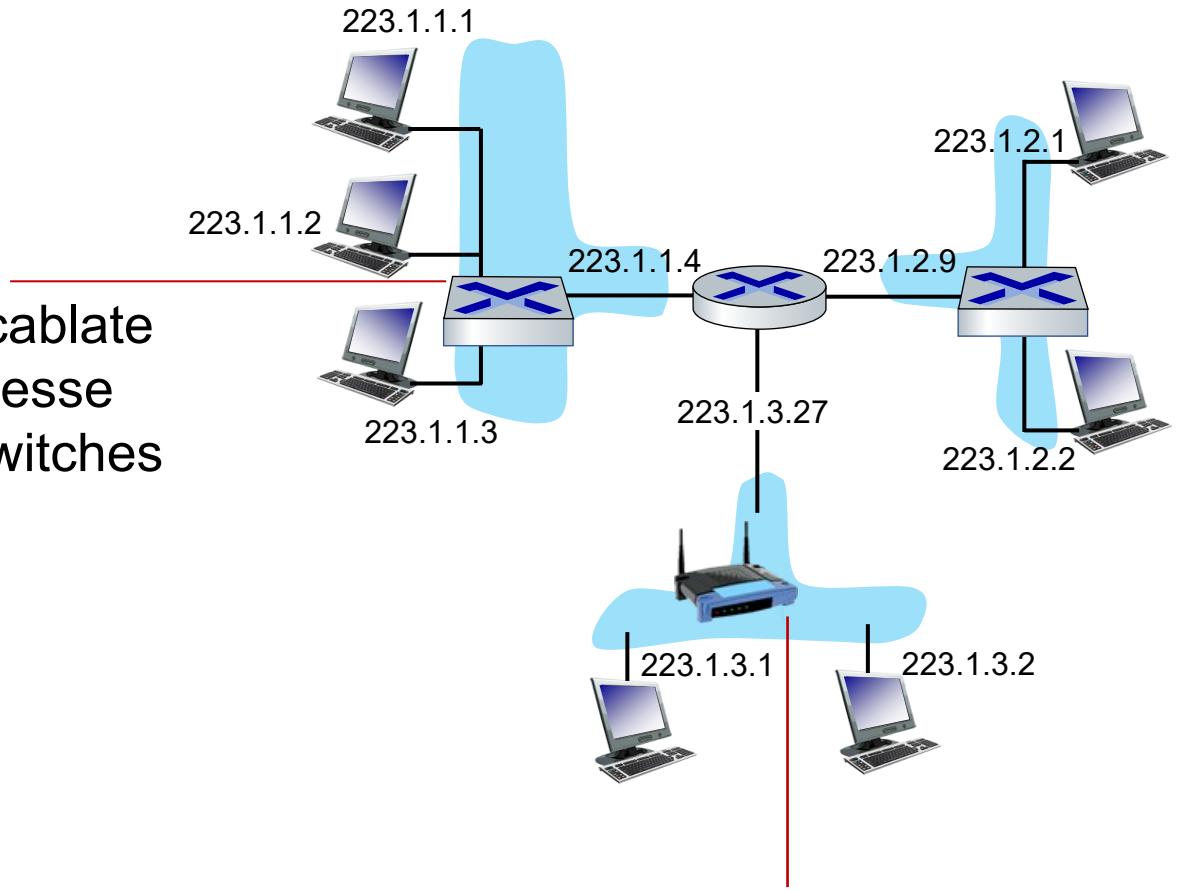
11011111	00000001	00000001	00000001
223	1	1	1

Indirizzamento IP: introduzione

D: come sono effettivamente collegate le interfacce?

R: interfacce cablate Ethernet connesse da Ethernet switches

Per ora: non c'è bisogno di preoccuparsi di come una interfaccia sia connessa a un'altra (senza l'intervento di alcun router)



R: interfacce wireless WiFi connesse da stazioni base WiFi

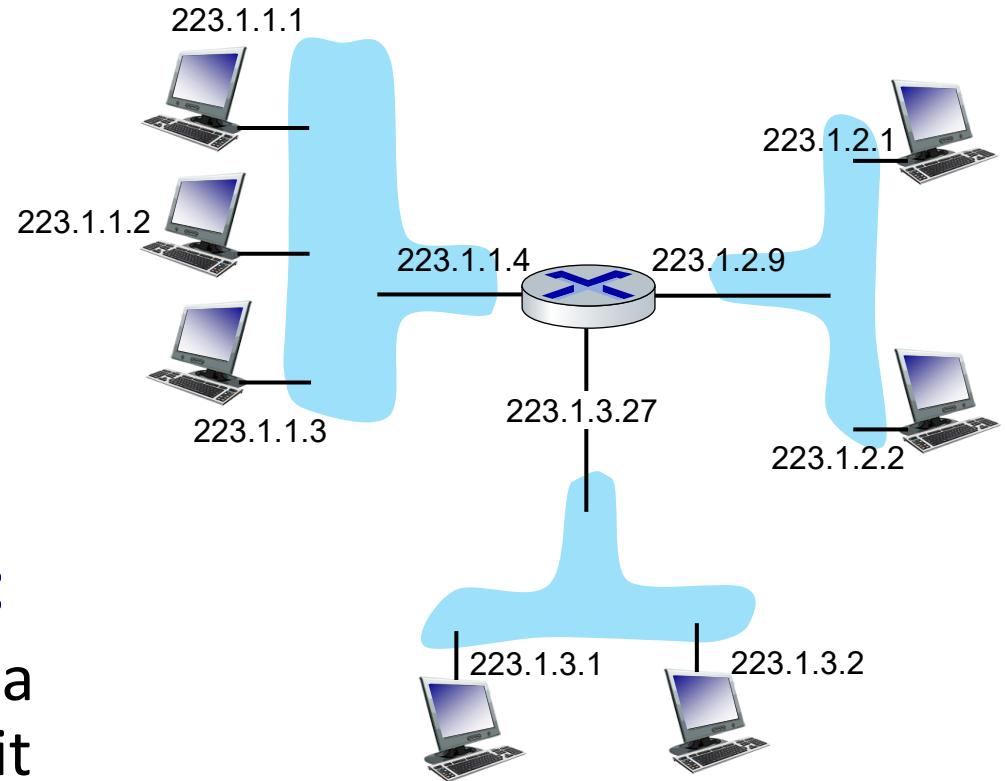
Sottoreti (subnet)

■ Cos'è una sottorete?

- Interfacce di dispositivi che possono raggiungersi fisicamente **senza passare attraverso un router intermedio**

■ Gli indirizzi IP hanno una struttura:

- **parte della sottorete:** i dispositivi della stessa sottorete hanno in comune i bit di ordine superiore
- **Parte dell'host:** i rimanenti bit di ordine inferiore

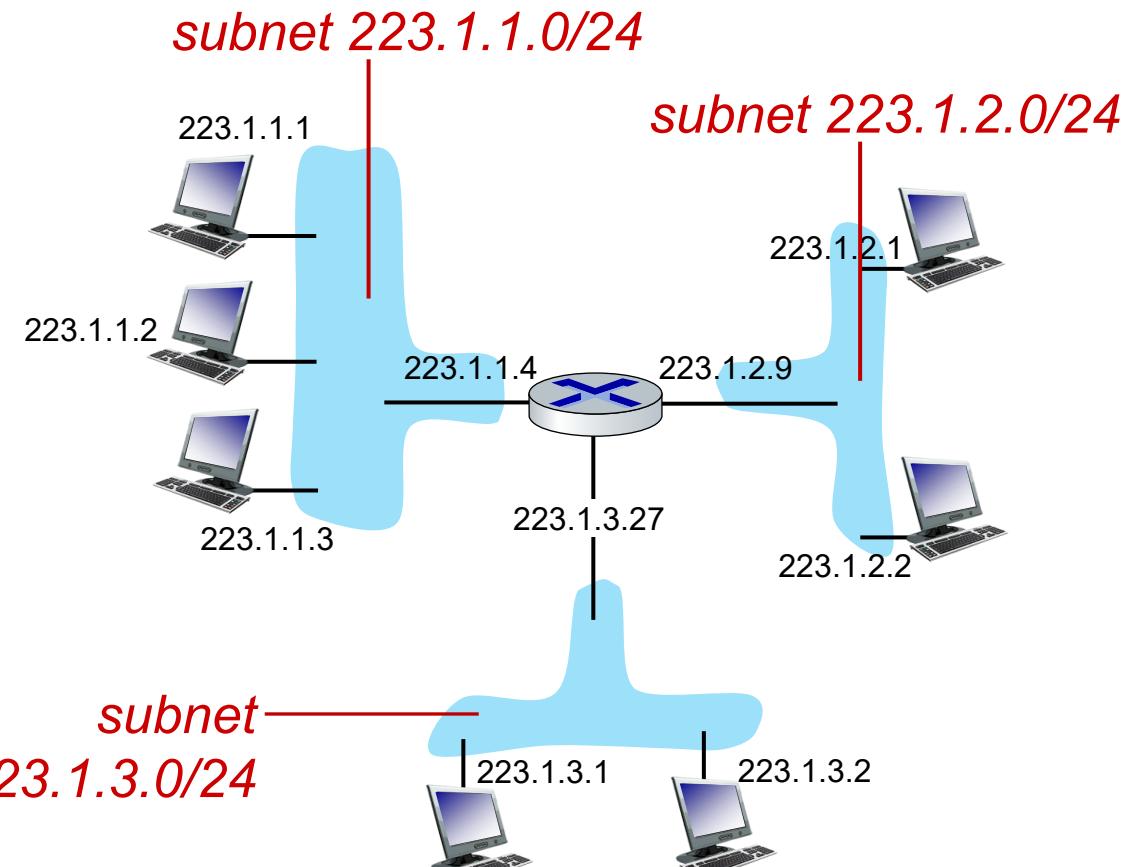


rete composta da 3 sottoreti

Sottoreti (subnet)

Procedura per definire le sottoreti:

- si sgancino le interfacce da host e router in maniera tale da creare "isole" di reti isolate delimitate dalle interfacce
- ognuna di queste reti isolate viene detta *sottorete* (subnet)

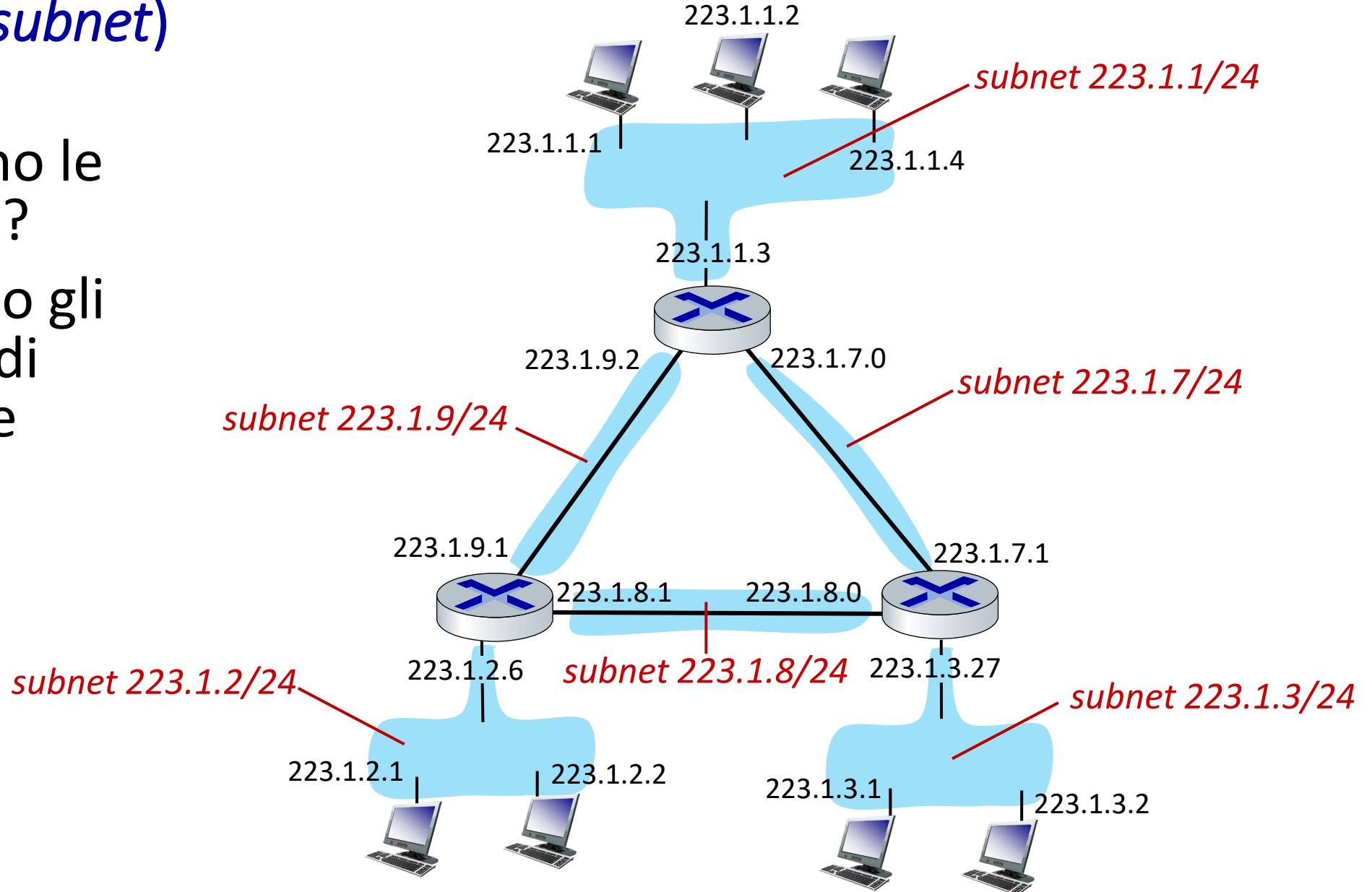


subnet
223.1.3.0/24

maschera di sottorete (subnet mask): /24
(24 bit di ordine superiore: parte di sottorete
dell'indirizzo IP)

Sottoreti (*subnet*)

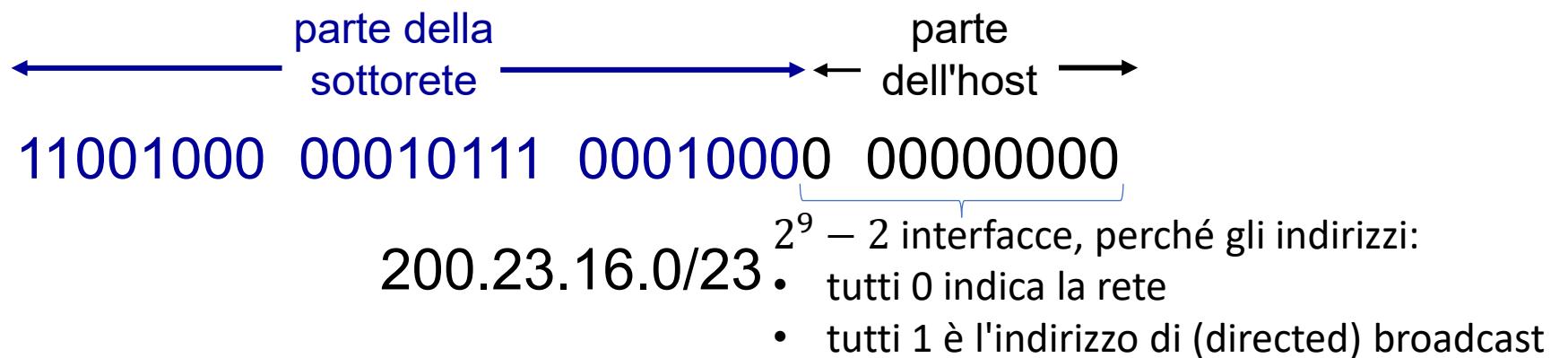
- dove sono le sottoreti?
- cosa sono gli indirizzi di sottorete /24?



Indirizzamento IP: CIDR

CIDR: Classless InterDomain Routing (pronounced “cider”)

- parte della sottorete dell'indirizzo di lunghezza arbitraria
 - formato dell'indirizzo: **a.b.c.d/x**, dove x è il numero di bit della porzione di sottorete dell'indirizzo



Esiste un altro tipo di broadcast, detto limited, (255.255.255.255) che corrisponde al broadcast L2

Indirizzamento IP: classful addressing

Spazio di indirizzamento IPv4 suddiviso in blocchi con prefisso di rete di 8, 16 e 24 bit

Classe	Bit iniziali	parte della sottorete	Parte dell'host	Numero di reti	Numero di indirizzi per rete	Numero totale di indirizzi	Indirizzo iniziale	Indirizzo finale	Maschera di rete in dot-decimal notation	Notazione CIDR
Classe A	0	8	24	128 (2^7)	16,777,216 (2^{24})	2^{31}	0.0.0.0	127.255.255.255	255.0.0.0	/8
Classe B	10	16	16	16,384 (2^{14})	65,536 (2^{16})	2^{30}	128.0.0.0	191.255.255.255	255.255.0.0	/16
Classe C	110	24	8	2,097,152 (2^{21})	256 (2^8)	2^{29}	192.0.0.0	223.255.255.255	255.255.255.0	/24
Classe D <i>(multicast)</i>	1110	non definita	non definita	non definito	non definito	2^{28}	224.0.0.0	239.255.255.255	non definita	/4
Classe E <i>(reserved)</i>	1111	non definita	non definita	non definito	non definito	2^{28}	240.0.0.0	255.255.255.255	non definita	non definita

Indirizzamento IP: classful addressing

- L'indirizzamento per classi è stato ormai abbandonato, in favore di CIDR
- CIDR ha diversi vantaggi:
 - più efficiente allocazione di blocchi di indirizzi
 - aggregazione di indirizzi (vedi dopo) con conseguente riduzione delle tabelle di instradamento

Università degli Studi di Roma "Tor Vergata"
Laurea in Informatica

Sistemi Operativi e Reti
(modulo Reti)
a.a. 2024/2025

Livello di rete: piano dei dati (parte2)

dr. Manuel Fiorelli

manuel.fiorelli@uniroma2.it

<https://art.uniroma2.it/fiorelli>

Basate sulle slide del libro di testo:

https://gaia.cs.umass.edu/kurose_ross/ppt.php

Introduction: 1-58

Indirizzi IP: come ottenerne uno?

In realtà si tratta di **due** domande:

1. D: Come fa un *host* a ottenere l'indirizzo IP all'interno della sua rete (parte host dell'indirizzo)?
2. D: Come fa una *rete* a ottenere l'indirizzo IP (parte dell'indirizzo relativa alla rete)?

Come *l'host* ottiene l'indirizzo IP?

- codificato dal sysadmin nel file di configurazione
- **DHCP: Dynamic Host Configuration Protocol:** permette a un host di ottenere un indirizzo IP in modo automatico
 - “plug-and-play”

DHCP: Dynamic Host Configuration Protocol

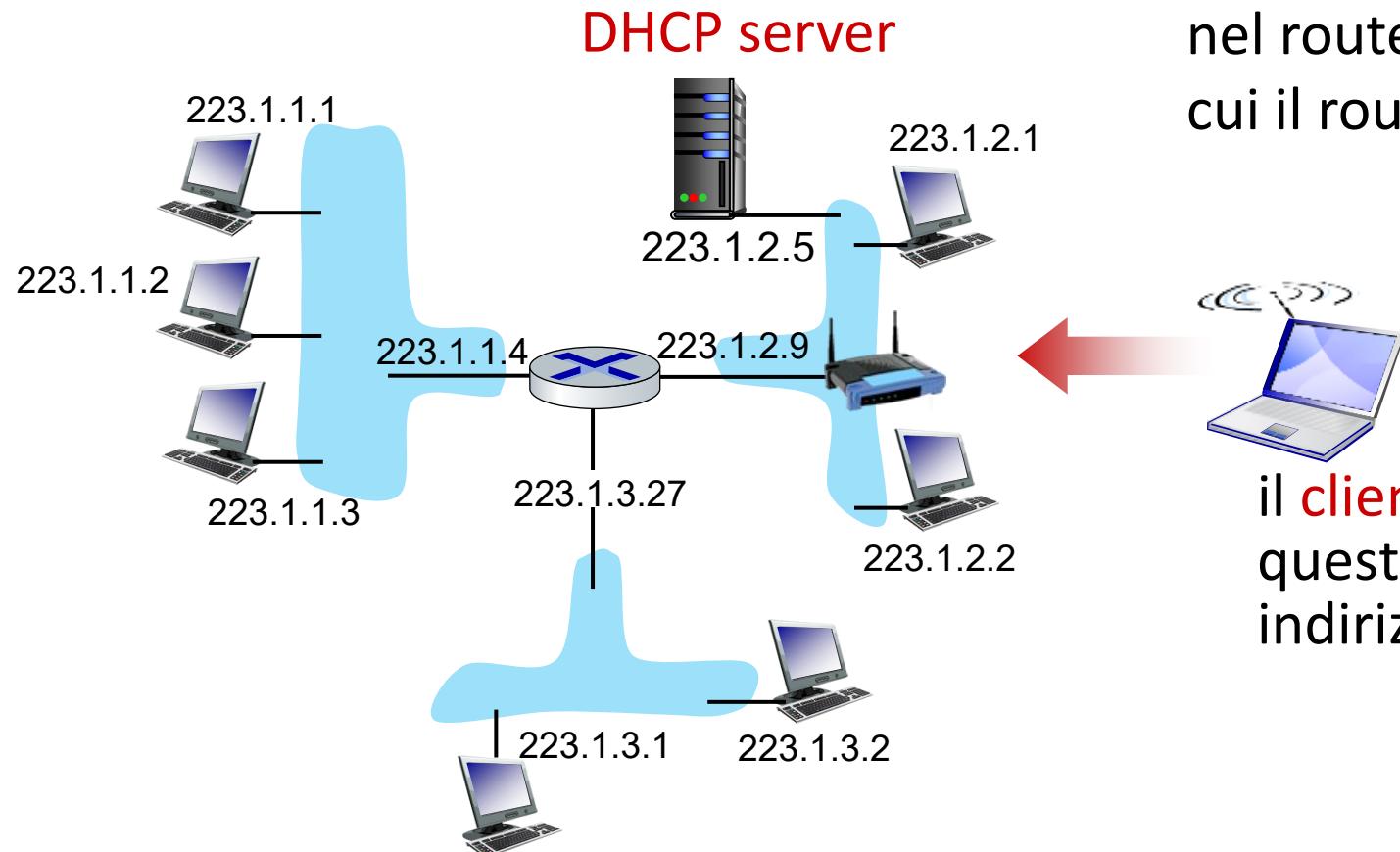
obiettivo: l'host ottiene *dinamicamente* l'indirizzo IP dal server di rete quando si "unisce" alla rete.

- può rinnovare la propria concessione per l'indirizzo in uso
- permette il riutilizzo degli indirizzi (mantiene l'indirizzo solo quando è collegato/acceso)
- supporto per gli utenti mobili che si uniscono/abbandonano la rete (ma non permette il mantenimento di una connessione TCP attiva, perché quando ci si unisce a una nuova sottorete si ottiene un indirizzo IP differente)

Panoramica di DHCP

- l'host invia in broadcast un messaggio **DHCP discover** [opzionale]
- il server DHCP risponde con messaggio **DHCP offer** [opzionale]
- l'host richiede un indirizzo IP: messaggio **DHCP request**
- il server DHCP invia un indirizzo IP: messaggio **DHCP ack**

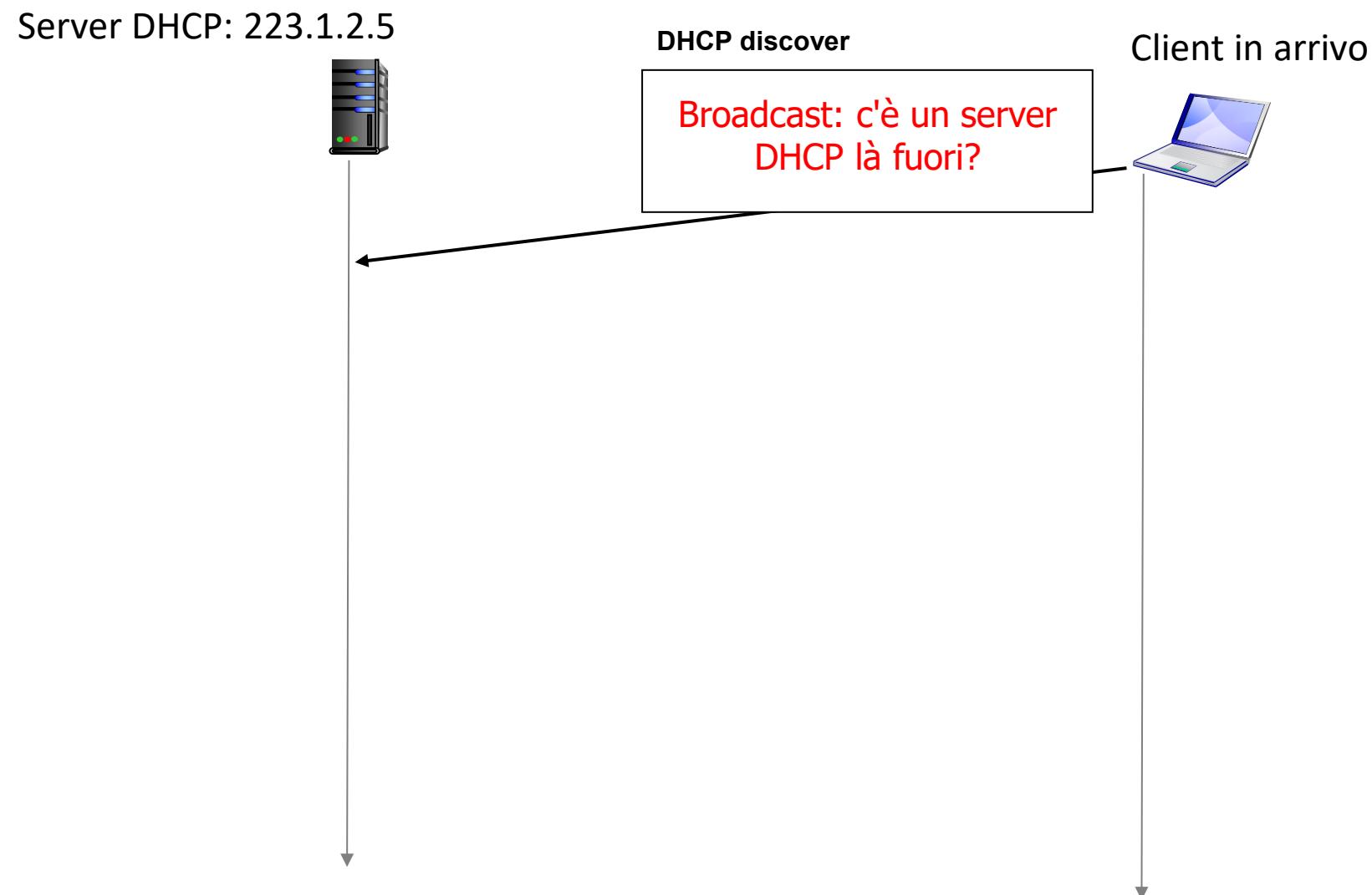
DHCP client-server scenario



In genere, il server DHCP è collocato nel router e serve tutte le sottoreti a cui il router è collegato.

il **client DHCP** in arrivo su questa rete ha bisogno di indirizzo

DHCP client-server scenario



DHCP client-server scenario

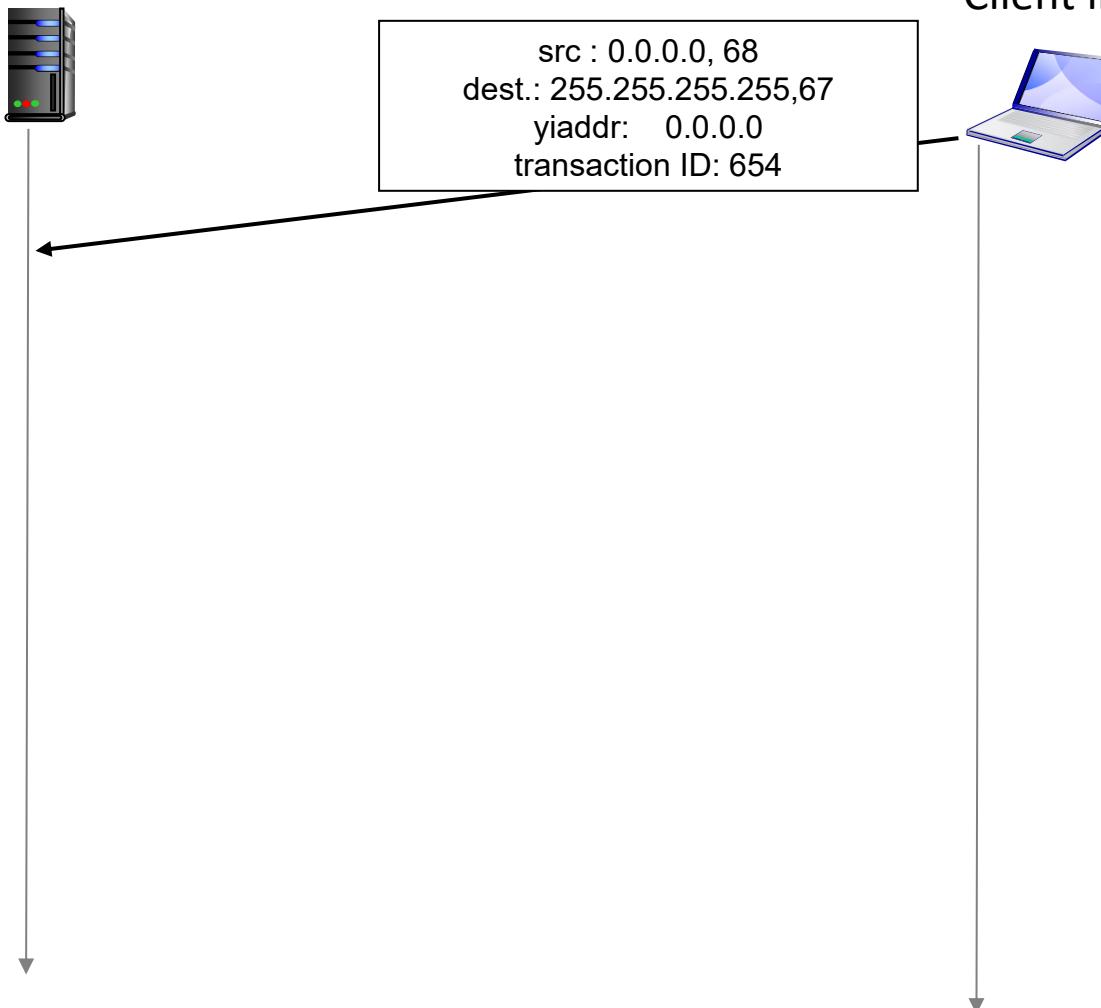
Server DHCP: 223.1.2.5



DHCP discover

```
src : 0.0.0.0, 68  
dest.: 255.255.255.255,67  
yiaddr: 0.0.0.0  
transaction ID: 654
```

Client in arrivo



DHCP client-server scenario

Server DHCP: 223.1.2.5



DHCP discover

src : 0.0.0.0, 68
dest.: 255.255.255.255,67
yiaddr: 0.0.0.0
transaction ID: 654

Client in arrivo



DHCP offer

Broadcast: Sono un server
DHCP. Questo è un indirizzo
IP che puoi usare

DHCP client-server scenario

Server DHCP: 223.1.2.5



DHCP discover

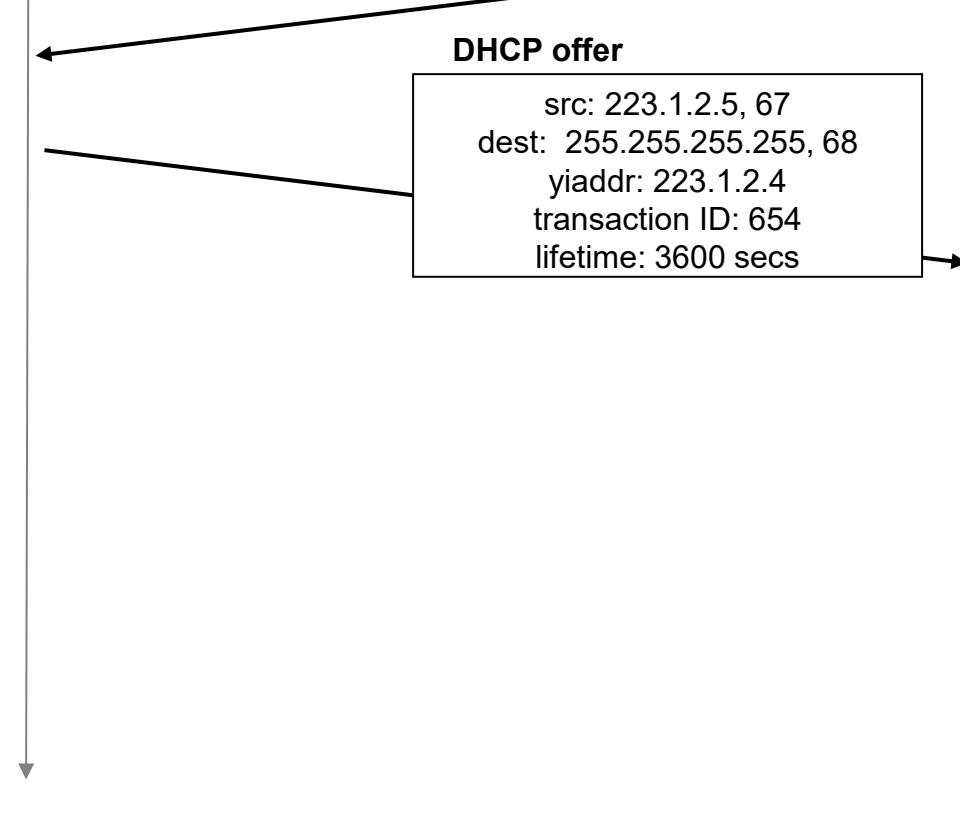
```
src : 0.0.0.0, 68  
dest.: 255.255.255.255,67  
yiaddr: 0.0.0.0  
transaction ID: 654
```

Client in arrivo



DHCP offer

```
src: 223.1.2.5, 67  
dest: 255.255.255.255, 68  
yiaddr: 223.1.2.4  
transaction ID: 654  
lifetime: 3600 secs
```



DHCP client-server scenario

Server DHCP: 223.1.2.5



DHCP discover

```
src : 0.0.0.0, 68  
dest.: 255.255.255.255,67  
yiaddr: 0.0.0.0  
transaction ID: 654
```

Client in arrivo



DHCP offer

```
src: 223.1.2.5, 67  
dest: 255.255.255.255, 68  
yiaddr: 223.1.2.4  
transaction ID: 654  
lifetime: 3600 secs
```

DHCP request

Broadcast: OK. Voglio
usare questo indirizzo IP!

I due passaggi precedenti
possono essere saltati "se
un client si ricorda e
desidera riutilizzare un
indirizzo di rete
precedentemente
assegnato".
[RFC 2131]

DHCP client-server scenario

Server DHCP: 223.1.2.5



DHCP discover

```
src : 0.0.0.0, 68  
dest.: 255.255.255.255,67  
yiaddr: 0.0.0.0  
transaction ID: 654
```

Client in arrivo



DHCP offer

```
src: 223.1.2.5, 67  
dest: 255.255.255.255, 68  
yiaddr: 223.1.2.4  
transaction ID: 654  
lifetime: 3600 secs
```

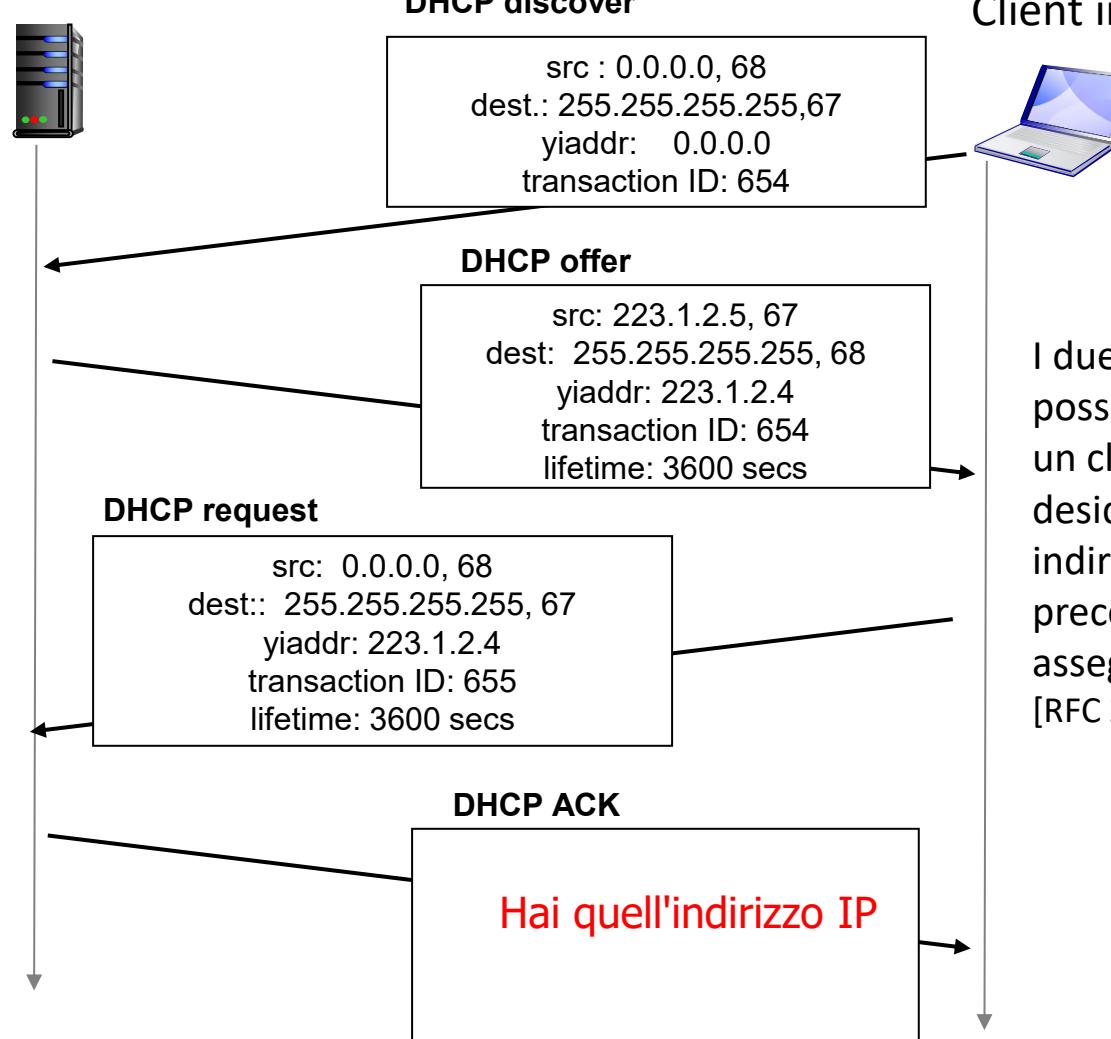
DHCP request

```
src: 0.0.0.0, 68  
dest: 255.255.255.255, 67  
yiaddr: 223.1.2.4  
transaction ID: 655  
lifetime: 3600 secs
```

I due passaggi precedenti
possono essere saltati "se
un client si ricorda e
desidera riutilizzare un
indirizzo di rete
precedentemente
assegnato".
[RFC 2131]

DHCP client-server scenario

Server DHCP: 223.1.2.5

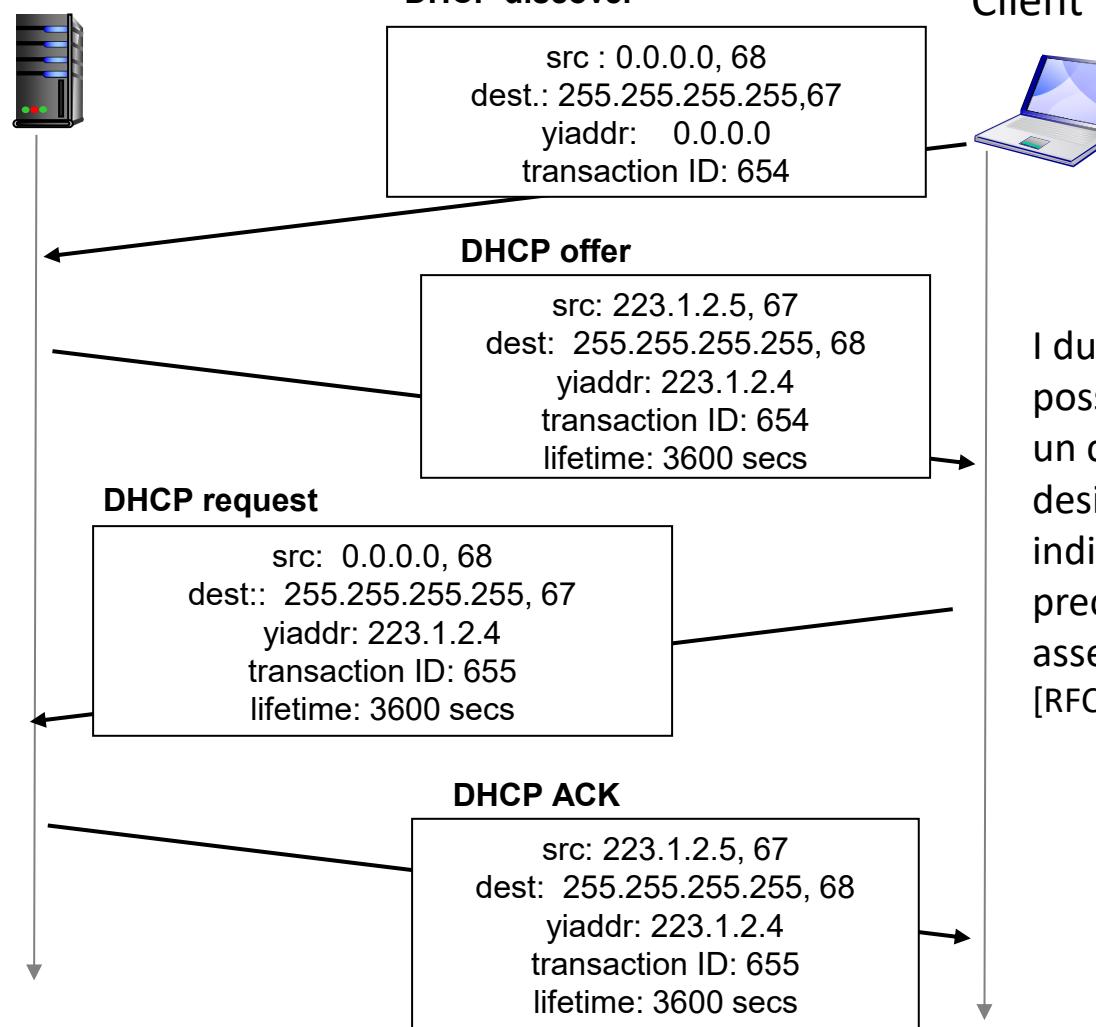


Client in arrivo

I due passaggi precedenti
possono essere saltati "se
un client si ricorda e
desidera riutilizzare un
indirizzo di rete
precedentemente
assegnato".
[RFC 2131]

DHCP client-server scenario

Server DHCP: 223.1.2.5



Client in arrivo

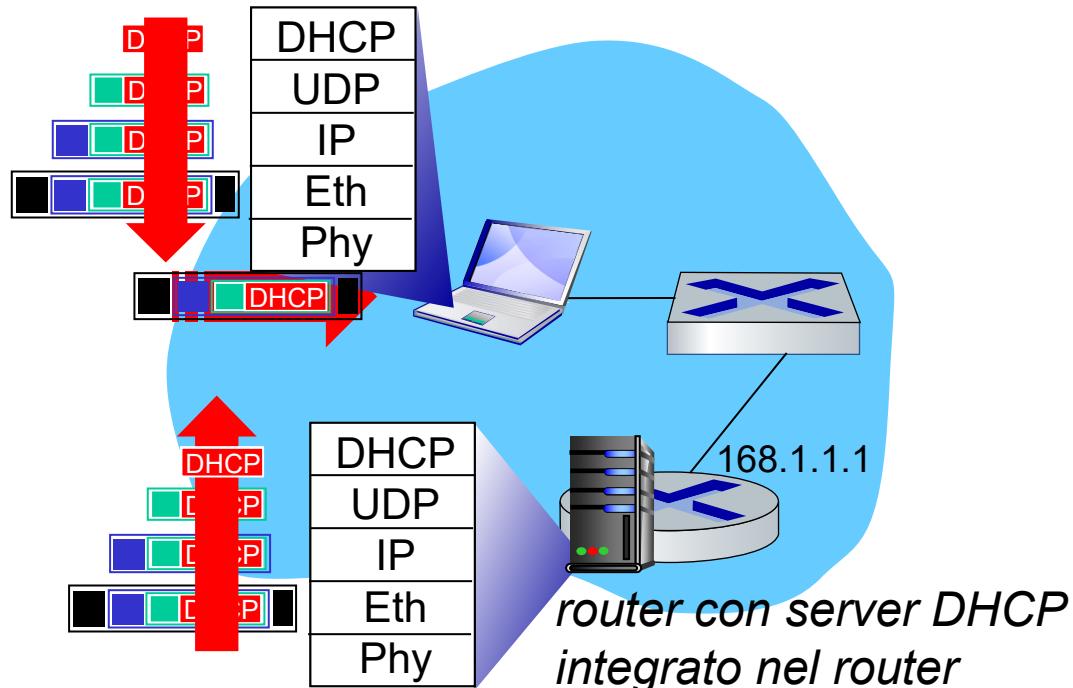
I due passaggi precedenti
possono essere saltati "se
un client si ricorda e
desidera riutilizzare un
indirizzo di rete
precedentemente
assegnato".
[RFC 2131]

DHCP: non solo indirizzi IP

Il DHCP può restituire più di un indirizzo IP assegnato alla sottorete:

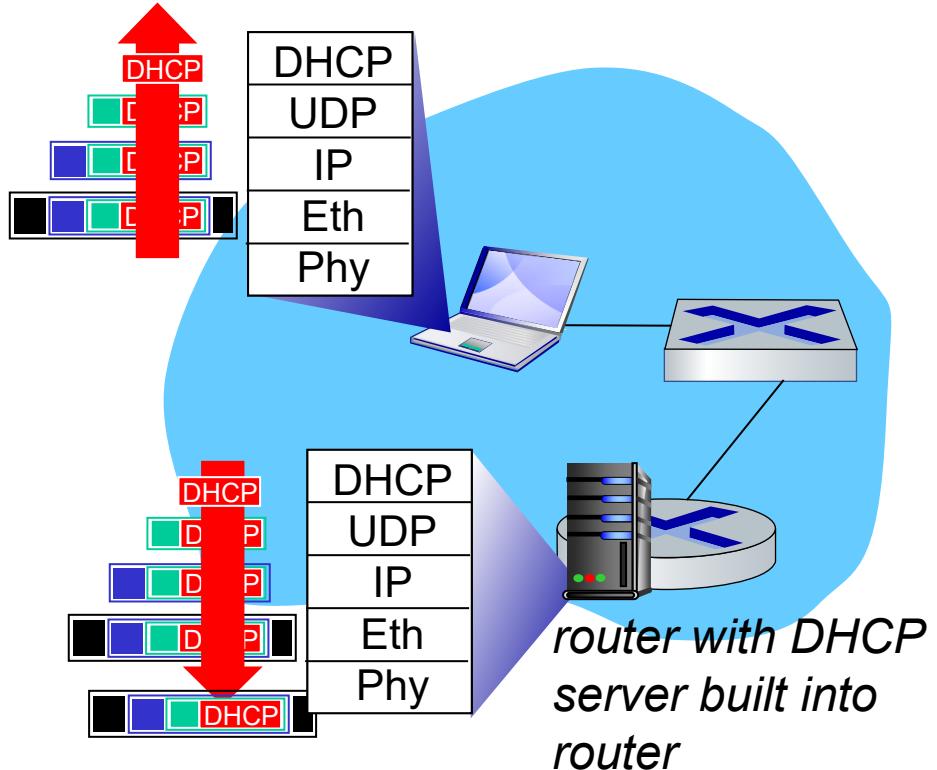
- indirizzo del router first-hop o router (o gateway) predefinito (per comunicare al di là della sottorete)
- nome e indirizzo IP del server DNS
- maschera di rete (che indica la porzione di rete rispetto a quella di host dell'indirizzo)

DHCP: esempio



- Il portatile che si collega utilizzerà il DHCP per ottenere l'indirizzo IP, l'indirizzo del router first-hop e l'indirizzo del server DNS.
- Messaggio di richiesta DHCP encapsulato in UDP, encapsulato in IP, encapsulato in Ethernet
- Trasmissione di frame Ethernet (destinazione: `FFFFFFFFFF`) sulla LAN, ricevuto dal router che esegue il server DHCP
- Ethernet demultiplato in IP, IP demultiplato in UDP, UDP demultiplato in DHCP

DHCP: esempio



- Il server DHCP formula un DHCP ACK contenente l'indirizzo IP del client, l'indirizzo IP del router first-hop per il client, il nome e l'indirizzo IP del server DNS.
- risposta del server DHCP incapsulata inoltrata al client, de-muxing fino a DHCP sul client
- il cliente conosce ora il proprio indirizzo IP, il nome e l'indirizzo IP del server DNS, l'indirizzo IP del router first-hop

Indirizzi IP: come ottenerne uno?

D: Come fa la rete a ottenere la parte di sottorete dell'indirizzo IP?

R: ottiene l'assegnazione di una porzione dello spazio di indirizzi del suo provider ISP

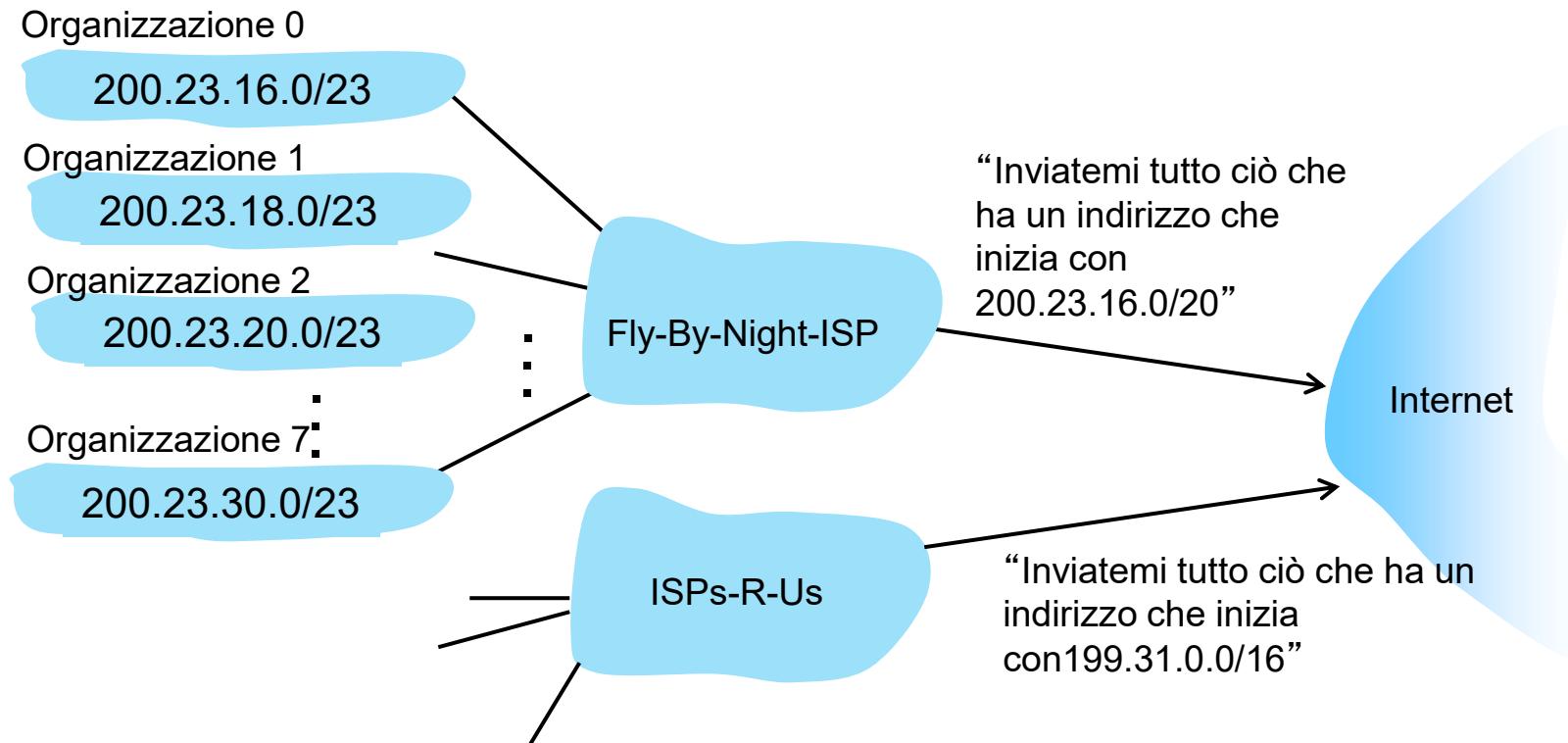
Blocco dell'ISP	11001000	00010111	00010000	00000000	200.23.16.0/20
-----------------	----------	----------	----------	----------	----------------

L'ISP può quindi allocare il suo spazio di indirizzi in 8 blocchi:

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

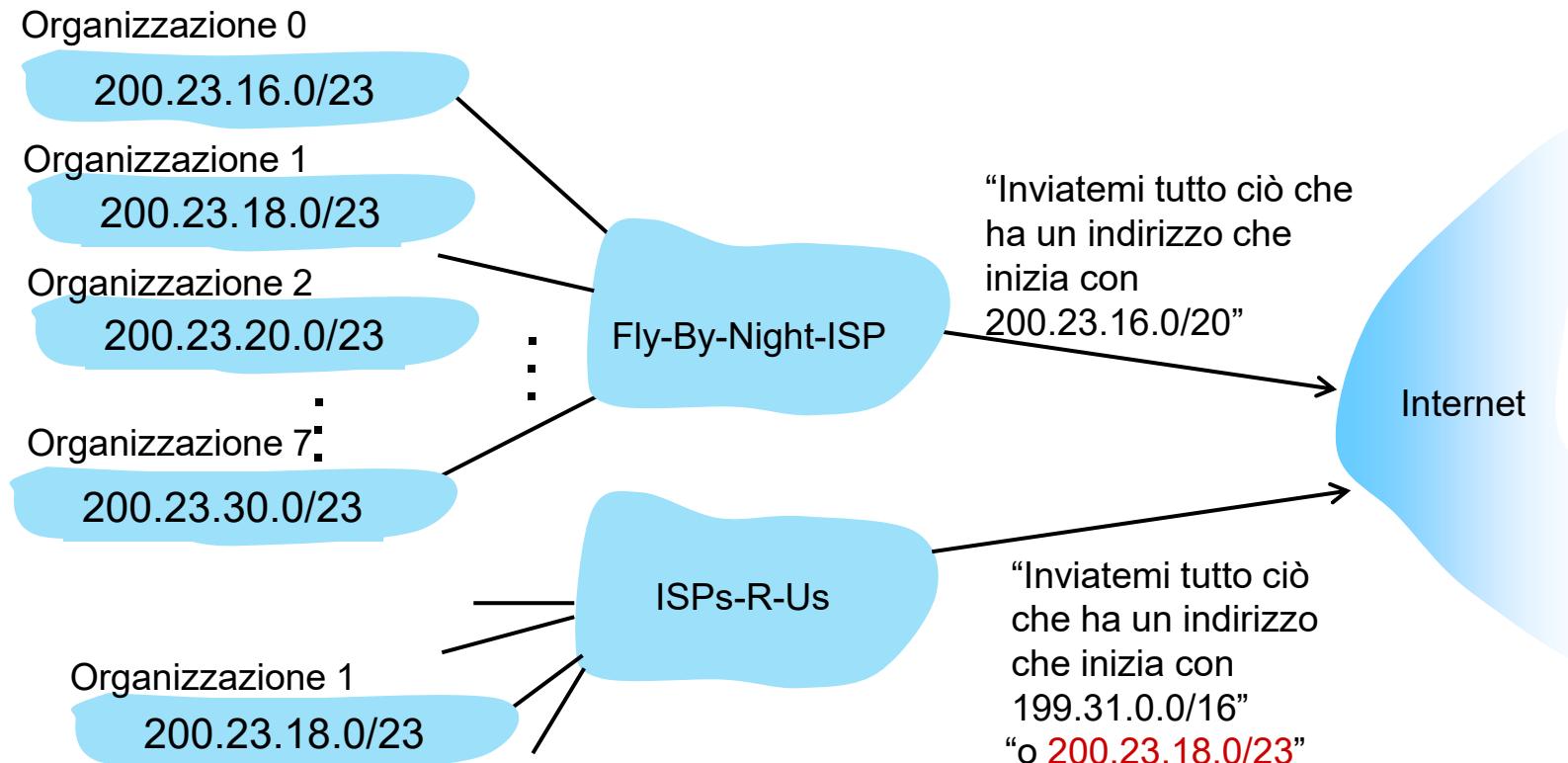
Indirizzamento gerarchico: aggregazione di indirizzi (route aggregation)

L'indirizzamento gerarchico consente di pubblicizzare in modo efficiente le informazioni di routing:



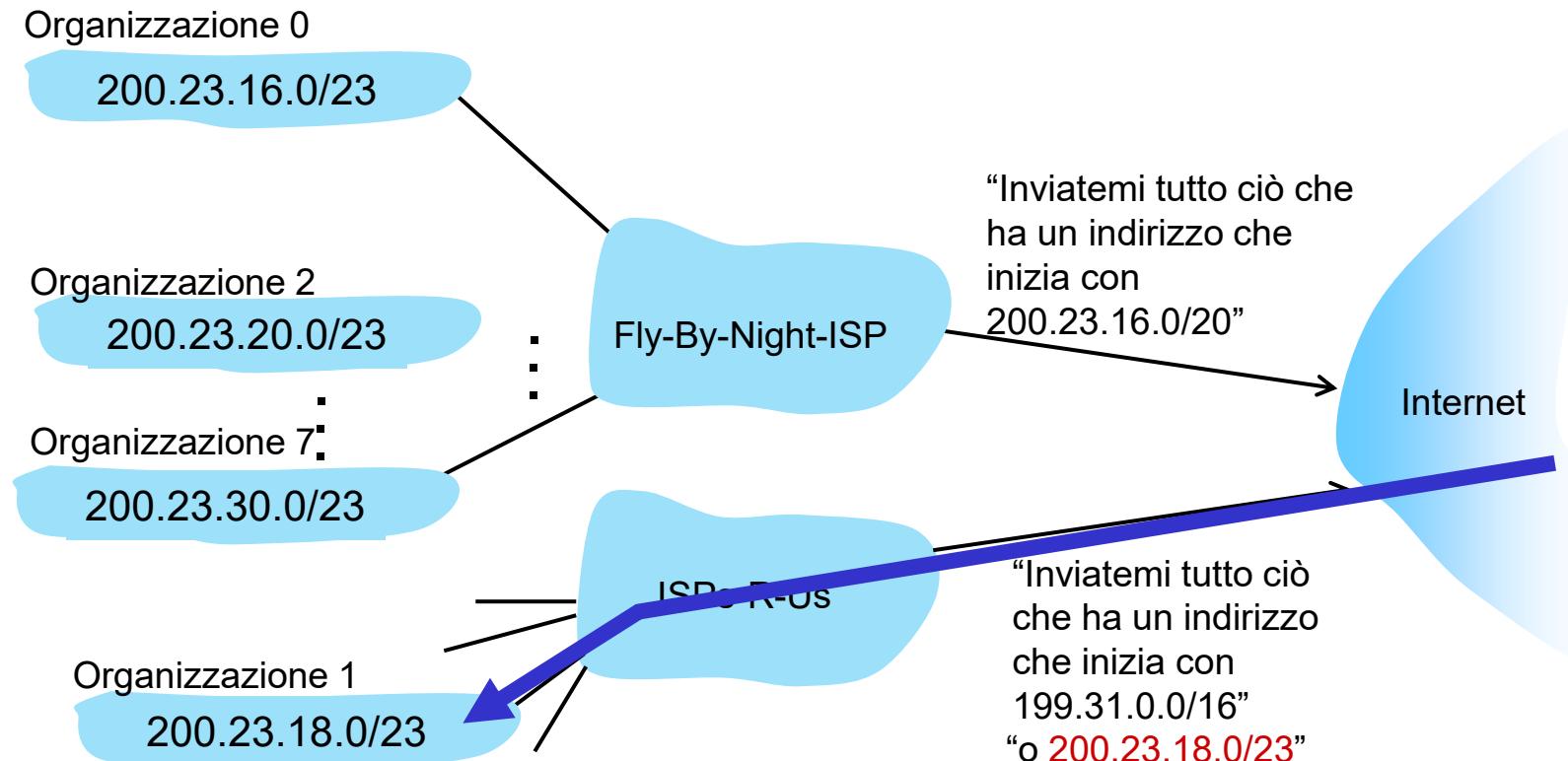
Indirizzamento gerarchico: percorsi più specifici

- L'organizzazione 1 si sposta da Fly-By-Night-ISP a ISP-R-Us
- ISP-R-Us ora pubblicizza un percorso più specifico verso l'Organizzazione 1



Indirizzamento gerarchico: percorsi più specifici

- L'organizzazione 1 si sposta da Fly-By-Night-ISP a ISP-R-Us
- ISP-R-Us ora pubblicizza un percorso più specifico verso l'Organizzazione 1



Indirizzamento IP: ultime parole...

D: Come fa un ISP a ottenere un blocco di indirizzi?

R: ICANN: Internet Corporation for Assigned Names and Numbers

<http://www.icann.org/>

- Assegnazione degli indirizzi IP, attraverso **5 registri regionali (RR)** (che possono poi assegnare ai registri locali).
- Gestisce la zona radice del DNS, compresa la delega della gestione dei singoli TLD (.com, .edu , ...)

D: ci sono abbastanza indirizzi IP a 32 bit?

- L'ICANN ha assegnato l'ultima porzione di indirizzi IPv4 ai RR nel 2011.
- NAT (successivo) aiuta con l'esaurimento dello spazio degli indirizzi IPv4.
- IPv6 ha uno spazio di indirizzi a 128 bit

"Who the hell knew how much address space we needed?" Vint Cerf (riflettere sulla decisione di rendere l'indirizzo IPv4 lungo 32 bit)

Livello di rete: tabella di marcia sul “piano dei dati”

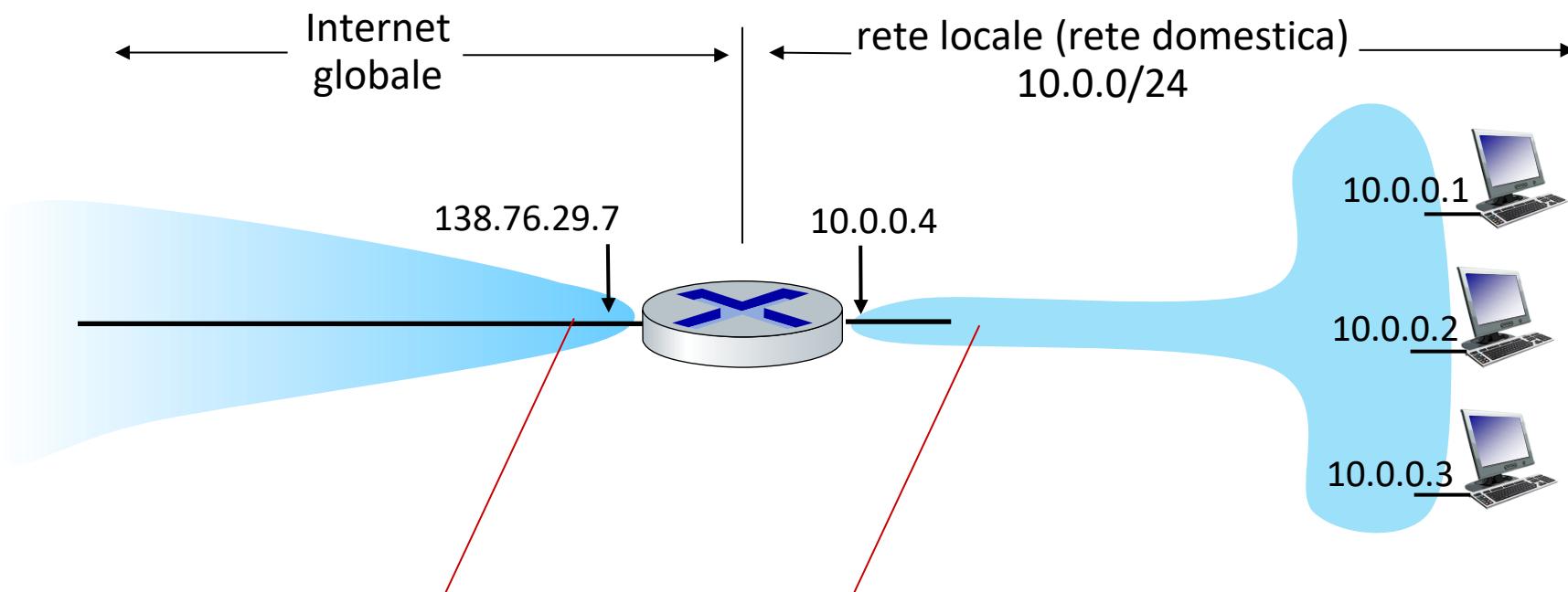
- Livello di rete: panoramica
 - piano dei dati
 - piano di controllo
- Cosa c'è dentro un router
 - Porte di ingresso, struttura di commutazione, porte di uscita
 - buffer management, scheduling
- IP: il Protocollo Internet
 - Formato dei datagrammi
 - indirizzamento
 - Traduzione degli indirizzi di rete
 - IPv6



- Inoltro generalizzato, SDN
 - Match+action
 - OpenFlow: match+action in azione
- Middlebox

NAT: network address translation

NAT: Tutti i dispositivi della rete locale condividono un solo indirizzo IPv4 per il mondo esterno



tutti i datagrammi che *escono* dalla rete locale hanno lo *stesso* indirizzo IP sorgente: 138.76.29.7, ma *differenti* numeri di porta sorgente

I datagrammi con sorgente/destinazione in questa rete hanno indirizzo 10.0.0/24 per la sorgente e la destinazione (come al solito)

NAT: network address translation

- tutti i dispositivi della rete locale hanno indirizzi a 32 bit in uno spazio di indirizzi IP "privato" (prefissi 10/8, 172.16/12, 192.168/16) che possono essere utilizzati solo nella rete locale
- vantaggi:
 - è necessario **un solo** indirizzo IP dal provider ISP per *tutti* i dispositivi
 - può cambiare gli indirizzi degli host nella rete locale senza notificare il mondo esterno
 - può cambiare ISP senza modificare gli indirizzi dei dispositivi nella rete locale
 - sicurezza: dispositivi all'interno della rete locale non direttamente indirizzabili, visibili dall'esterno

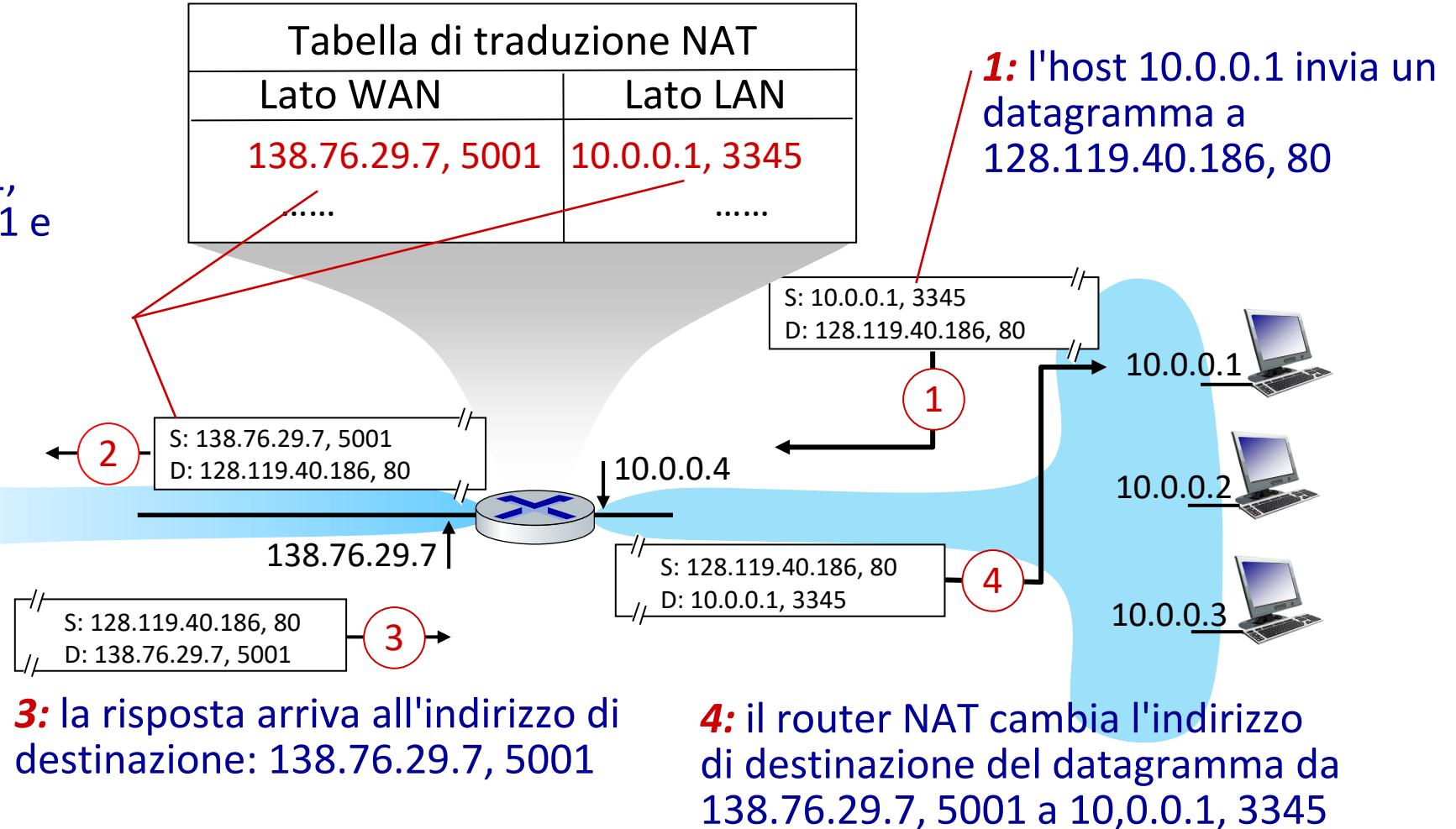
NAT: network address translation

implementazione: i router NAT devono (in maniera trasparente):

- **datagrammi in uscita: sostituire** (indirizzo IP sorgente, n. porta sorgente) di ogni datagramma in uscita con (indirizzo IP NAT, nuovo n. porta)
 - i client/server remoti risponderanno con (indirizzo IP NAT, nuovo n. porta) come indirizzo di destinazione
- ricordare (nella "Tabella di traduzione NAT") ogni coppia di traduzione da (indirizzo IP sorgente, n. porta) a (indirizzo IP NAT, nuovo n. porta)
- **Datagrammi in ingresso: sostituire** (indirizzo IP NAT, nuovo n. porta) nei campi di destinazione di ogni datagramma in ingresso con il corrispondente (indirizzo IP NAT, nuovo n. porta) memorizzato nella tabella NAT

NAT: network address translation

2: Il router NAT cambia l'indirizzo di origine del datagramma da 10.0.0.1, 3345 a 138.76.29.7, 5001 e aggiorna la tabella



NAT: network address translation

- Il NAT è oggetto di controversie:
 - i router “dovrebbero” elaborare i pacchetti solo fino al livello 3
 - la “scarsità” di indirizzi dovrebbe essere risolta da IPv6
 - viola il cosiddetto argomento punto-punto (numero di porta manipolato da un dispositivo a livello di rete)
 - attraversamento NAT (*NAT traversal*): cosa succede se un client vuole connettersi a un server dietro NAT?
- ma il NAT è qui per restare:
 - ampiamente utilizzato nelle reti domestiche e istituzionali, nelle reti cellulari 4G/5G

IPv6: motivazione

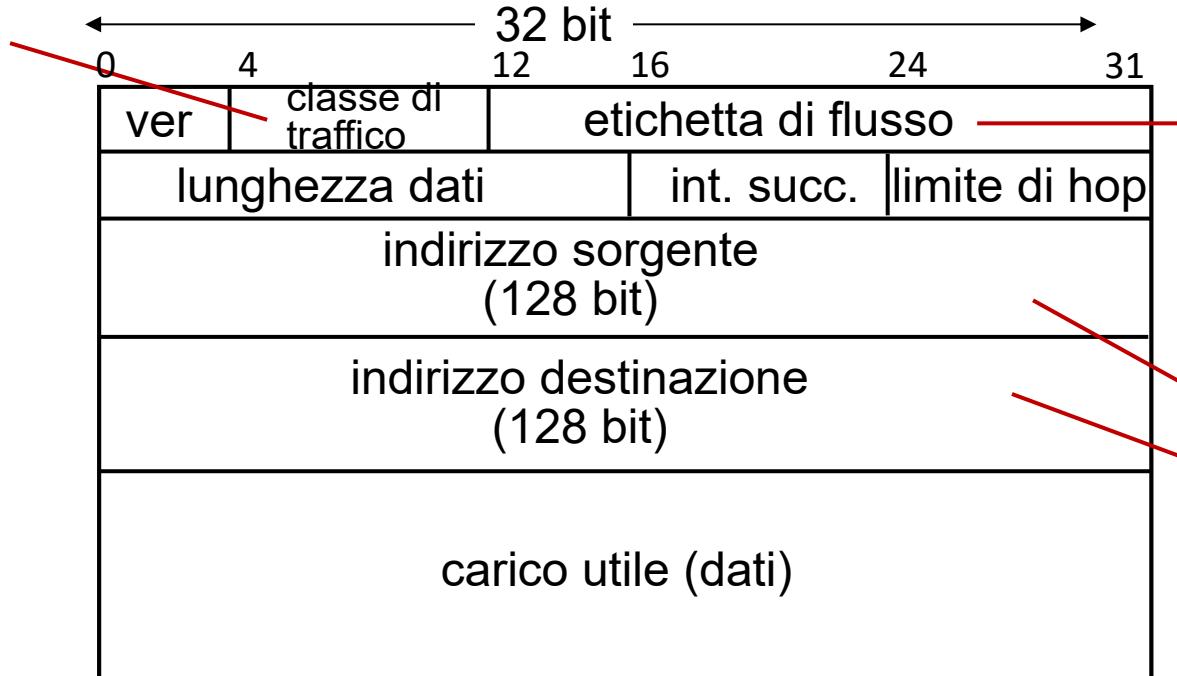
- **motivazione iniziale:** lo spazio degli indirizzi IPv4 a 32 bit sarebbe stato completamente allocato
- motivazioni aggiuntive:
 - velocità di elaborazione/inoltro: intestazione con una lunghezza fissa di 40 byte
 - consentire un diverso trattamento dei "flussi" a livello di rete (elevando il concetto di flusso al rango di *first-class citizen* mentre prima il focus era sui datagrammi)

Formato del datagramma IPv6

classe di traffico:

attribuisce priorità a datagrammi all'interno di un flusso o proveniente da specifiche applicazioni.

I 6 bit più significativi sono dedicati a DiffServ (per la classificazione e differenziazione del traffico) e i 2 bit meno significativi all'ECN



Etichetta di flusso:

identifica i datagrammi appartenenti allo stesso flusso (concetto di "flusso" non ben definito)

128-bit
indirizzi IPv6
(supporta unicast,
multicast [consegna a un
gruppo], anycast
[consegna al più vicino di
un gruppo])

Cosa manca (rispetto a IPv4):

- no checksum (per velocizzare l'elaborazione presso i router)
- no frammentazione/riassemblaggio (messaggio di errore ICMPv6 *Packet Too Big* con *MTU* del collegamento di uscita): in realtà, effettuato solo dal mittente e destinatario attraverso una *opzione*
- no opzioni (disponibile come "intestazione successiva" del protocollo di livello superiore)

Flussi IPv6

RFC 2460 a riguardo della etichettatura dei flussi:

l'etichettatura di pacchetti che appartengono a flussi particolari per i quali il mittente richiede una gestione speciale, come una qualità di servizio diversa da quella di default o un servizio in tempo reale”

Notazione degli indirizzi IPv6

RFC 4291: Gli indirizzi IPv6 (da 128 bit) sono scritti preferibilmente nella forma

x:x:x:x:x:x:x:x

dove le x rappresentano da 1 a 4 cifre esadecimali (pertanto, al più 16 bit).

2001:0db8:0000:0000:0000:8a2e:0370:7344

Sono possibili alcune abbreviazioni:

- gli zeri iniziali all'interno di ciascun campo possono essere omessi (ma ogni campo deve contenere almeno una cifra, ad eccezione di quanto detto nel punto successivo)

2001:db8:0:0:0:8a2e:370:7344

- una (e una sola!) sequenza di campi 0 contigui può essere abbreviata con il simbolo :: (il numero di campi compressi si determina per differenza rispetto a quelli scritti, nell'esempio è 3 = 8 - 5)

2001:db8::8a2e:370:7344

Notazione degli indirizzi IPv6

Un indirizzo IPv6 può avere diverse rappresentazioni testuali (es. in base a quali e quante abbreviazioni sono usate).

RFC 5952 raccomanda una **rappresentazione testuale canonica** (unica per ogni indirizzo) a supporto degli scenari in cui si ha bisogno di confrontare le rappresentazioni testuali degli indirizzi:

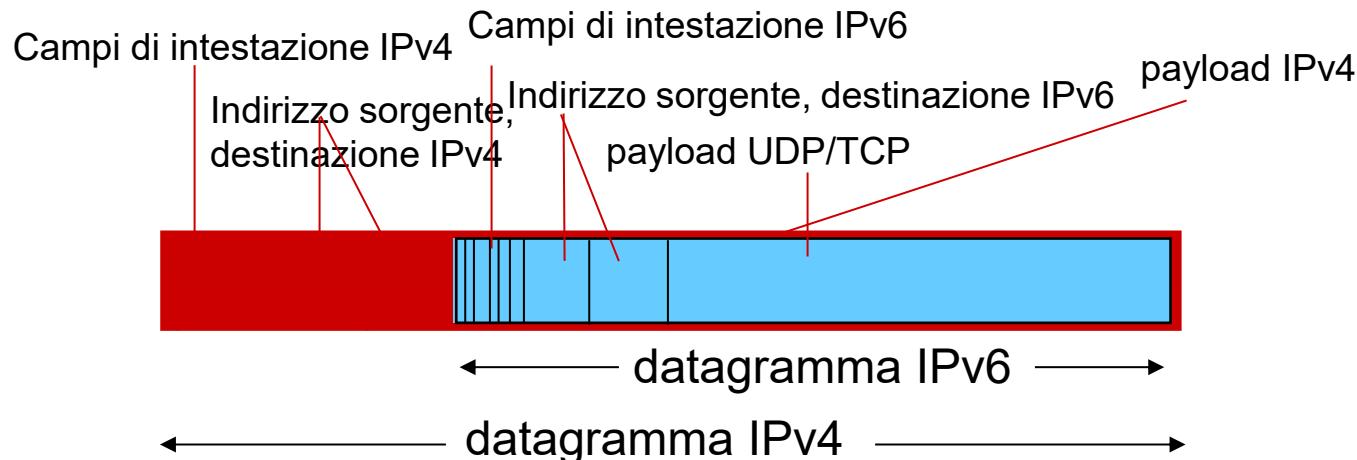
- solo lettere minuscole (per le cifre a, b, c, d, e, f)
- gli zeri iniziali sono abbreviati
- : : viene usato al massimo della sua capacità:

Per esempio, 2001 : db8 : : **0** : 8a2e : 370 : 7344 non va bene perché abbiamo lasciato uno zero (quello in rosso)

- : : è usato per abbreviare la sequenza più lunga (se ci sono due o più alternative) di almeno due o più campi 0

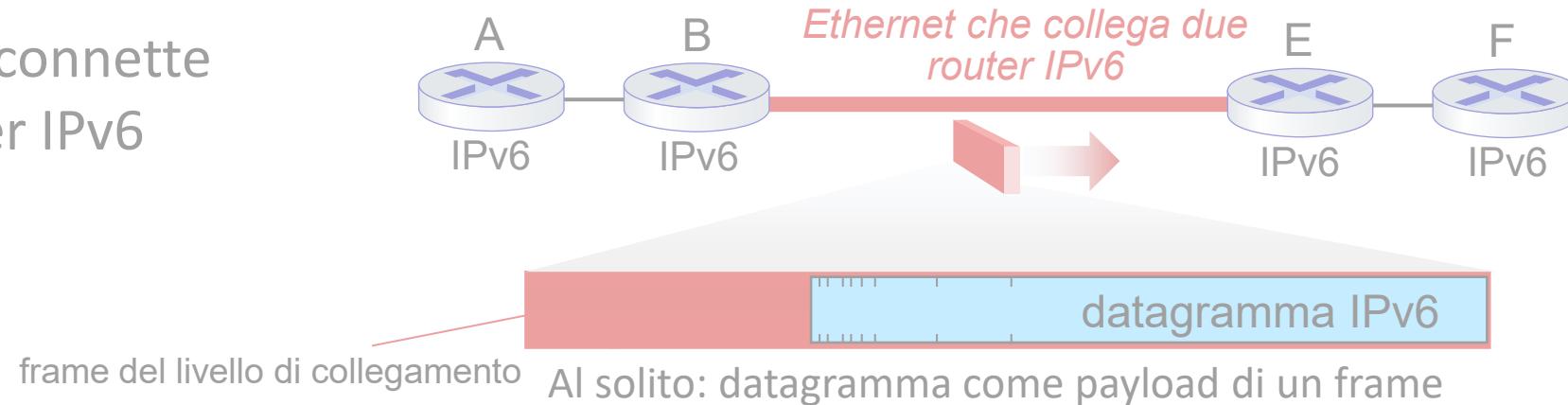
Transizione da IPv4 a IPv6

- non tutti i router possono essere aggiornati contemporaneamente
 - no "flag day" (ovvero, una "giornata campale" in cui tutte le macchine sono spente e aggiornate a IPv6)
 - come funzionerà la rete con un mix di router IPv4 e IPv6?
- **tunneling:** datagramma IPv6 trasportato come *payload* in un datagramma IPv4 tra i router IPv4 ("pacchetto nel pacchetto")
 - tunneling utilizzato ampiamente in altri contesti (4G/5G)

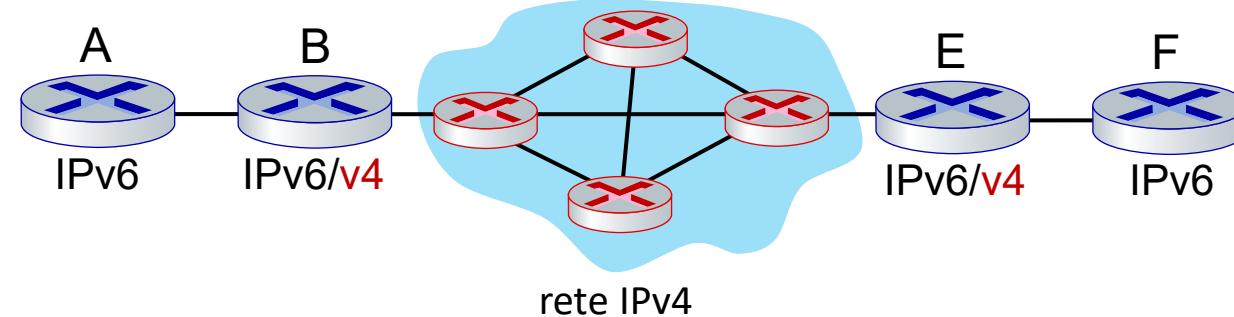


Tunneling e incapsulamento

Ethernet connette
due router IPv6

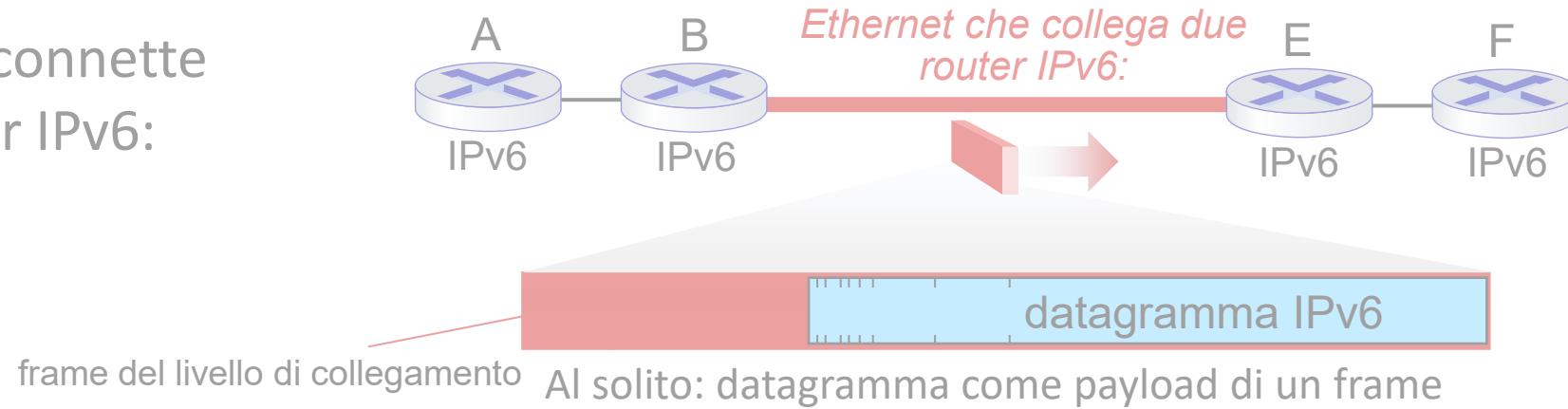


una rete IPv4
connette due
router IPv6

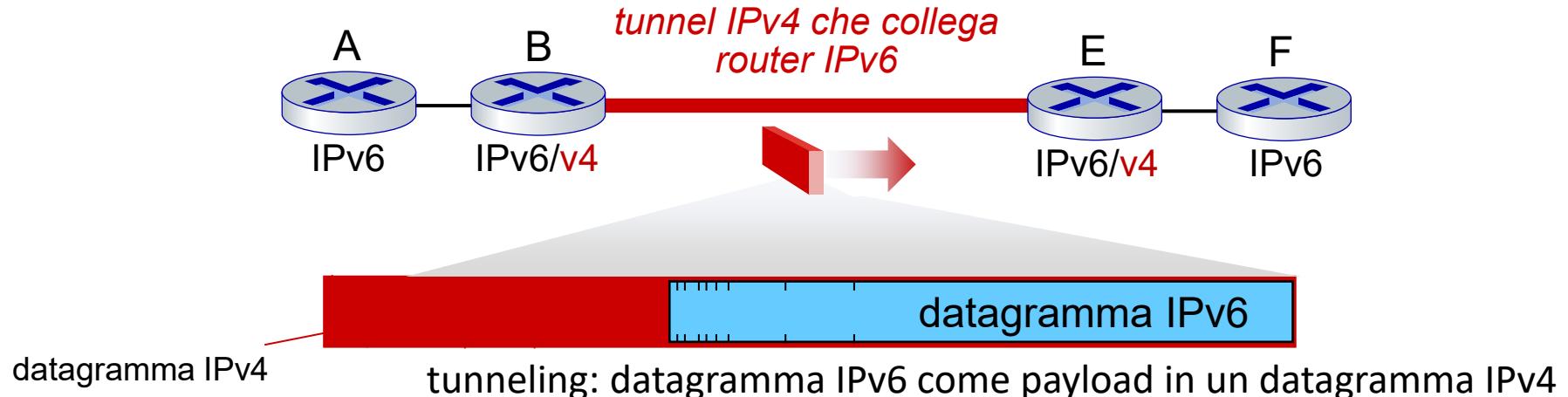


Tunneling e encapsulamento

Ethernet connette
due router IPv6:



tunnel IPv4
connette due
router IPv6



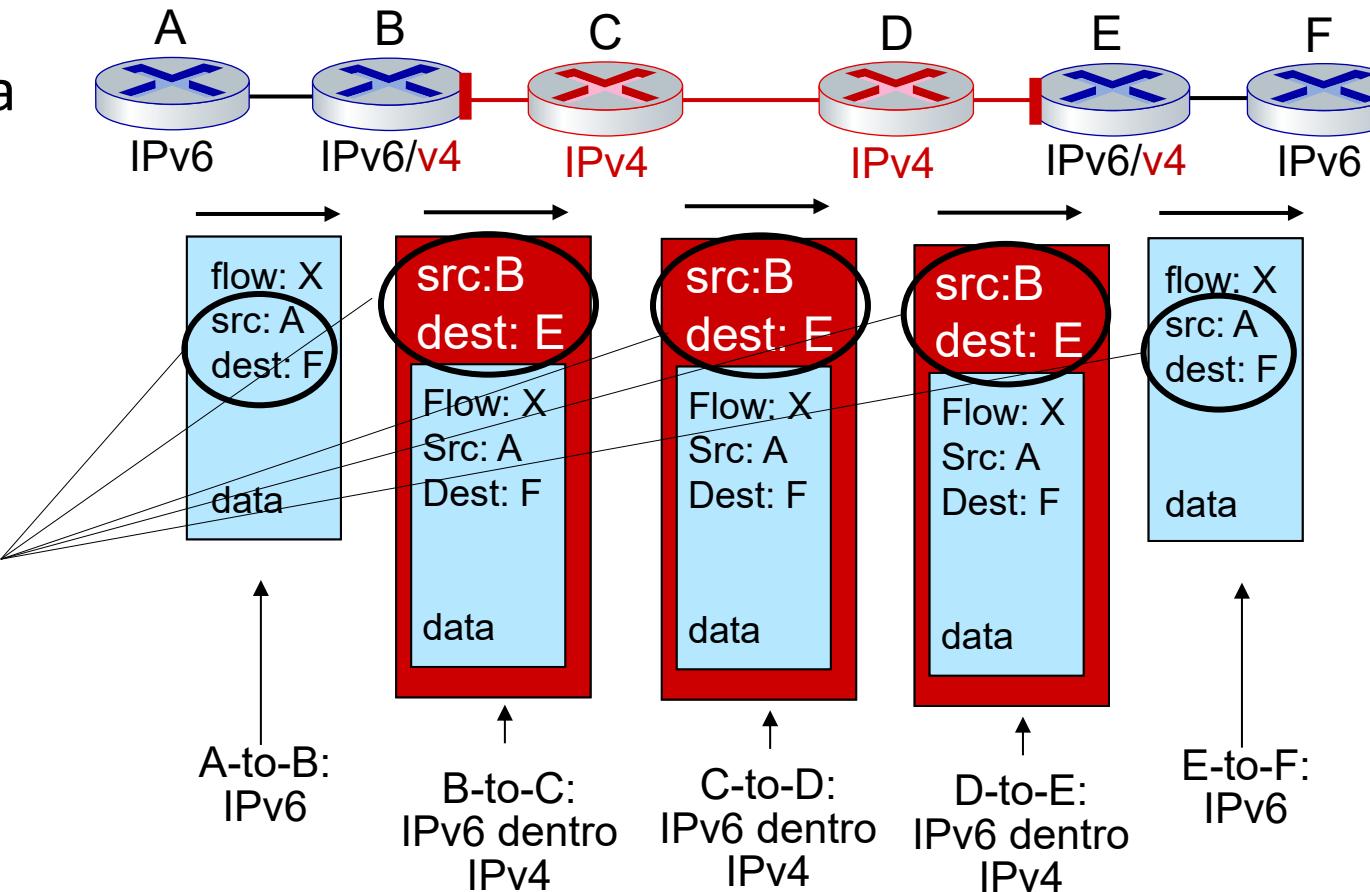
Tunneling

vista logica



visione fisica

osservate gli
indirizzi di
destinazione

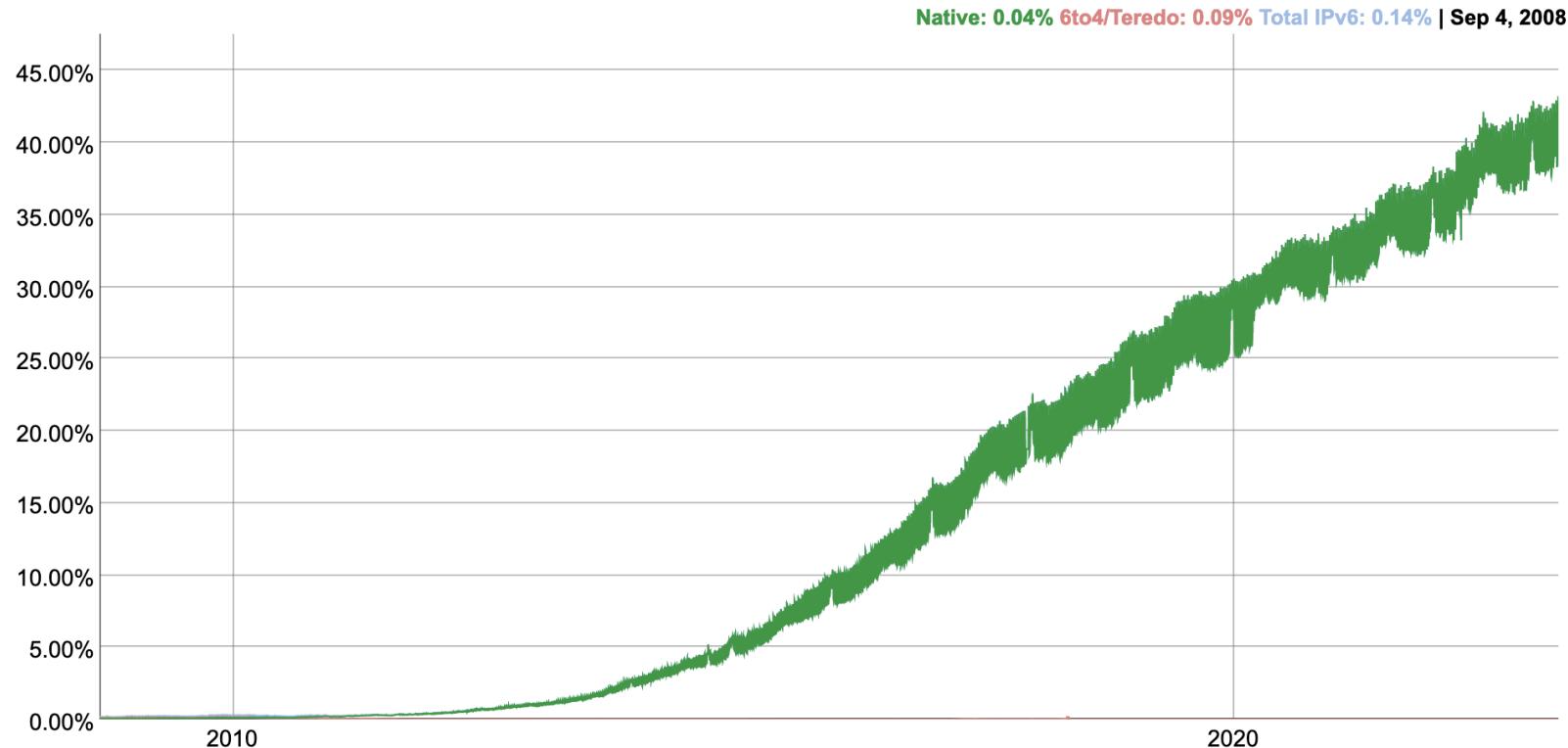


IPv6: adozione

- Google¹: ~ 40% dei client accede ai suoi servizi attraverso IPv6 (2023)
- NIST: 1/3 di tutti i domini governativi US sono abilitati a IPv6

IPv6 Adoption

We are continuously measuring the availability of IPv6 connectivity among Google users. The graph shows the percentage of users that access Google over IPv6.



IPv6: adozione

- Google¹: ~ 40% dei client accede ai suoi servizi attraverso IPv6 (2023)
- NIST: 1/3 di tutti i domini governativi US sono abilitati a IPv6
- Lungo (lunghissimo!) tempo per l'installazione e l'uso
 - 25 anni e oltre!
 - pensate ai cambiamenti a livello di applicazione negli ultimi 25 anni: WWW, social media, streaming multimediale, gaming, telepresenza, ...
 - *Perché?*

¹ <https://www.google.com/intl/en/ipv6/statistics.html>

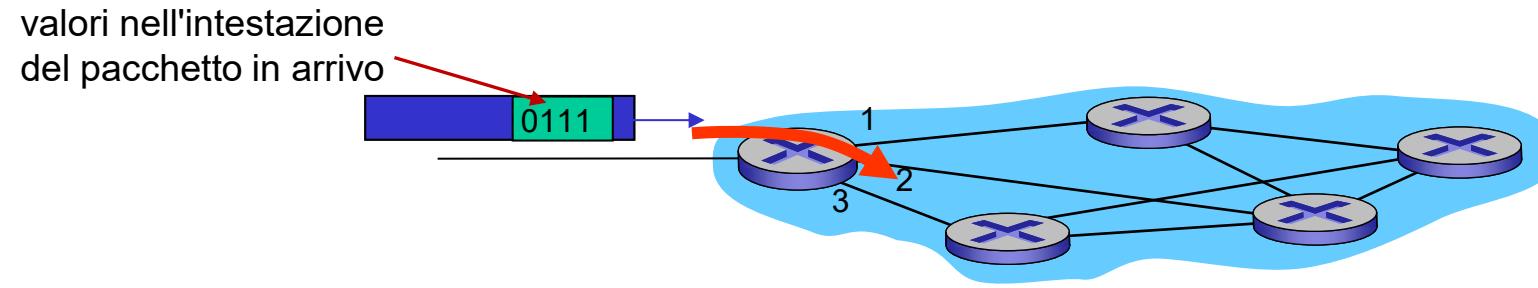
Livello di rete: tabella di marcia sul “piano dei dati”

- Livello di rete: panoramica
 - piano dei dati
 - piano di controllo
- Cosa c'è dentro un router
 - Porte di ingresso, struttura di commutazione, porte di uscita
 - buffer management, scheduling
- IP: il Protocollo Internet
 - Formato dei datagrammi
 - indirizzamento
 - Traduzione degli indirizzi di rete
 - IPv6



- Inoltro generalizzato, SDN
 - Match+action
 - OpenFlow: match+action in azione
- Middlebox

Inoltro generalizzato: match plus action



Inoltro generalizzato: match plus action

Ripasso: ciascun router ha una **tabella di inoltro**(o: **tabella dei flussi**)

- astrazione “**match plus action**”: cerca corrispondenze nei bit dei pacchetti in arrivo, agisce
 - *inoltro basato sulla destinazione*: inoltra in base all'indirizzo IP del destinatario
 - *inoltro generalizzato*:
 - più campi di intestazione posso determinare l'azione
 - più azioni possibili: scarta/copia/modifica/logga il pacchetto

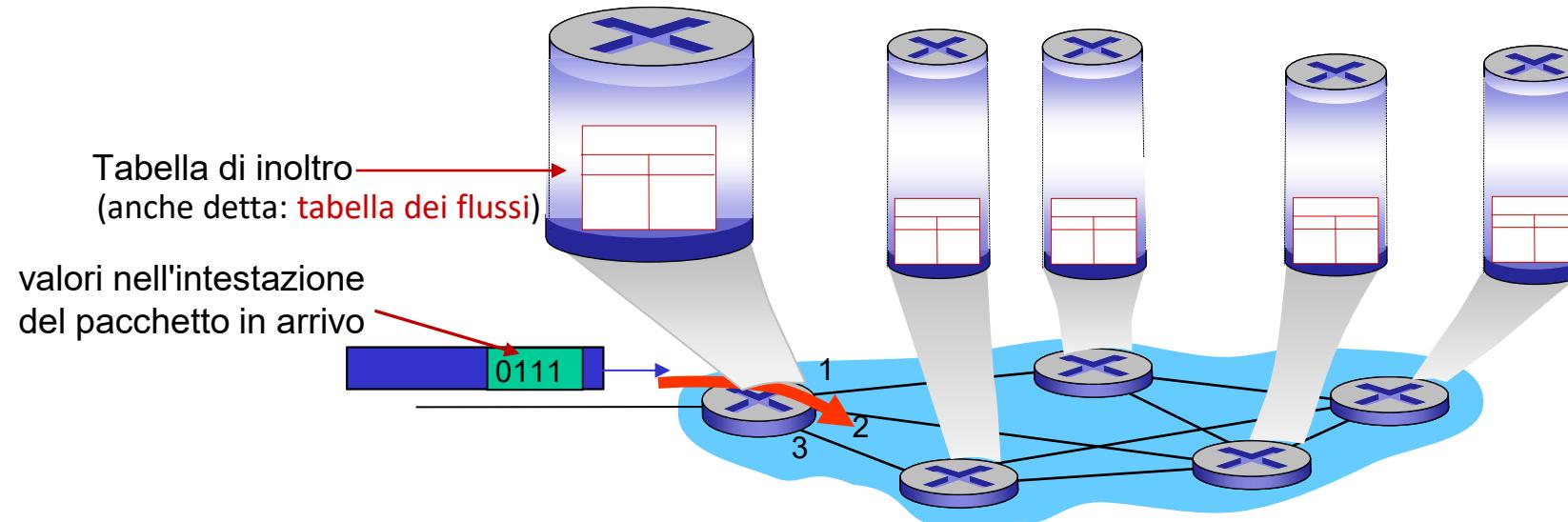


Tabella dei flussi

- **flusso:** definito dai valori campi di intestazione (a livello di collegamento, rete o trasporto)
- **inoltro generalizzato:** **semplici** regole per la gestione dei pacchetti
 - **match:** pattern sui valori dei campi di intestazione
 - **actions:** per il pacchetto in cui viene trovata una corrispondenza: scartare (drop), inoltrare (forward), modificare l'intestazione (modify), o inviare al controllore
 - **priorità:** disambigua pattern sovrapposti
 - **contatori:** numero di byte e numero di pacchetti , marca temporale ultimo aggiornamento

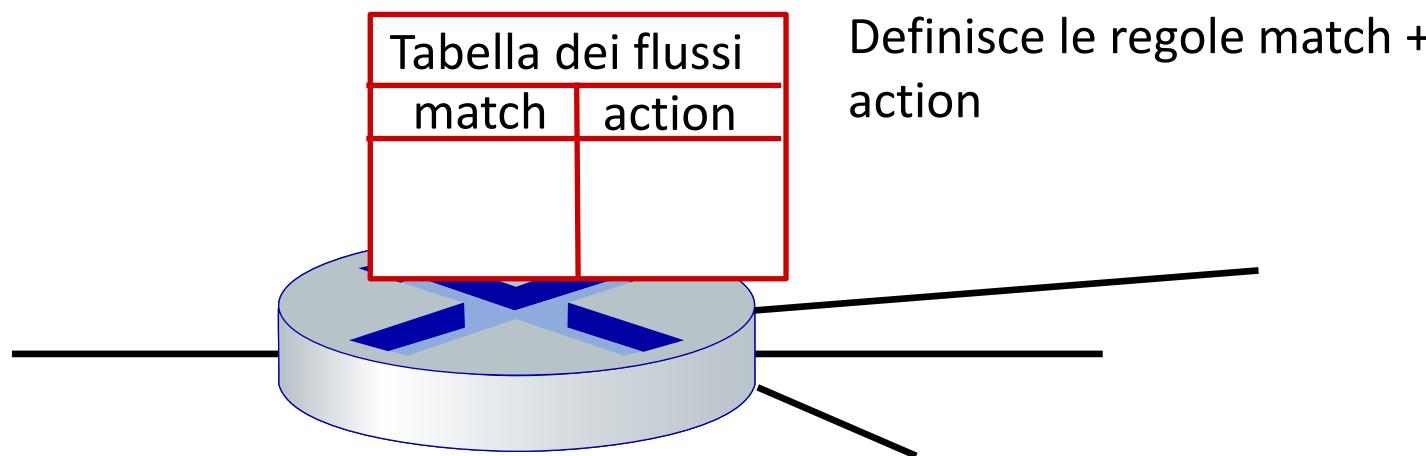
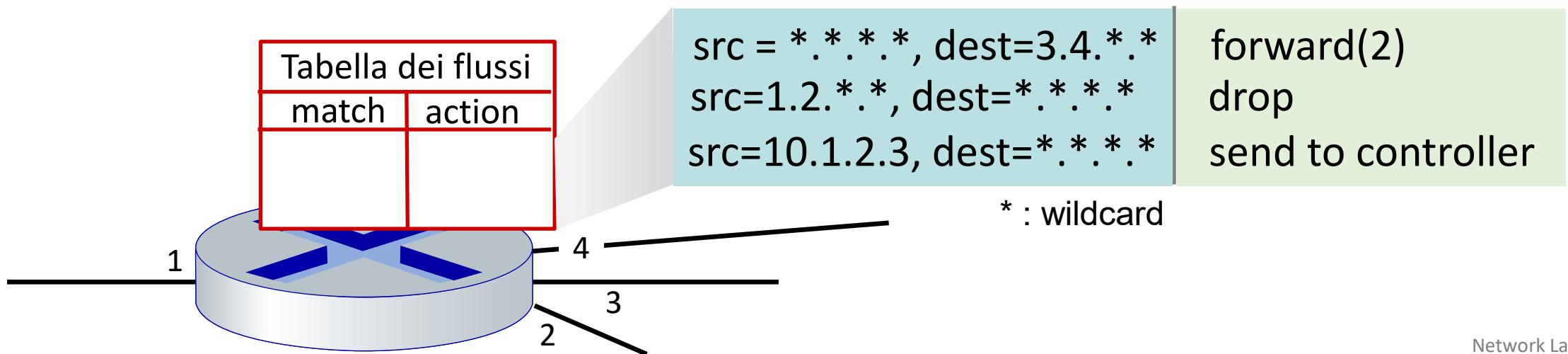
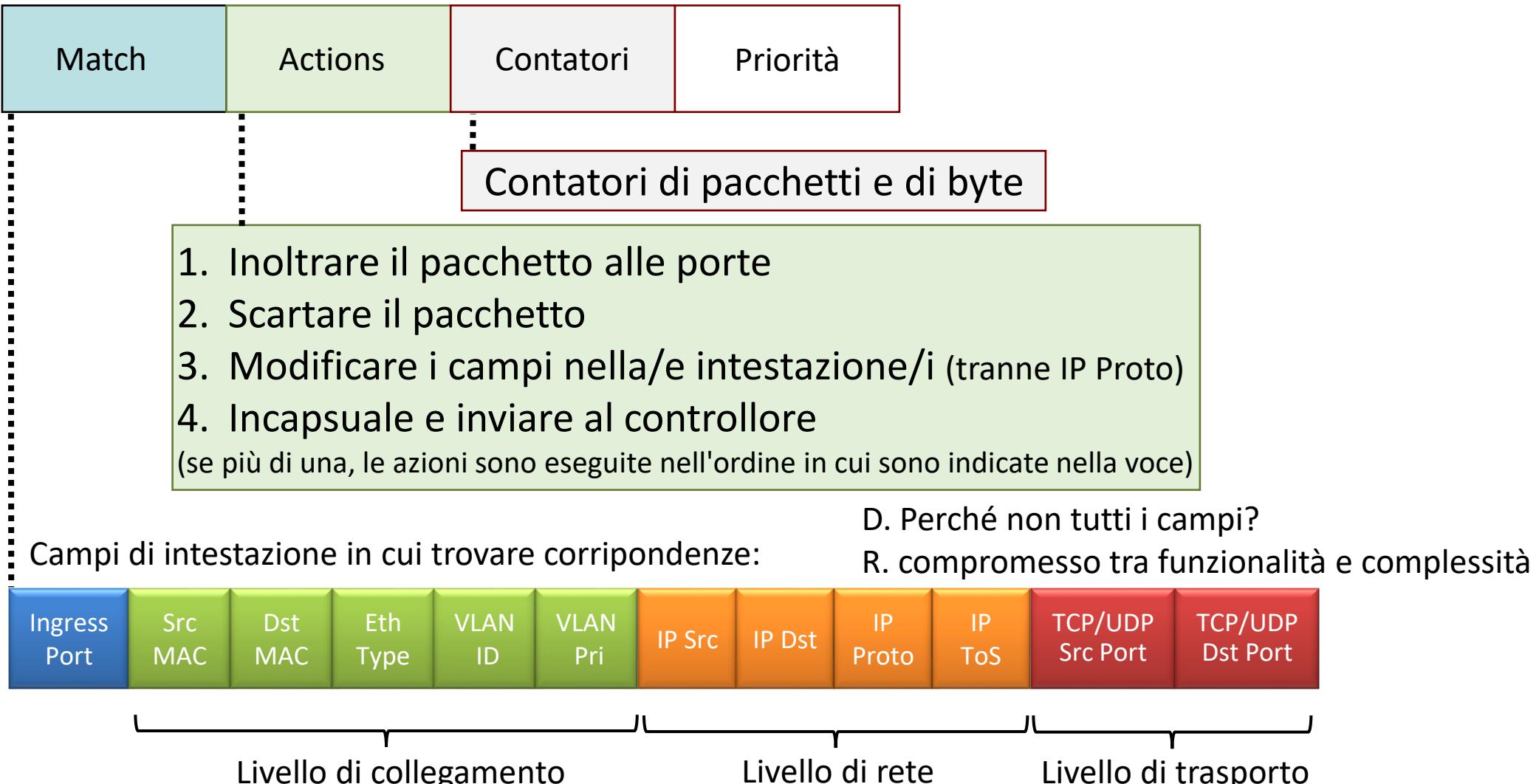


Tabella dei flussi

- **flusso:** definito dai valori campi di intestazione (a livello di collegamento, rete o trasporto)
- **inoltro generalizzato:** semplici regole per la gestione dei pacchetti
 - **match:** pattern sui valori dei campi di intestazione
 - **actions:** per il pacchetto in cui viene trovata una corrispondenza: scartare (drop), inoltrare (forward), modificare l'intestazione (modify), o inviare al controllore (che può, per esempio, aggiornare la tabella dei flussi prima di restituire il pacchetto per il suo inoltro)
 - **priorità:** disambigua pattern sovrapposti
 - **contatori:** numero di byte e numero di pacchetti, marca temporale ultimo aggiornamento



OpenFlow: voci della tabella di flusso



OpenFlow: esempi

Inoltro basato sulla destinazione:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	*	51.6.0.8	*	*	*	*	port6

I datagrammi IP destinati all'indirizzo IP 51.6.0.8 devono essere inoltrati alla porta di uscita 6 del router.

Firewall:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	*	*	*	*	*	*	22 drop

Bloccare (non inoltrare) tutti i datagrammi destinati alla porta TCP 22 (numero di porta ssh)

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	128.119.1.1	*	*	*	*	*	drop

Bloccare (non inoltrare) tutti i datagrammi inviati dall'host 128.119.1.1

OpenFlow: esempi

Inoltro basato sulla destinazione a Livello 2:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	22:A7:23: 11:E1:02	*	*	*	*	*	*	*	*	*	port3

frame di livello 2 con indirizzo MAC di destinazione 22:A7:23:11:E1:02 devono essere inoltrati alla porta di uscita 3

Load balancing

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
3	*	*	*	*	*	*	10.1.*.*	*	*	*	*	port2
4	*	*	*	*	*	*	10.1.*.*	*	*	*	*	port1

I pacchetti destinati a 10.1.*.* provenienti dalle porta 3 e 4 sono inviati rispettivamente sulle porta 2 e 1 (non possibile con l'inoltro basato sulla destinazione).

Astrazione in OpenFlow

- **match+action:** astrae dispositivi differenti

Router

- *match:* prefisso IP di destinazione più lungo
- *action:* inoltro (*forward*) attraverso un collegamento

Firewall

- *match:* indirizzi IP e numeri di porta TCP/UDP
- *action:* consentire (*permit*) o negare (*deny*)

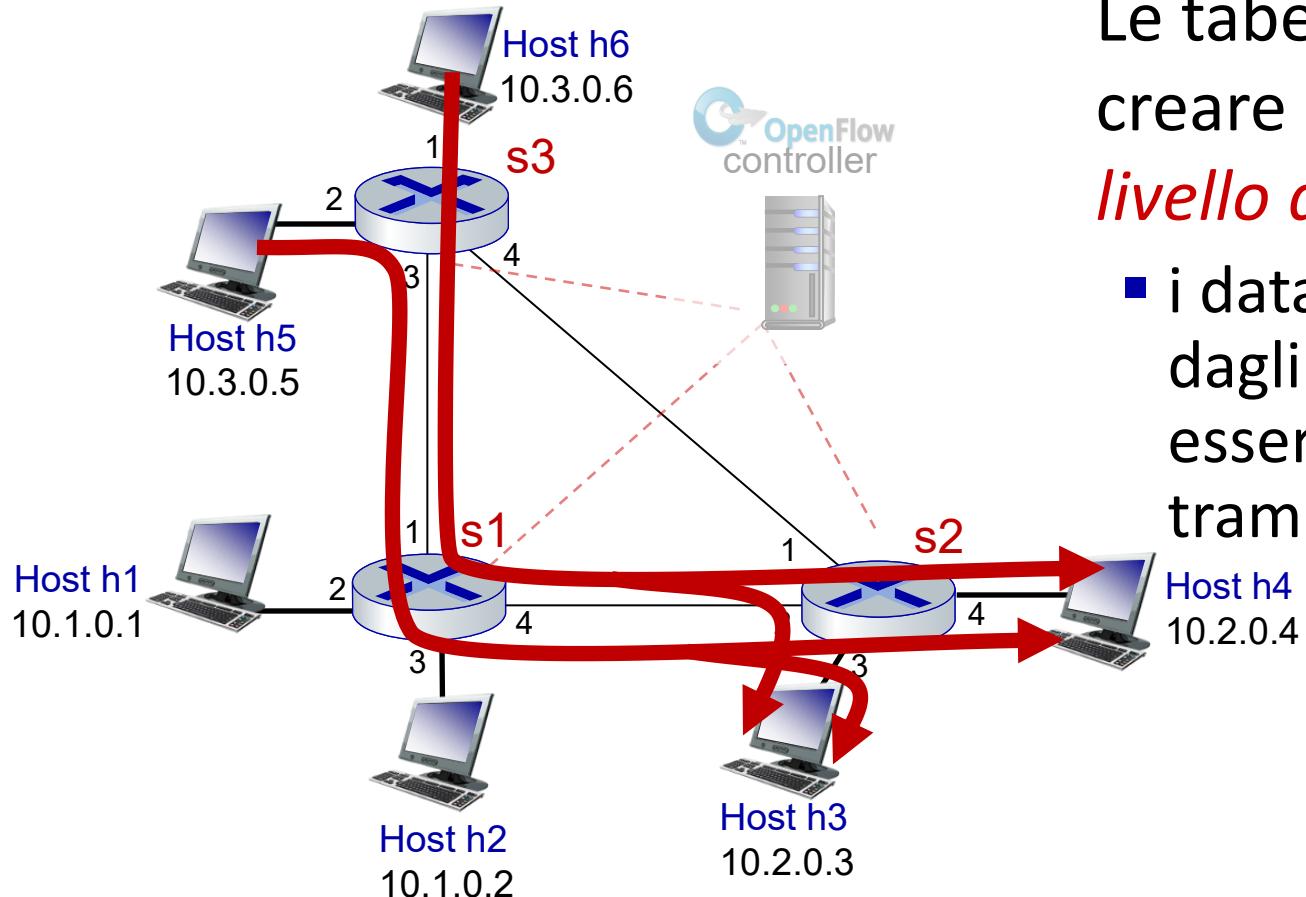
Switch

- *match:* indirizzo MAC di destinazione
- *action:* inoltra (*forward*) o inonda (*flood*)

NAT

- *match:* indirizzo IP e porta
- *action:* riscrive (*rewrite*) l'indirizzo e la porta

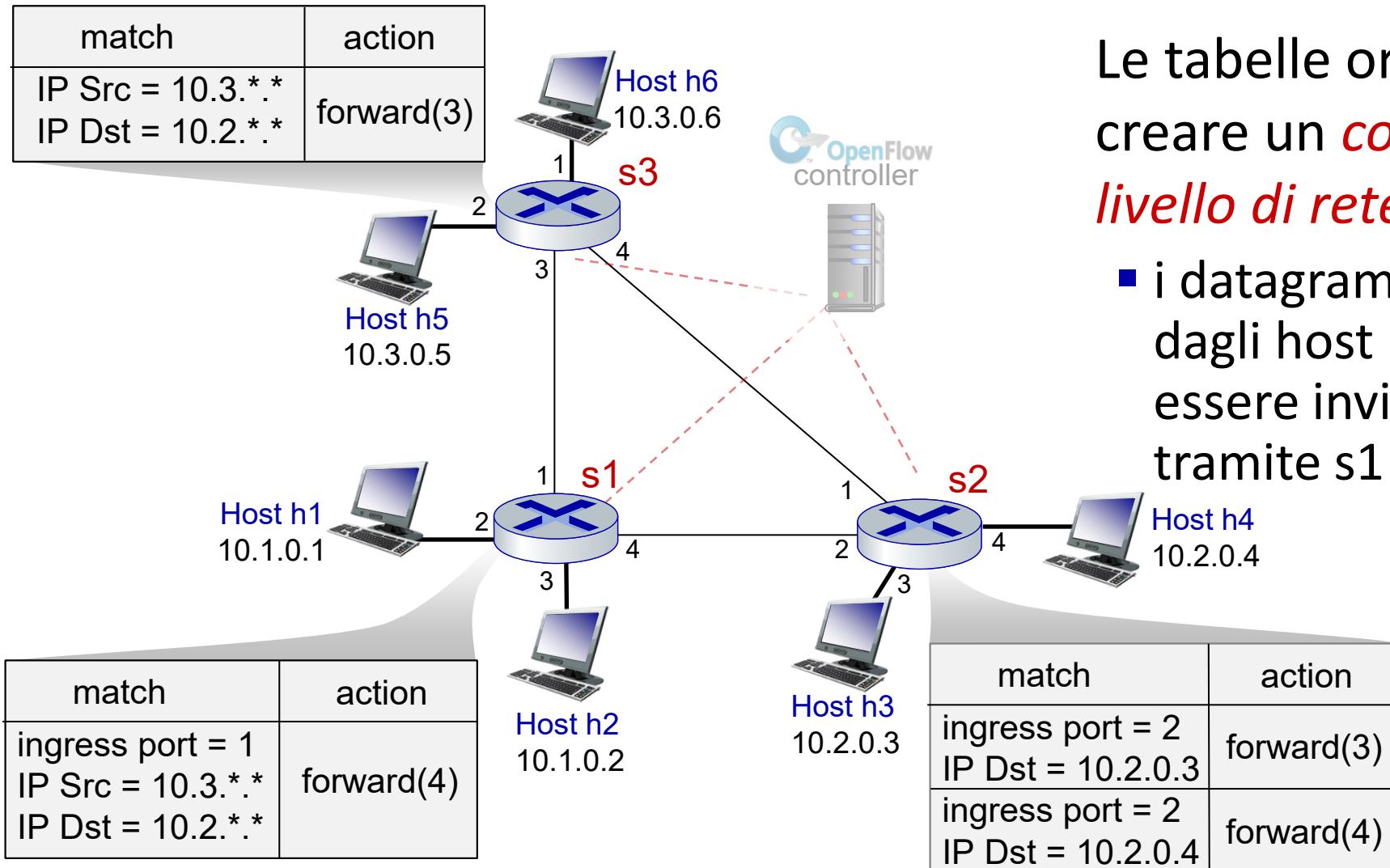
OpenFlow example



Le tabelle orchestrate possono creare un *comportamento a livello di rete*, es.:

- i datagrammi provenienti dagli host h5 e h6 devono essere inviati a h3 o h4, tramite s1 e da qui a s2

Esempio OpenFlow



Le tabelle orchestrate possono creare un *comportamento a livello di rete*, es.,:

- i datagrammi provenienti dagli host h5 e h6 devono essere inviati a h3 o h4, tramite s1 e da qui a s2

Inoltro generalizzato: riassunto

- astrazione “**match plus action**”: trova corrispondenze (*match*) nei bit nell'intestazione (di qualsiasi livello) dei pacchetti in arrivo, agisce (*action*)
 - trova corrispondenze su molti campi (livello di collegamento, rete, trasporto)
 - azioni locali: scarta (*drop*), inoltra (*forward*), modifica (*modify*), o invia il pacchetto al controllore
 - “programmare” *comportamenti di rete*
- una forma semplice di “programmabilità della rete”
 - “elaborazione” programmabile per pacchetto
 - *radici storiche: il networking attivo*
 - *oggi: programmazione più generalizzata:*
P4 (vedi p4.org).

Livello di rete: tabella di marcia sul "piano dei dati"

- Livello di rete: panoramica
- Cosa c'è dentro un router
- IP: il Protocollo Internet
- Inoltro generalizzato
- **Middlebox**
 - funzioni delle middlebox
 - evoluzione e principi architetturali di Internet



Middlebox

inoltro basato sulla destinazione

Middlebox (RFC 3234)

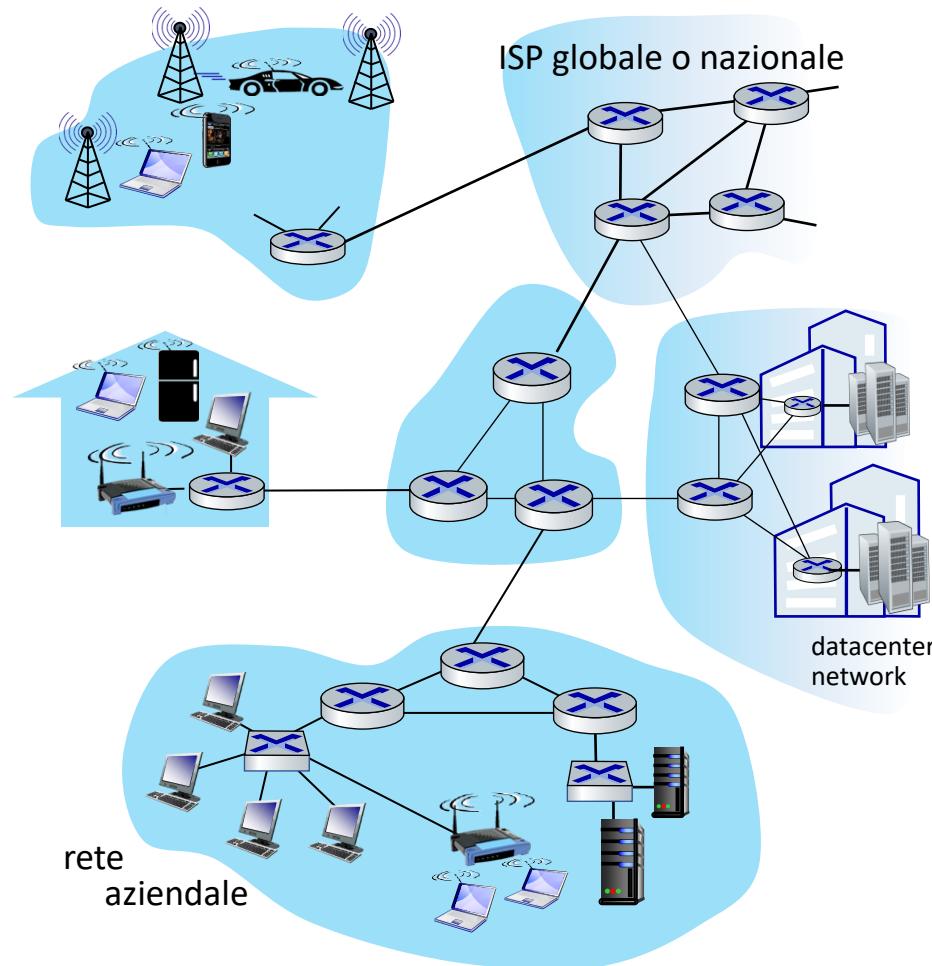
“qualsiasi box intermedio che svolge funzioni diverse da quelle normali e standard di un router IP sul percorso dei dati tra un host di origine e un host di destinazione”

si sta parlando di funzioni del piano dei dati all'interno della rete

Le middlebox sono ovunque!

NAT: nelle reti di accesso domestiche, aziendali e cellulare

Application-specific: fornitori di servizi, istituzionali, CDN



Firewalls, IDS (Intrusion Detection System): aziendale, istituzionale, fornitori di servizi, ISP

Load balancer: aziendale, fornitore di servizi, data center, reti mobili

Cache: fornitore di servizi, mobile, CDN

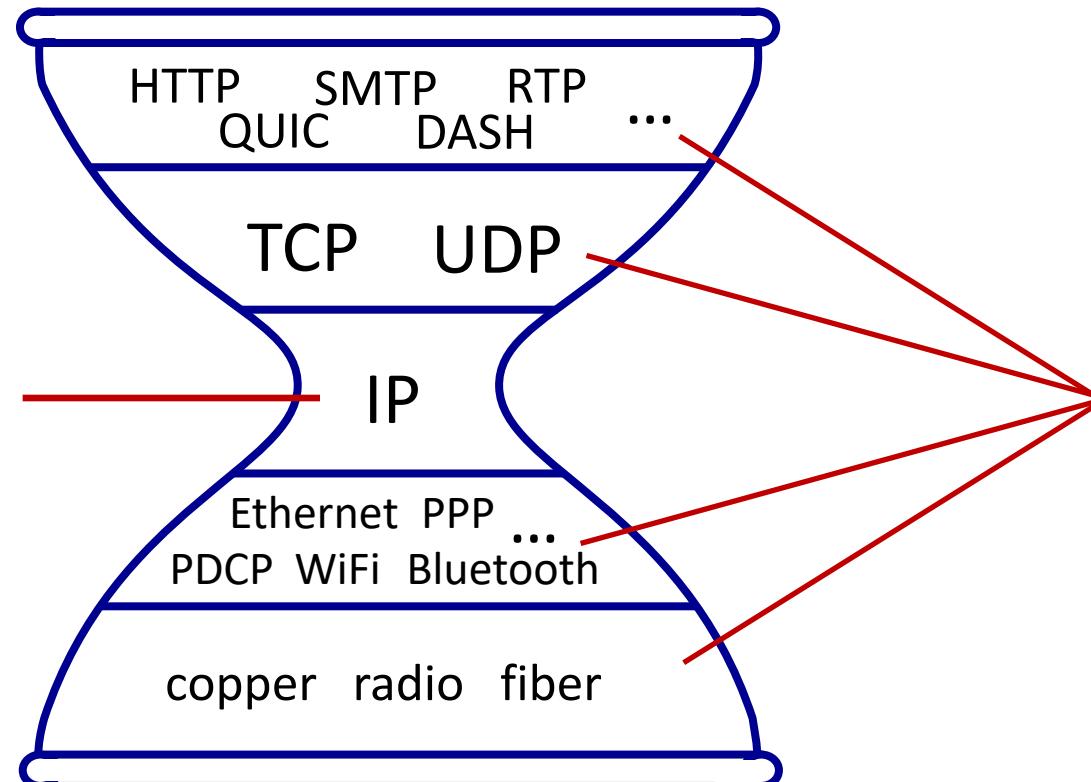
Middlebox

- inizialmente: soluzioni hardware proprietarie (chiuse)
- passaggio a hardware “whitebox” che implementa API aperte (es. OpenFlow)
 - abbandonare le soluzioni hardware proprietarie
 - azioni locali **programmabili** attraverso match+action
 - orientarsi verso l'innovazione/differenziazione nel software
- SDN: disaccoppia piano di controllo (centralizzato) da piano dei dati (distribuito)
- **Network Functions Virtualization (NFV)**: astrae le funzioni di rete dall'hardware: le funzioni di rete (es. router, switch, firewall) sono programmate in software e eseguite su hardware COTS (commodity off-the-shelf) (tramite VM o container), sfruttando risorse di calcolo, storage e rete. Sono usate svariate tecniche e tecnologie per migliorare le prestazioni. Possono essere quindi anche eseguite in cloud. NFV è complementare a SDN.

Le clessidra IP

La “vita stretta” di Internet:

- *un protocollo a livello di rete: IP*
- *deve essere implementato da ognuno dei (miliardi di) dispositivi connessi a Internet*

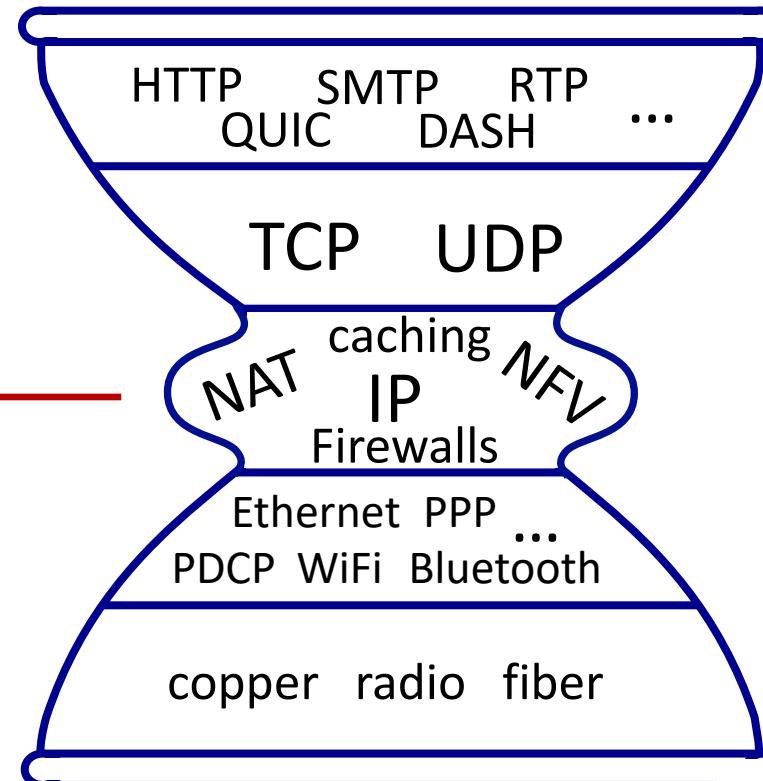


molti protocolli
nei livelli di
applicazione,
trasporto,
collegamento e
fisico

La clessidra IP, alla mezza età

Le “maniglie dell'amore”
della mezza età su
Internet?

- middlebox, che
operano all'interno
della rete



Principi architetturali di Internet

RFC 1958

"Molti membri della comunità di Internet sostengono che non esista un'architettura, ma solo una tradizione, mai messa per iscritto per i primi 25 anni (o almeno non dallo IAB). Tuttavia, in termini molto generali, la comunità crede che

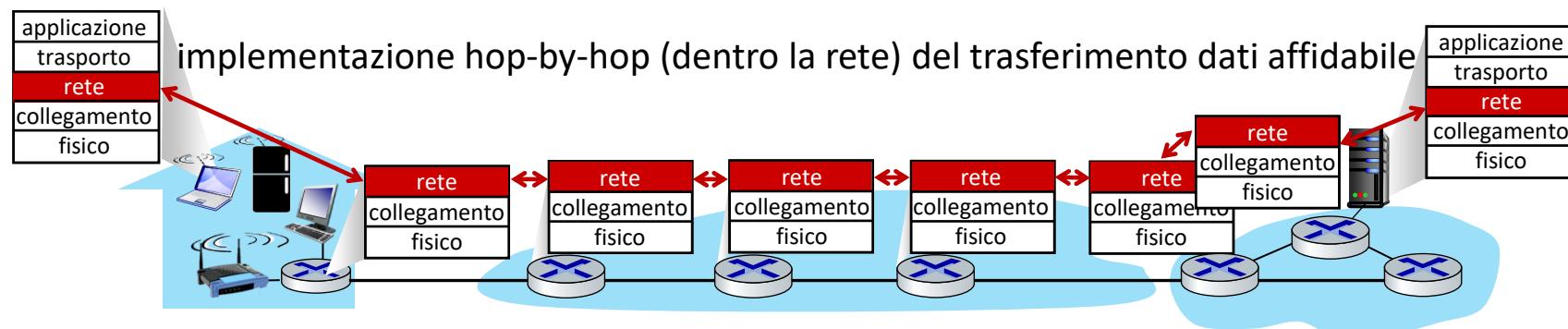
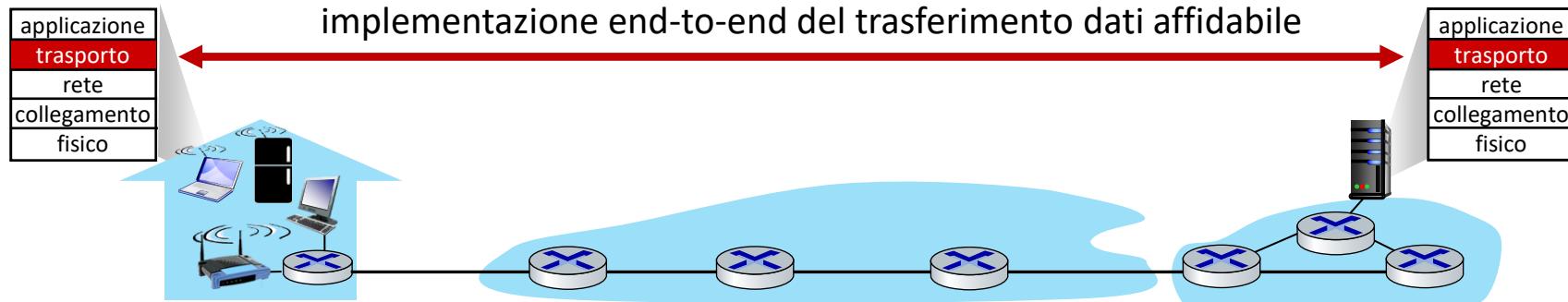
l'obiettivo sia la connettività, lo strumento sia il protocollo Internet e che l'intelligenza risieda più nel paradigma end-to-end che nascosta all'interno della rete"

Tre convinzioni fondamentali:

- connettività semplice (trasferimento di datagrammi tra host)
- protocollo IP: quella vita stretta (nasconde la eterogeneità sottostante)
- intelligenza, complessità alla periferia della rete

Il principio "end-to-end"

- alcune funzionalità (es., trasferimento dati affidabile, controllo della congestione) possono essere implementate nel **nucleo della rete** o nella **periferia della rete**



Il principio "end-to-end"

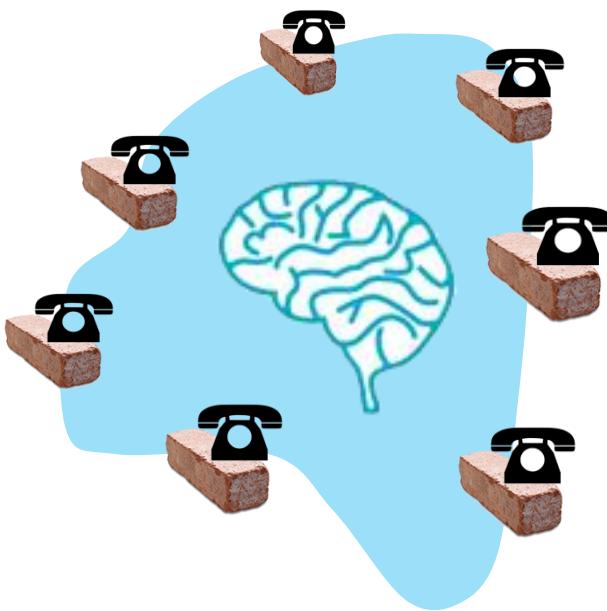
- alcune funzionalità (es., trasferimento dati affidabile, controllo della congestione) possono essere implementate nel nucleo della rete o nella periferia della rete

“La funzione in questione può essere implementata completamente e correttamente solo con la conoscenza e l’aiuto dell’applicazione che sta nell’endpoint del sistema di comunicazione. Pertanto non è possibile fornire tale funzione messa in discussione come una caratteristica del sistema di comunicazione stesso (a volte una versione incompleta della funzione fornita dal sistema di comunicazione può risultare utile come potenziamento delle prestazioni).

Questa linea di pensiero contraria all’implementazione delle funzioni nei livelli bassi viene chiamata “principio end-to-end”.

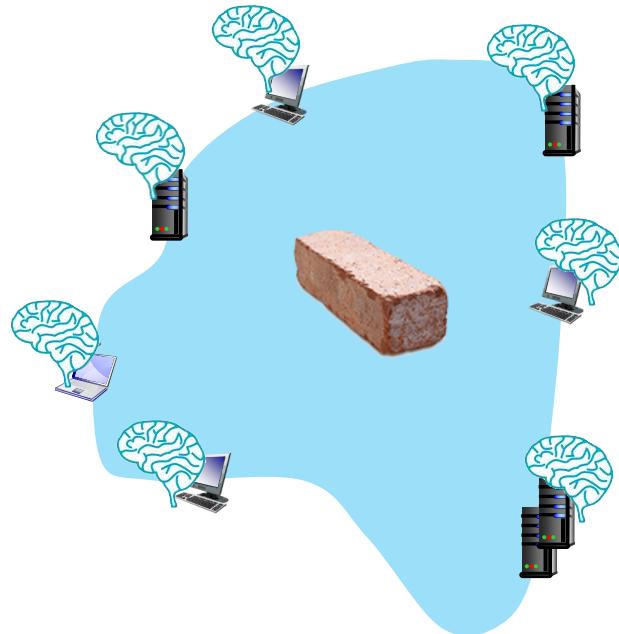
Saltzer, Reed, Clark 1981

Dove è l'intelligenza?



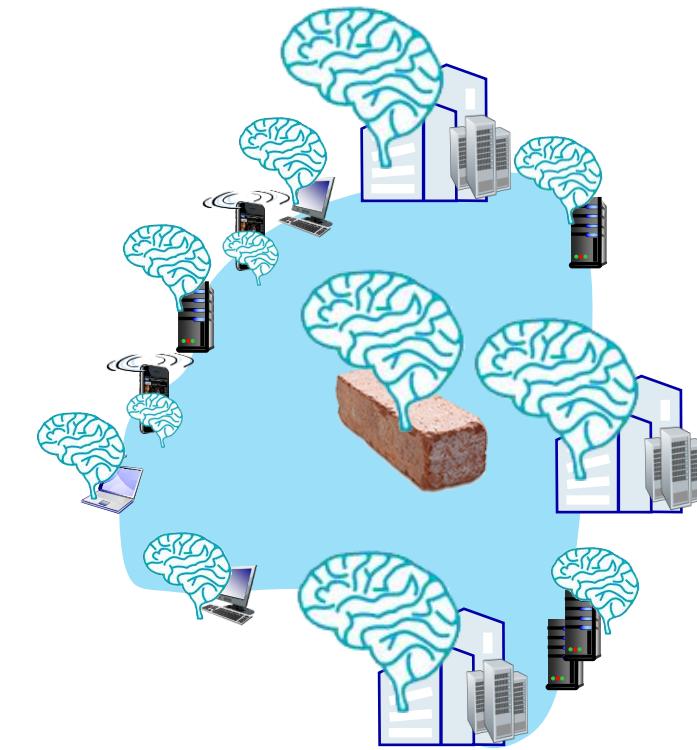
Rete telefonica del 20° secolo:

- intelligenza/calcolo negli switch di rete



Internet (pre-2005)

- intelligenza e calcolo nella periferia



Internet (post-2005)

- dispositivi di rete programmabili
- intelligenza, calcolo, infrastruttura massiccia a livello di applicazione alla periferia

Conclusione

- Livello di rete: panoramica
- Cosa c'è dentro un router
- IP: il Protocollo Internet
- Inoltro generalizzato, SDN
- Middlebox



Domanda: come sono calcolate le tabelle di inoltro (per l'inoltro basato sulla destinazione) o le tabelle dei flussi (per l'inoltro generalizzato)?

Risposta: dal piano di controllo