

# Livello di Rete

Gaia Maselli

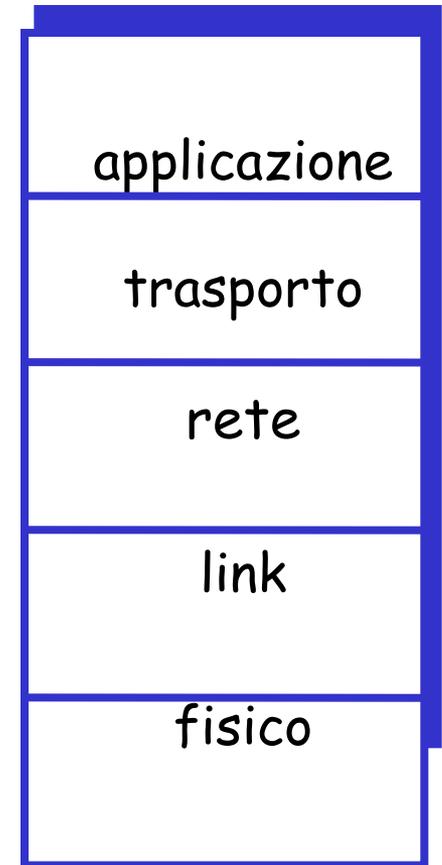
maselli@di.uniroma1.it

Parte di queste slide sono state prese dal materiale associato ai libri:

- 1) B.A. Forouzan, F. Mosharraf – Reti di calcolatori. Un approccio top-down. Copyright © 2013 McGraw-Hill Education Italy srl. Edizione italiana delle slide a cura di Gabriele D'Angelo e Gaia Maselli
- 2) Computer Networking: A Top Down Approach , 6th edition. All material copyright 1996-2009 J.F Kurose and K.W. Ross, All Rights Reserved

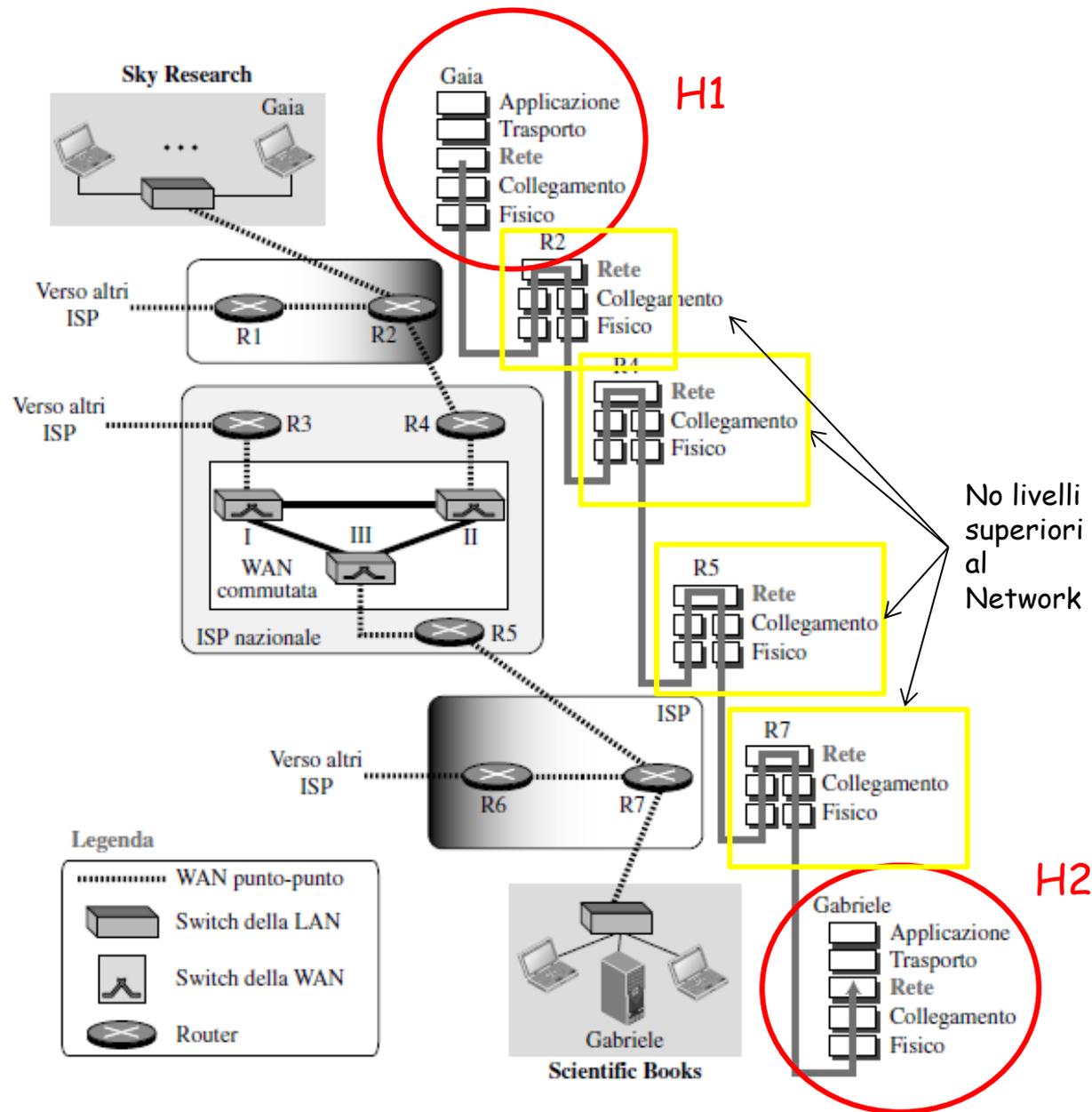
# Pila di protocolli Internet

- ❑ **applicazione:** di supporto alle applicazioni di rete
  - FTP, SMTP, HTTP
- ❑ **trasporto:** trasferimento dei messaggi a livello di applicazione tra il modulo client e server di un'applicazione
  - TCP, UDP
- ❑ **rete:** instradamento dei datagrammi dall'origine al destinatario
  - IP, protocolli di instradamento
- ❑ **link (collegamento):** instradamento dei datagrammi attraverso una serie di commutatori di pacchetto
  - PPP, Ethernet
- ❑ **fisico:** trasferimento dei singoli bit



# Esempio

- Livello di **trasporto**: comunicazione tra **processi**
- Livello di **rete**: comunicazione tra **host**
- Il livello di rete di H1 prende i segmenti dal livello di trasporto, li incapsula in un datagramma, e li trasmette al router più vicino
- Il livello di rete di H2 riceve i datagrammi da R7, estrae i segmenti e li consegna al livello di trasporto
- Il livello di rete dei nodi intermedi inoltra verso il prossimo router



Quando un router riceve un pacchetto come fa a sapere a chi deve inviarlo?

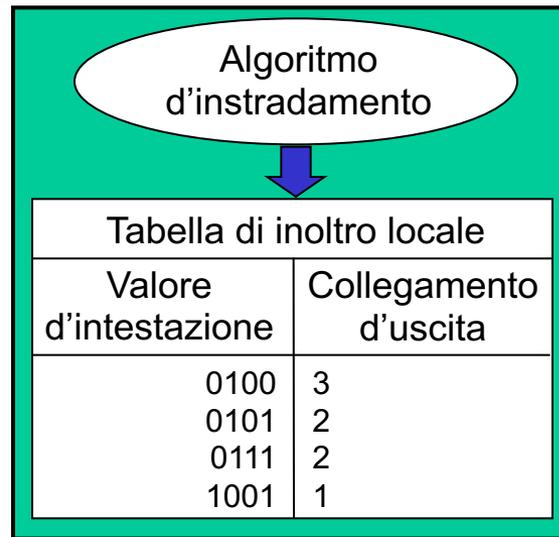
# Funzioni chiave del livello di rete

- *Instradamento (routing):*  
determina il percorso  
seguito dai pacchetti  
dall'origine alla destinazione
- *Inoltro (forwarding):*  
trasferisce i pacchetti  
dall'input di un router  
all'output del router  
appropriato
  - *Gli algoritmi di routing creano  
le tabelle di routing che  
vengono usate per il  
forwarding*

## analogia:

- *instradamento:* processo  
di pianificazione di un  
viaggio dall'origine alla  
destinazione
- *inoltro:* processo di  
attraversamento di un  
determinato svincolo

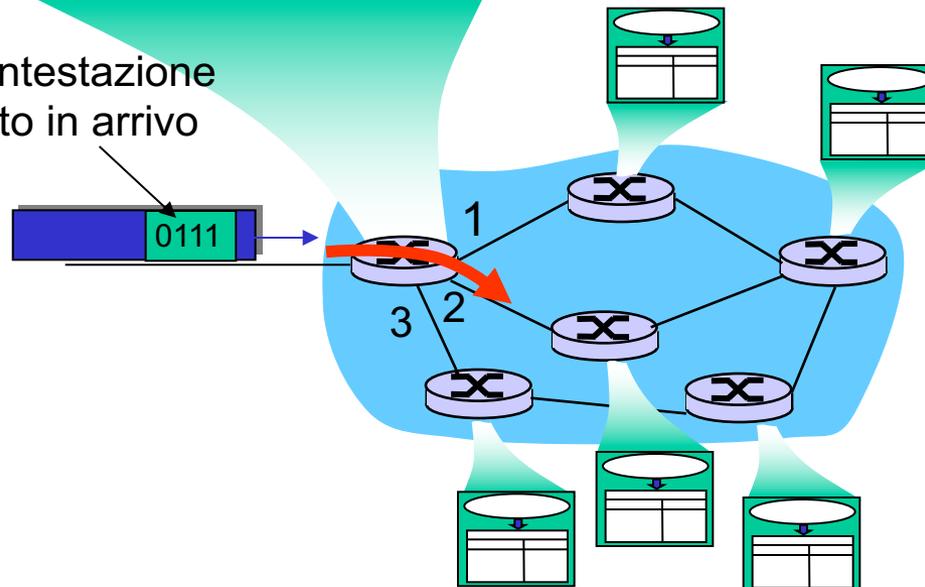
# Routing e forwarding



**Routing algorithm:** crea la forwarding table (determina i valori inseriti nella tabella)

**Forwarding table:** specifica quale collegamento di uscita bisogna prendere per raggiungere la destinazione

Valore nell'intestazione del pacchetto in arrivo



Ogni router ha la propria forwarding table

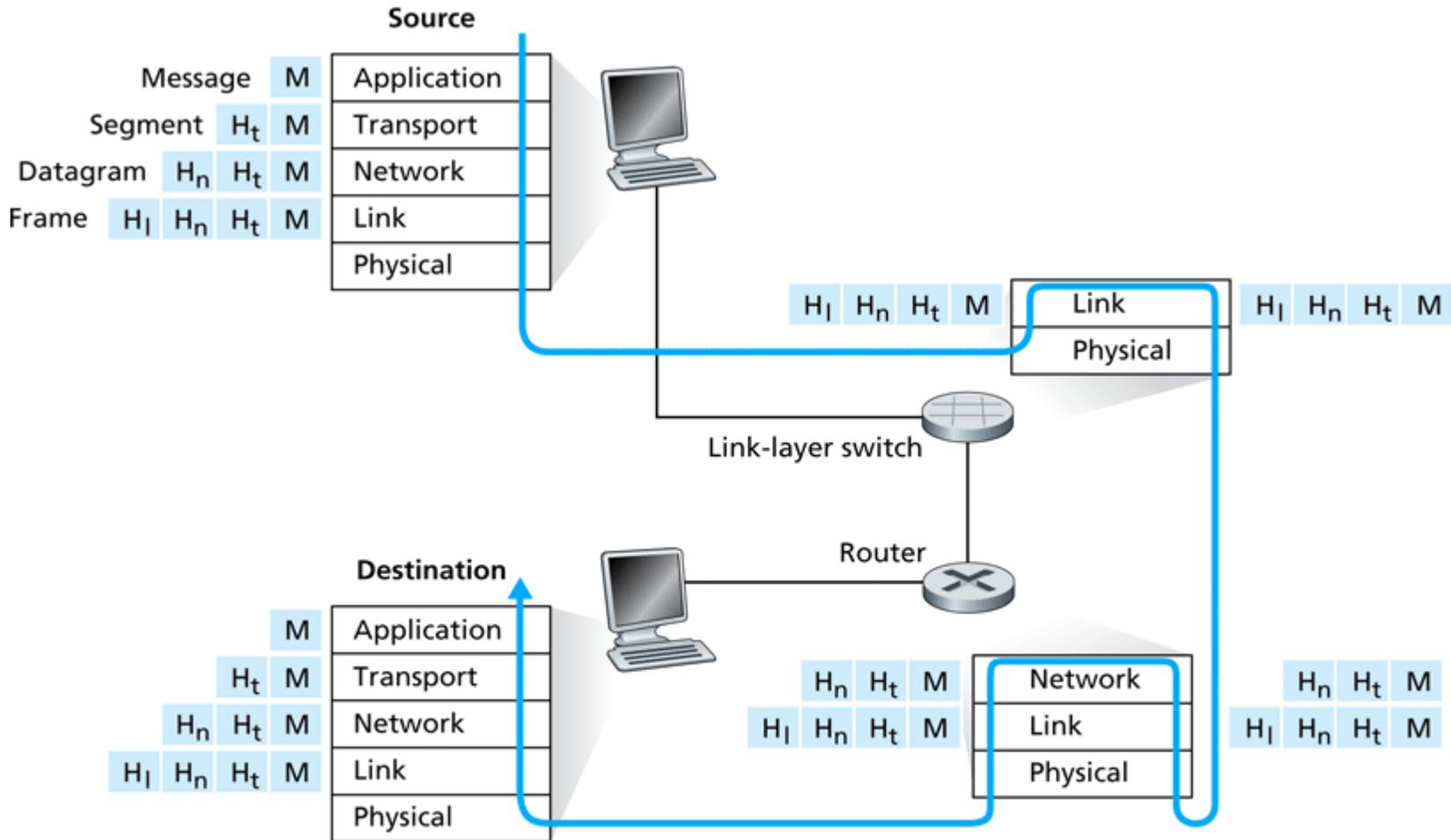
# Switch e router

*Packet switch (commutatore di pacchetto):*

dispositivo che si occupa del trasferimento dall'interfaccia di ingresso a quella di uscita, in base al valore del campo dell'intestazione del pacchetto

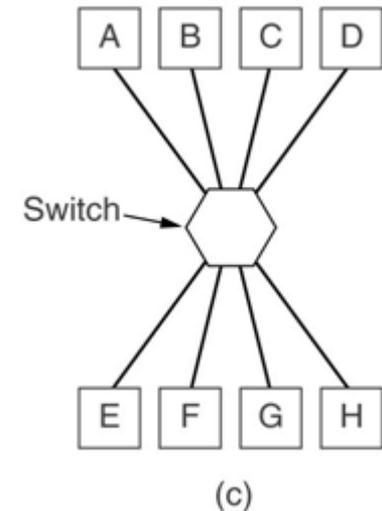
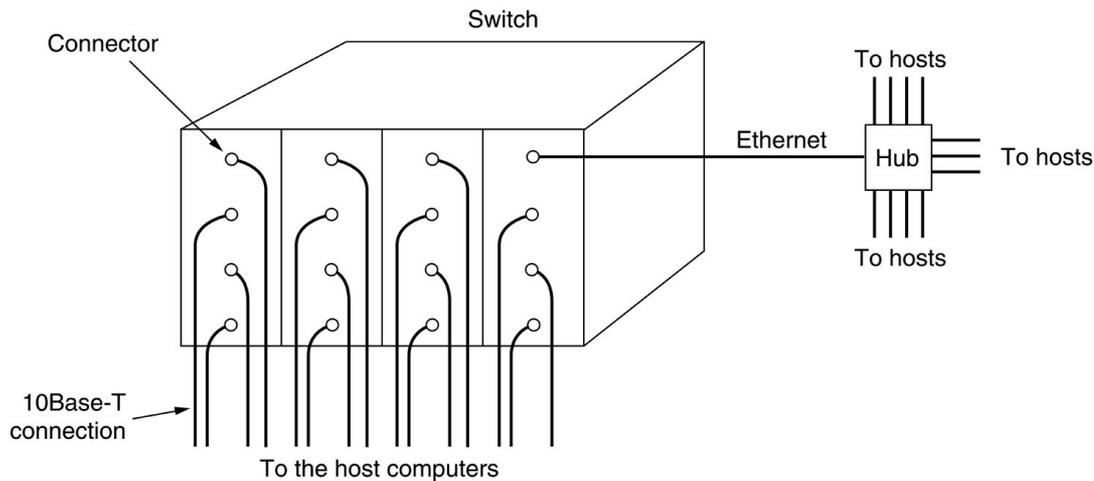
1. *Link-layer switch (commutatore a livello di collegamento):* stabiliscono l'inoltro in relazione al valore del campo nel livello di collegamento (livello 2)
2. *Router:* stabiliscono l'inoltro in base al valore del campo nel livello di rete (livello 3)

# Host, router, link-layer switch



# Link-layer switch

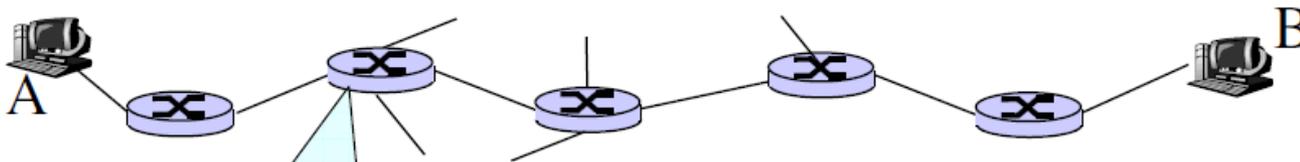
- ❑ Instrada pacchetti al livello 2 (collegamento)
- ❑ Utilizzato per collegare singoli computer all'interno di una rete LAN



# Router

□ Instrada pacchetti al livello 3 (rete)

□ Forward a chunk of information (called *packet*) arriving on one of its communication links to one of its outgoing communications link (the *next hop* on the source-to-destination path)



forwarding

- Receives the packet
- Based on a routing table and the destination address, computes the 'next hop' to the destination
- Forwards the packet to the next hop
- The process of computing and maintaining the routing table is called Routing

Routing table

Dest. Address	Next Hop

Circuit switching  
vs.  
packet switching

# Switching

- ❑ Approccio a circuito virtuale (servizio orientato alla connessione)
  - ❑ Prima che i datagrammi fluiscano, i due sistemi terminali e i router intermedi stabiliscono una connessione virtuale
- ❑ Approccio a datagramma (servizio senza connessione)
  - Ogni datagramma viaggia indipendente dagli altri

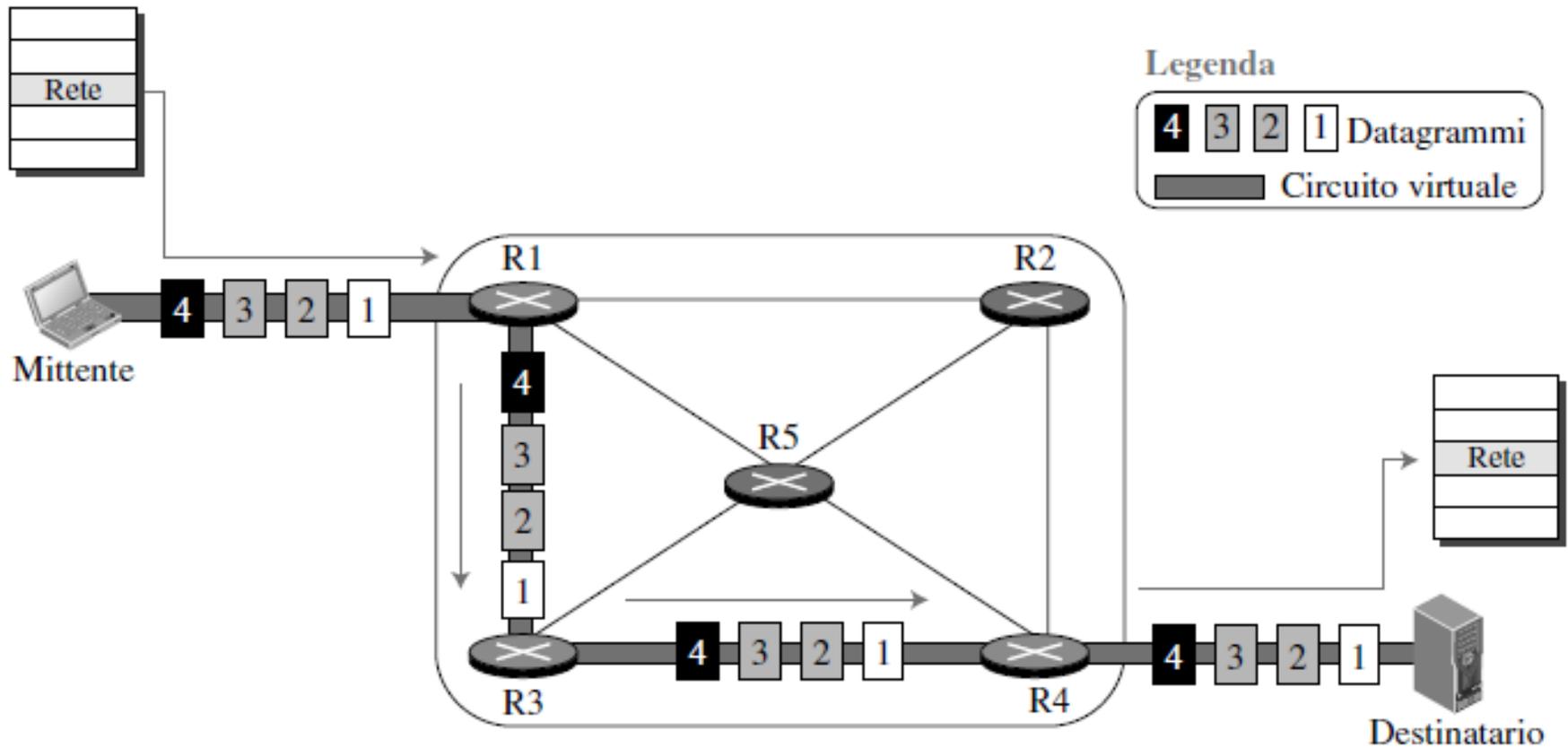
# Reti a circuito virtuale

“il percorso tra origine e destinazione si comporta in modo analogo a un circuito telefonico”

- ❑ prestazioni
- ❑ coinvolgimento della rete durante il percorso tra sorgente e destinazione

- ❑ Il pacchetto di un circuito virtuale ha un numero VC (etichetta di circuito) nella propria intestazione.
- ❑ Un circuito virtuale può avere un numero VC diverso su ogni collegamento.
- ❑ Ogni router sostituisce il numero VC con un nuovo numero.

# Rete a circuito virtuale

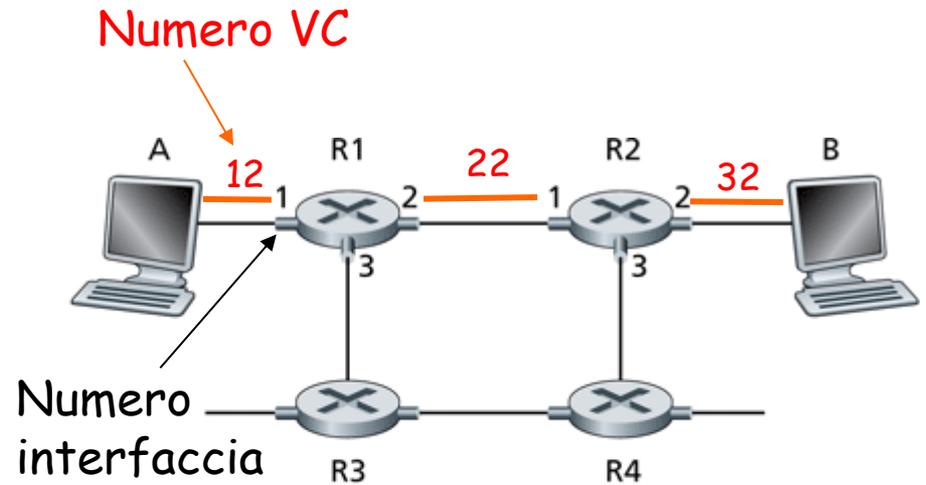


# Implementazioni

Un circuito virtuale consiste in:

1. un percorso tra gli host origine e destinazione
  2. numeri VC, uno per ciascun collegamento
  3. righe nella tabella d'inoltro in ciascun router.
- Il pacchetto di un circuito virtuale ha un numero VC nella propria intestazione che rappresenta un'etichetta di flusso.
  - Il numero VC del pacchetto cambia su tutti i collegamenti lungo un percorso.
    - Un nuovo numero VC viene rilevato dalla tabella d'inoltro.

# Tabella d'inoltro



## Tabella d'inoltro R1:

Interf.in ingresso	Nr. VC entrante	Interf. in uscita	Nr. VC uscente
1	12	2	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

**I router mantengono le informazioni sullo stato delle connessioni!**

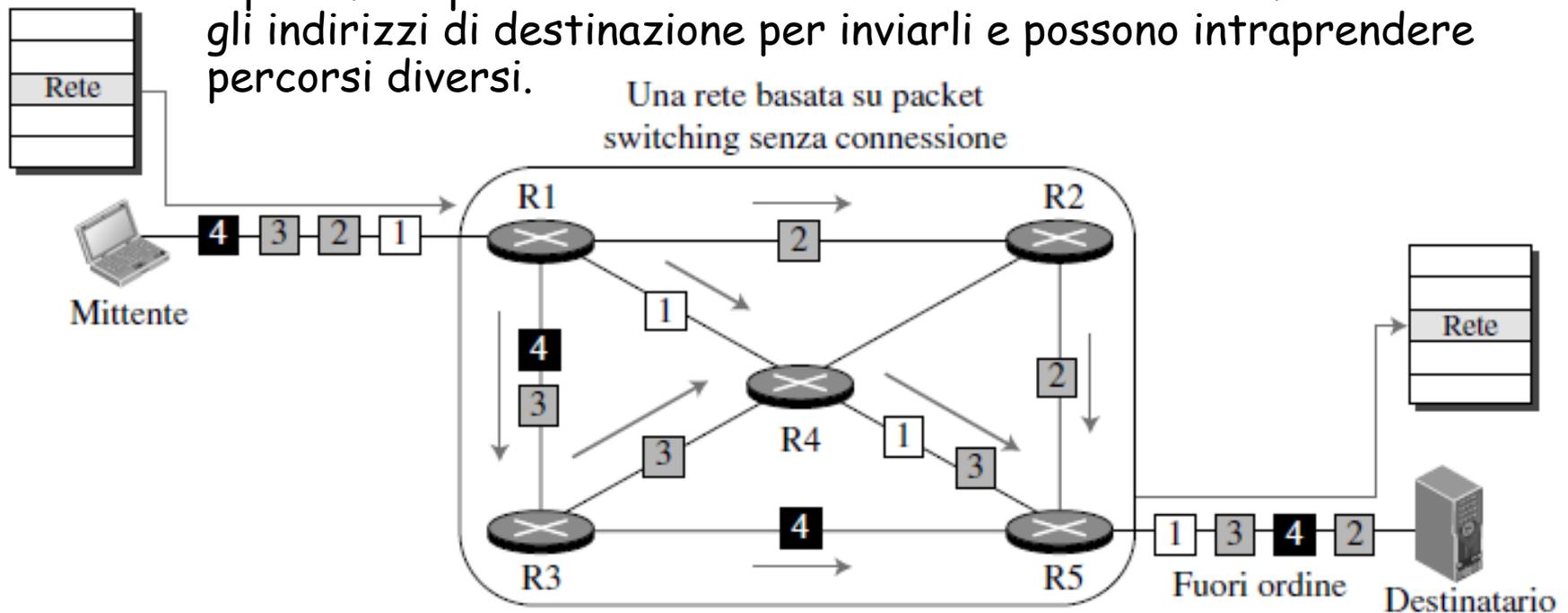
- Aggiungono alla tabella d'inoltro una nuova riga ogni volta che stabiliscono una nuova connessione (la cancellano quando la connessione viene rilasciata)

# ATM (Asynchronous transfer mode)

- ❑ Rete orientata alla connessione
- ❑ Progettata nei primi anni 90
- ❑ Scopo di unificare voce, dati, televisione via cavo, etc.
- ❑ Attualmente usata nella rete telefonica per trasportare (internamente) pacchetti IP
- ❑ Le connessioni vengono chiamate *circuiti virtuali* (in analogia con quelli telefonici che sono circuiti fisici)
- ❑ Quando una connessione è stabilita, ciascuna parte può inviare dati (suddivisi in celle di 53 bytes)

# Reti a datagramma

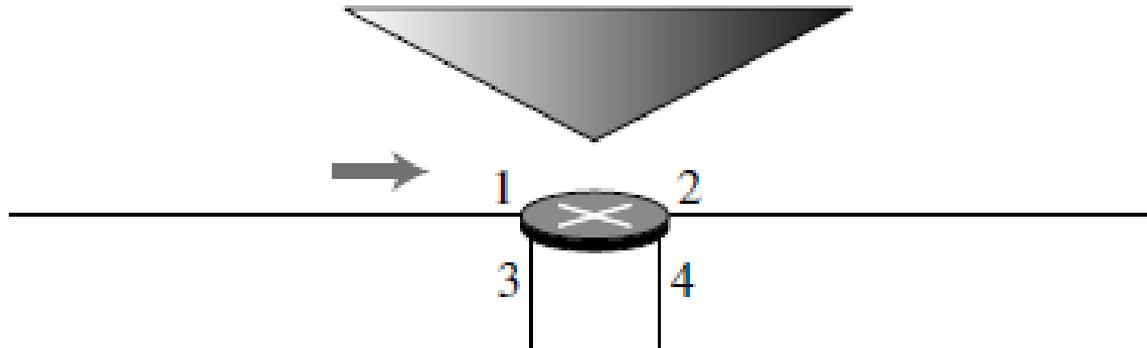
- ❑ **Internet** è una rete a datagramma (packet switched)
- ❑ L'impostazione della chiamata non avviene a livello di rete
- ❑ I router della rete a datagramma non conservano informazioni sullo stato dei circuiti virtuali
  - ❑ Non c'è il concetto di "connessione" a livello di rete
- ❑ I pacchetti vengono inoltrati utilizzando l'indirizzo dell'host destinatario.
- I pacchetti passano attraverso una serie di router che utilizzano gli indirizzi di destinazione per inviarli e possono intraprendere percorsi diversi.



# Processo di inoltrimento in una rete a datagramma

Tabella d'inoltro

Indirizzo di destinazione	Interfaccia di output
A	1
B	2
⋮	⋮
H	3



# Tabella d'inoltro

4 miliardi  
di possibili indirizzi

<u>Intervallo degli indirizzi di destinazione</u>	<u>Interfaccia</u>
<b>da</b> 11001000 00010111 00010000 00000000 <b>a</b> 11001000 00010111 00010111 11111111	0
<b>da</b> 11001000 00010111 00011000 00000000 <b>a</b> 11001000 00010111 00011000 11111111	1
<b>da</b> 11001000 00010111 00011001 00000000 <b>a</b> 11001000 00010111 00011111 11111111	2
<b>altrimenti</b>	3

# Confronta un prefisso dell'indirizzo

<u>Corrispondenza di prefisso</u>	<u>Interfaccia</u>
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
altrimenti	3

## Esempi:

con: 11001000 00010111 00010110 10100001      Qual è l'interfaccia?

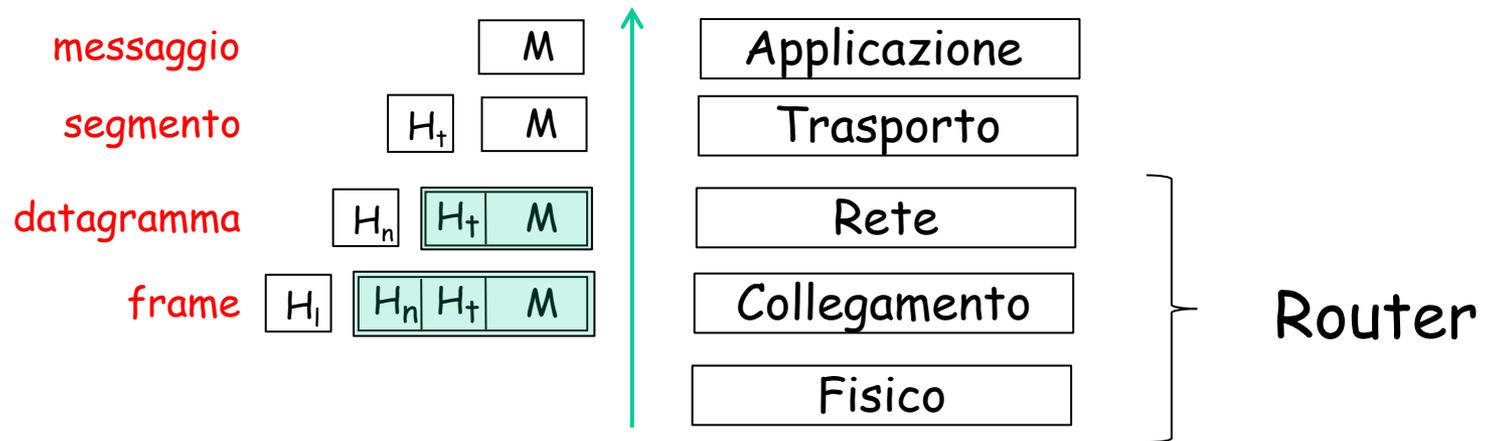
con: 11001000 00010111 00011000 10101010      Qual è l'interfaccia?

- Quando si verificano corrispondenze multiple si prende la corrispondenza a **prefisso più lungo**: viene determinata la corrispondenza più lunga all'interno della tabella e si inoltrano i pacchetti sull'interfaccia corrispondente.
- **Continuità** degli indirizzi

# Livello di rete

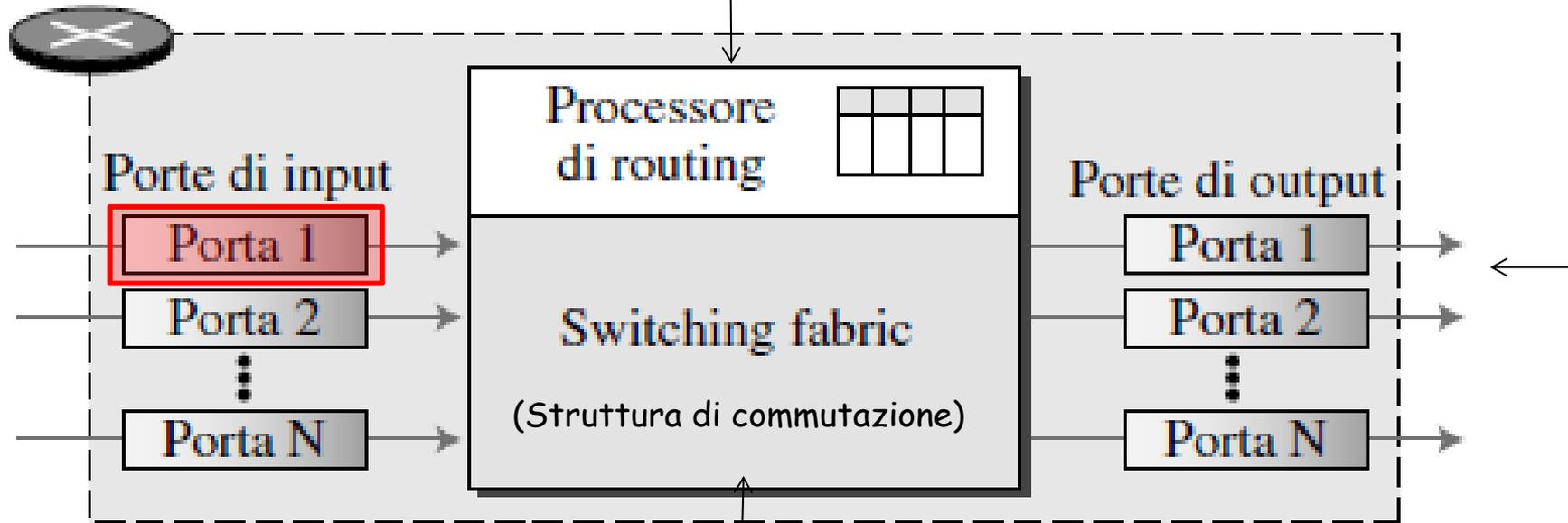
Che cosa si trova all'interno di un router?

# Funzioni del router



# Architettura del router

Implementa funzionalità del livello di rete  
Esegue table lookup (a volte implementata nelle porte)

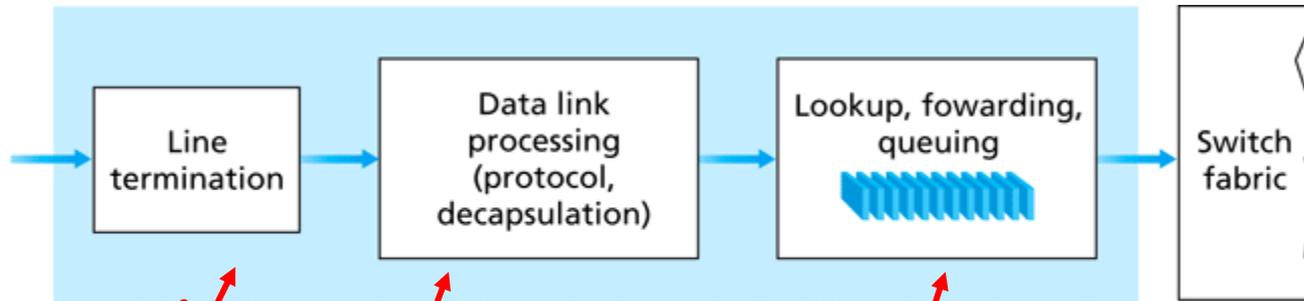


Implementa livello fisico e di collegamento:  
I bit vengono ricostruiti a partire dal segnale ricevuto, si estrae il frame e quindi il datagramma, se integro si passa al livello di rete

Sposta datagrammi dalla coda di input alla coda di output

Accoda datagrammi che vengono incapsulati in frame e tradotti in segnali da trasmettere attraverso il livello fisico

# Porte d'ingresso



**Livello fisico:**  
ricezione a livello di bit  
(terminazione elettrica)

**Livello di collegamento:**  
Es. Ethernet

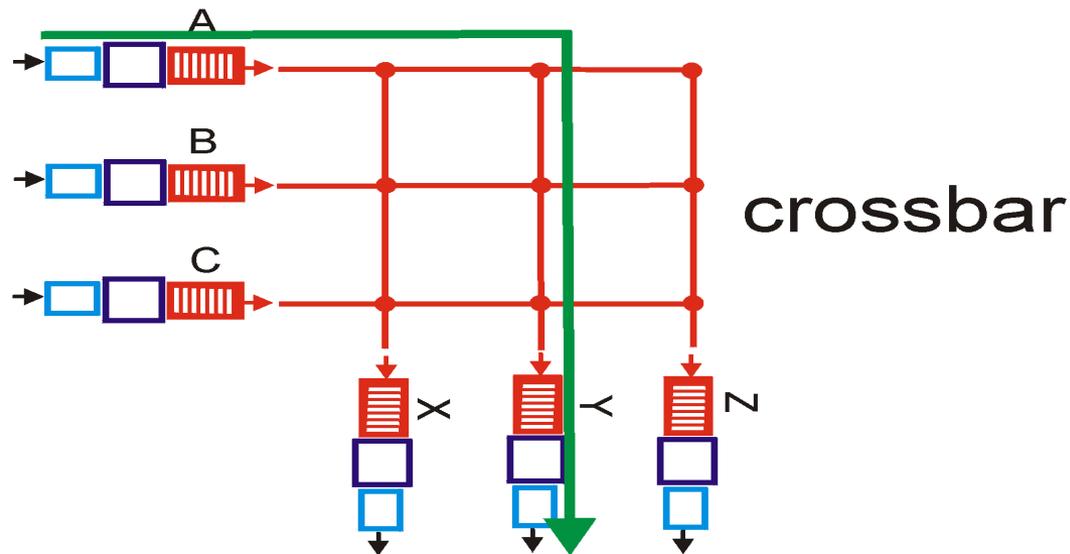
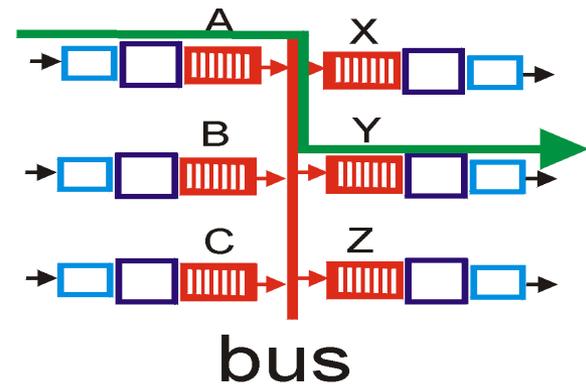
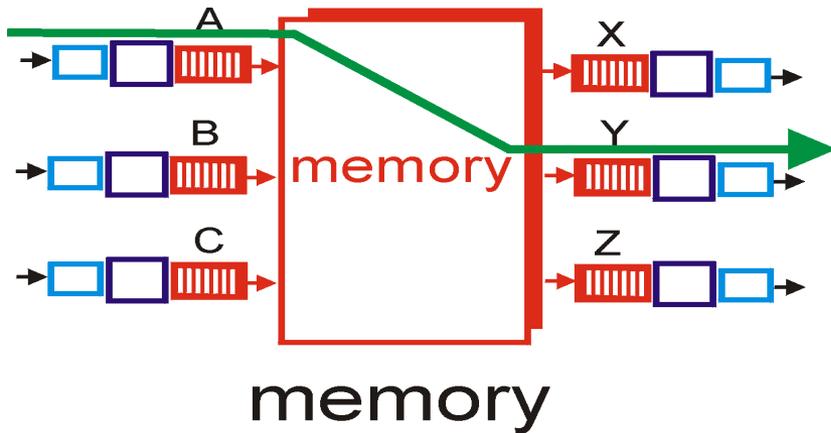
## **Commutazione decentralizzata:**

- ❑ Determina la porta d'uscita dei pacchetti utilizzando le informazioni della tabella d'inoltro (c'è una copia della tabella memorizzata nella porta di ingresso)
- ❑ Obiettivo: completare l'elaborazione allo stesso **tasso della linea** (evitare colli di bottiglia)
- ❑ Accodamento: se il tasso di arrivo dei datagrammi è superiore a quello di inoltro
- ❑ Una volta determinata la porta di uscita il pacchetto verrà inoltrato alla struttura di commutazione

# Ricerca nella tabella di inoltra

- ❑ Deve essere veloce (possibilmente stesso tasso della linea) per evitare accodamenti
- ❑ Tabella implementata in una **struttura ad albero**
  - Ogni livello dell'albero corrisponde a un bit dell'indirizzo di destinazione
  - Per cercare un indirizzo si comincia dalla radice dell'albero
    - Se 0 allora sottoalbero di sinistra
    - Se 1 allora sottoalbero di destra
  - Ricerca in N passi dove N è il numero di bit nell'indirizzo
- ❑ Argomento molto studiato, per approfondimenti  
**Survey and taxonomy of IP address lookup algorithms, IEEE Networks 2001**

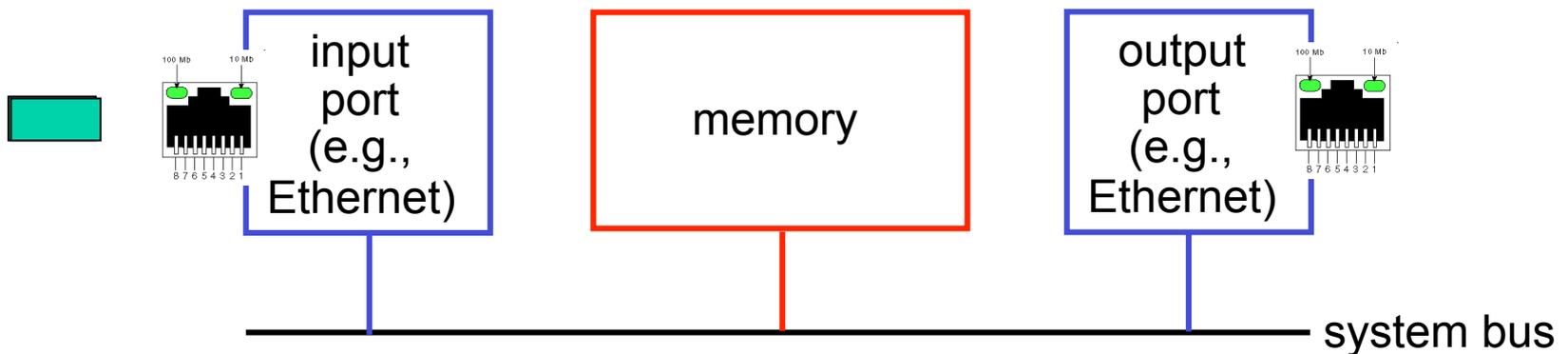
# Tre tecniche di commutazione



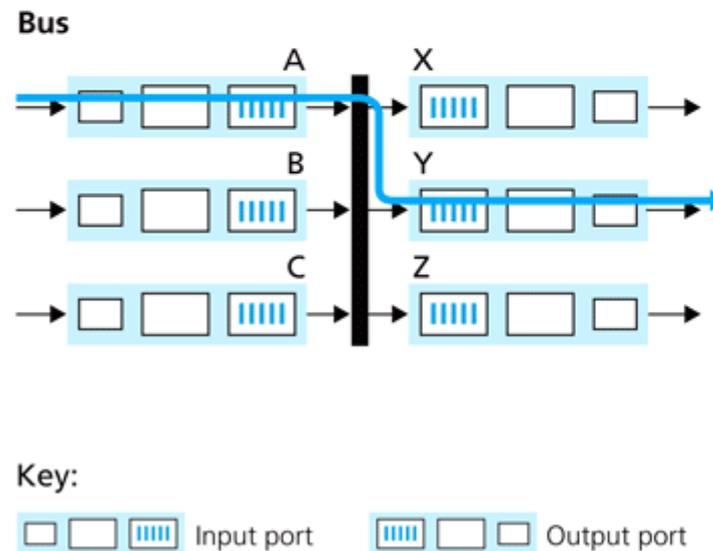
# Commutazione in memoria

## Prima generazione di router:

- Erano tradizionali calcolatori e la commutazione era effettuata sotto il controllo diretto della CPU.
- Il pacchetto veniva copiato nella memoria del processore.
- I pacchetti venivano trasferiti dalle porte d'ingresso a quelle d'uscita



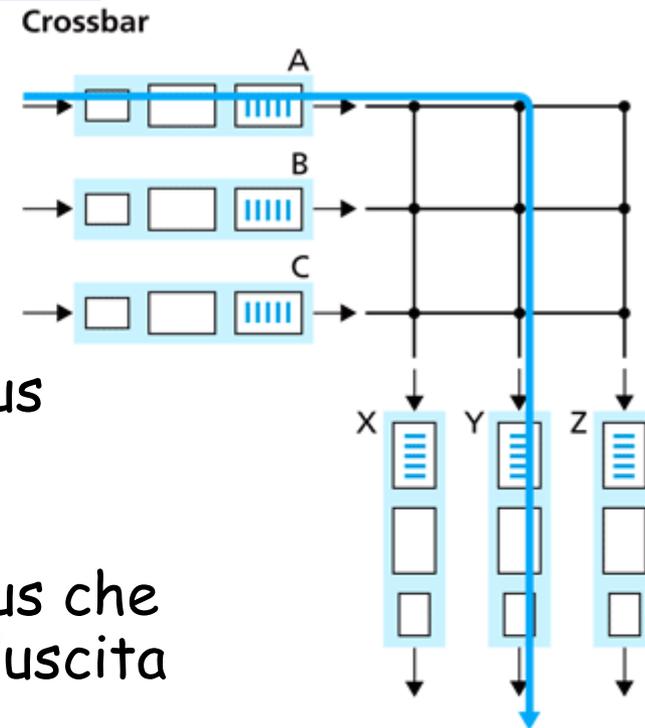
# Commutazione tramite bus



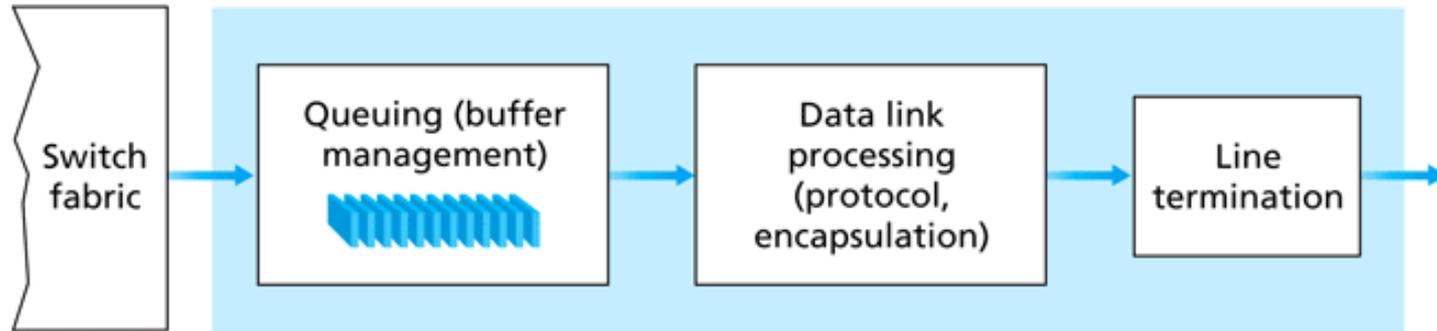
- ❑ Le porte d'ingresso trasferiscono un pacchetto direttamente alle porte d'uscita su un bus condiviso, senza intervento del processore di instradamento.
- ❑ Si può trasferire un solo pacchetto alla volta
- ❑ I pacchetti che arrivano e trovano il bus occupato vengono accodati alla porta di ingresso
- ❑ **Contesa per il bus:** la larghezza di banda della commutazione è limitata da quella del bus.
- ❑ Cisco 5600 opera con bus da 32 Gbps: è sufficiente per router che operano in reti d'accesso o in quelle aziendali

# Commutazione attraverso rete d'interconnessione

- ❑ Supera il limite di banda di un singolo bus condiviso.
- ❑ Un **crossbar switch** è una rete d'interconnessione che consiste di  $2n$  bus che collegano  $n$  porte d'ingresso a  $n$  porte d'uscita
- ❑ Tendenza attuale: frammentazione dei pacchetti IP a lunghezza variabile in celle di lunghezza fissa (riasmblati nella porta di uscita).
- ❑ Switch Cisco 12000: usano una rete d'interconnessione che raggiunge i 60 Gbps nella struttura di commutazione.



# Porte d'uscita



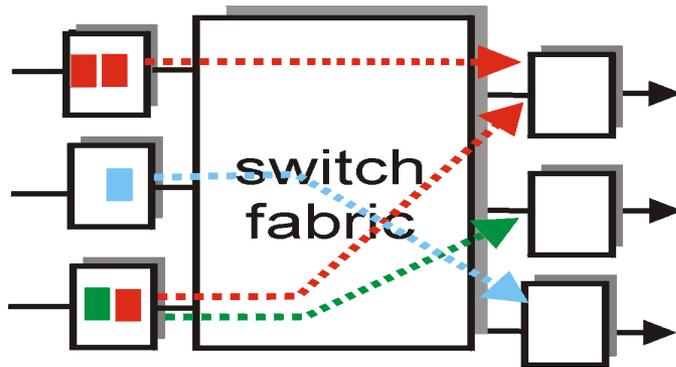
- ❑ **Funzionalità di accodamento:** quando la struttura di commutazione consegna pacchetti alla porta d'uscita a una frequenza che supera quella del collegamento uscente.
- ❑ **Schedulatore di pacchetti:** stabilisce in quale ordine trasmettere i pacchetti accodati.

# Dove si verifica l'accodamento?

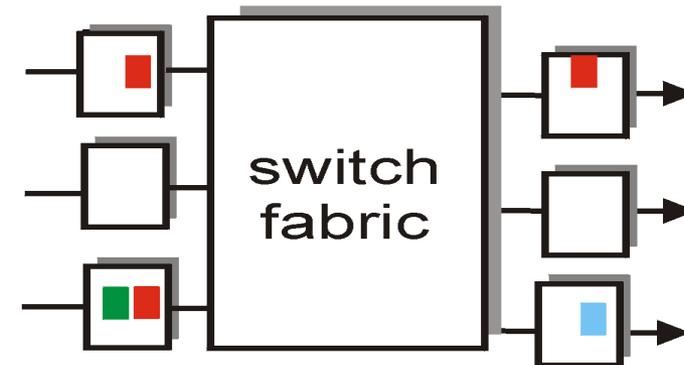
- Sia nelle porte di **ingresso** che nelle porte di **uscita**
  - Velocità di commutazione: frequenza alla quale tale struttura può trasferire i pacchetti dalle porte di ingresso a quelle di uscita
  - **Accodamento nelle porte di ingresso** - quando la struttura di commutazione ha una velocità inferiore a quello delle porte di ingresso (per non avere accodamento la velocità di commutazione dovrebbe essere  $n^*$  velocità della linea di ingresso)
  - **Accodamento nelle porte di uscita**
    - Quando la struttura di commutazione ha un rate superiore alla porta di uscita
    - Quando troppi pacchetti vanno sulla stessa porta di uscita

# Accodamento su porte di ingresso

- Oltre alla velocità inferiore della struttura di commutazione
- **Blocco in testa alla fila (HOL: head-of-the-line blocking):** un pacchetto nella coda d'ingresso deve attendere il trasferimento (anche se la propria destinazione è libera) in quanto risulta bloccato da un altro pacchetto in testa alla fila.
- **Se le code diventano troppo lunghe, i buffer si possono saturare e quindi causare una perdita di pacchetti!**



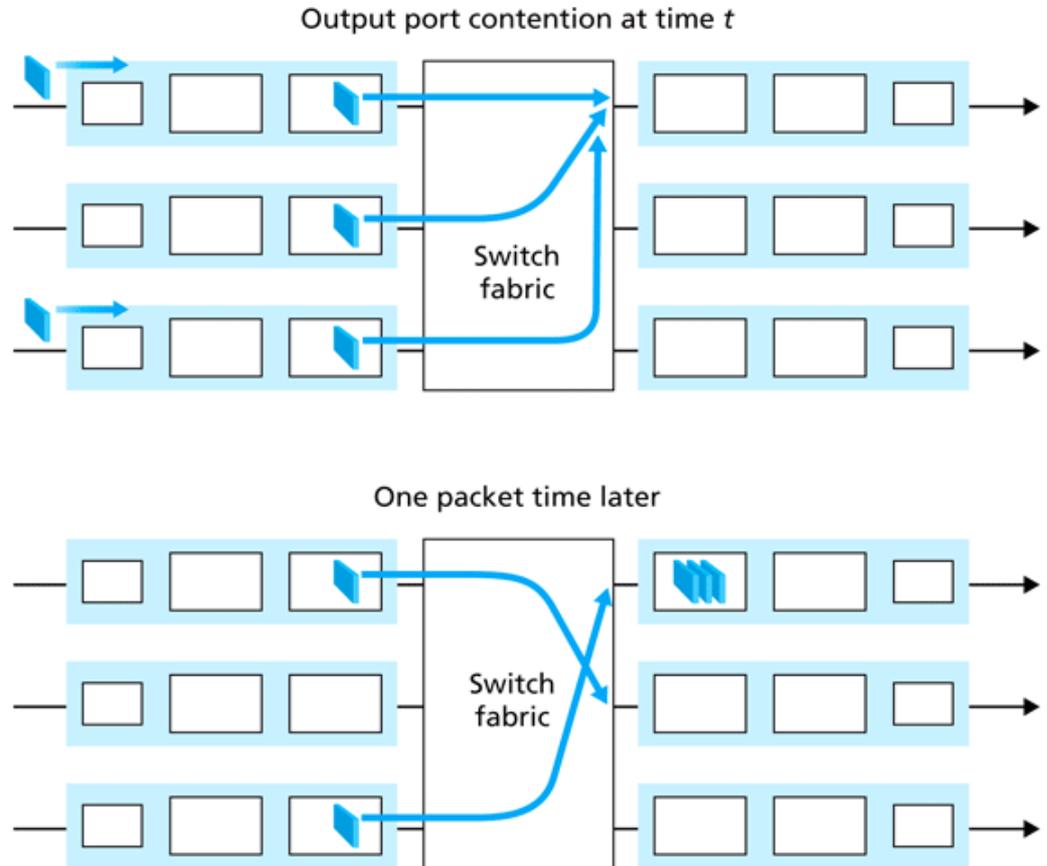
output port contention  
at time  $t$  - only one red  
packet can be transferred



green packet  
experiences HOL blocking

# Accodamento sulle porte di uscita

- Se la struttura di commutazione ha un rate superiore a quello della porta di uscita, si può verificare un accodamento.
- Se troppi pacchetti vanno sulla stessa uscita
- Se le code diventano troppo lunghe, i buffer si possono saturare e quindi causare una perdita di pacchetti!

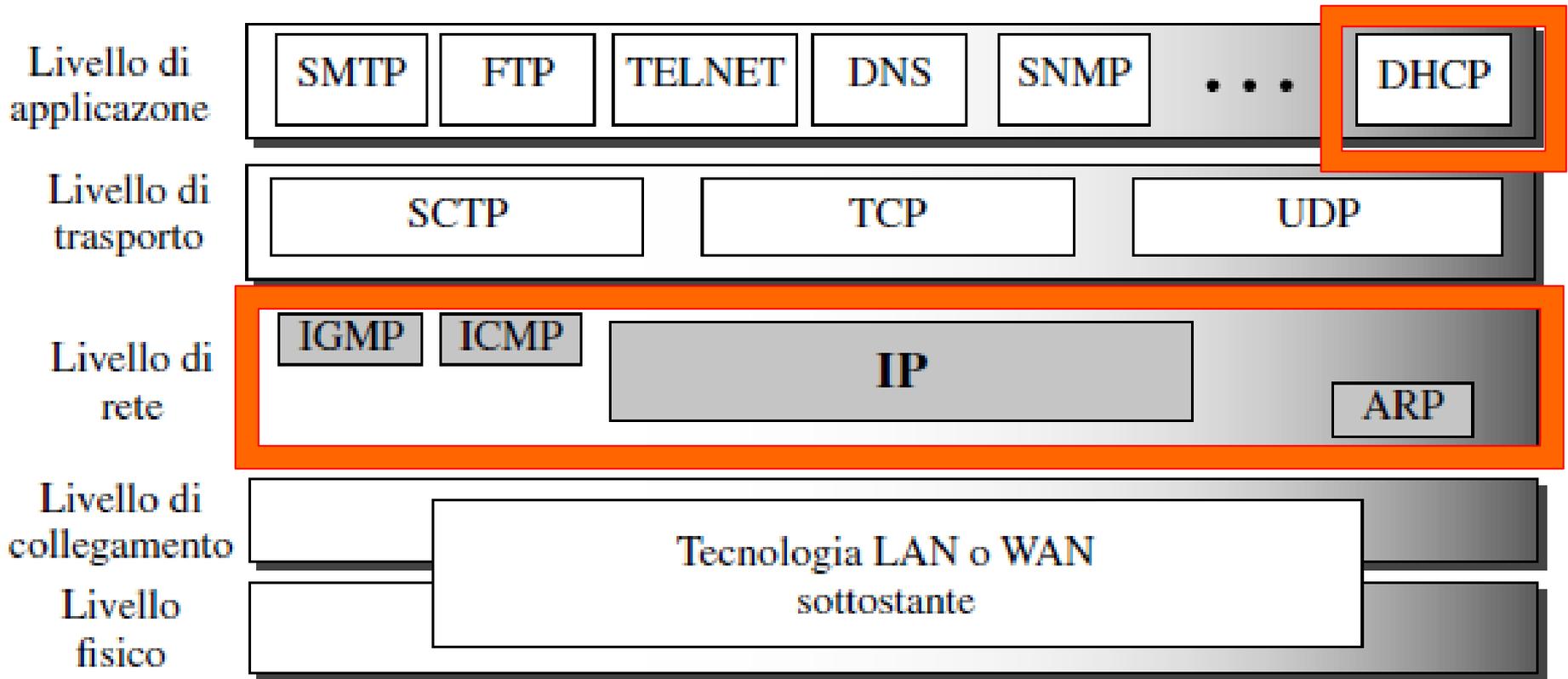


# Quale deve essere la capacità dei buffer?

- Per diversi anni si è seguita la regola definita in RFC 3439: la quantità di buffering dovrebbe essere uguale a una media del tempo di andata e ritorno (RTT ad esempio 250 msec) per la capacità del collegamento  $C$ 
  - Es.:  $C =$  collegamento da 10 Gbps: buffer 2.5 Gbit
- Attuali raccomandazioni dicono che la quantità di buffering necessaria per  $N$  flussi TCP è:

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

# Protocolli del livello di rete



IP: Internet Protocol v4 (anche v6)

IGMP: Internet Group Management Protocol (multicasting)

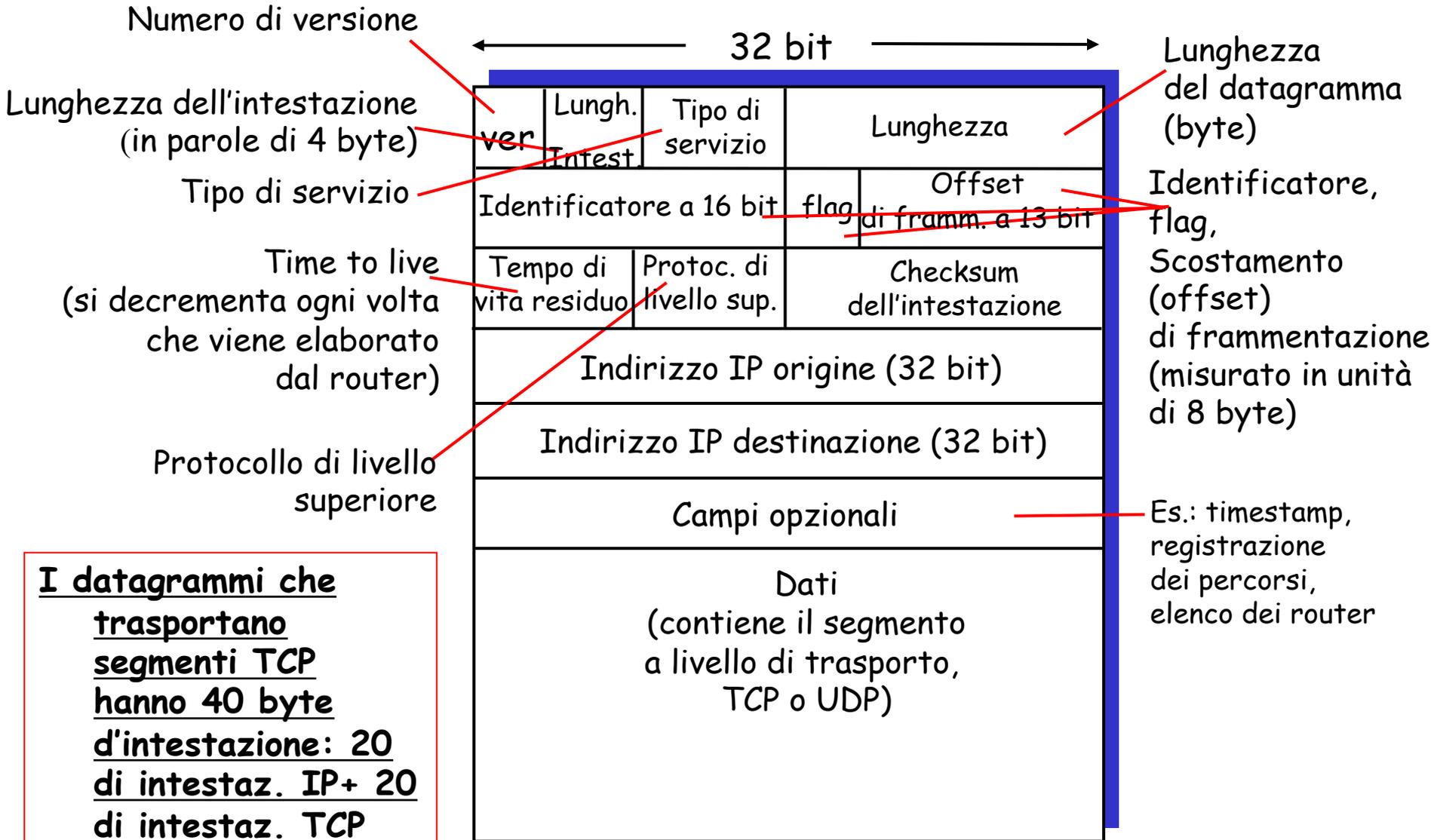
ICMP: Internet Control Message Protocol (gestione errori)

ARP: Address Resolution Protocol (associazione indirizzo IP - ind. collegamento) <sup>36</sup>

# Internet Protocol (IPv4)

- ❑ Responsabile della suddivisione in pacchetti, dell'inoltro (forwarding), e della consegna dei datagrammi al livello di rete (host to host)
- ❑ Inaffidabile, senza connessione, basato su datagrammi
- ❑ Offre un servizio di consegna best effort

# Formato dei datagrammi



# Formato dei datagrammi

- ❑ **Numero di versione:** consente al router la corretta interpretazione del datagramma
  - 4: IPv4
  - 6: IPv6
- ❑ **Lunghezza dell'intestazione:** poichè un datagramma IP può contenere un numero variabile di opzioni (incluse nell'intestazione), questi bit indicano dove inizia il campo dati. (Intestazione senza opzione = 20 byte)
- ❑ **Tipo di servizio:** serve per distinguere diversi datagrammi con requisiti di qualità del servizio diverse (realtime)
- ❑ **Lunghezza del datagramma:** rappresenta la lunghezza totale del datagramma IP inclusa l'intestazione (in byte). In genere non superiore ai 1500 byte. Serve per capire se il pacchetto è arrivato completamente

# Formato dei datagrammi

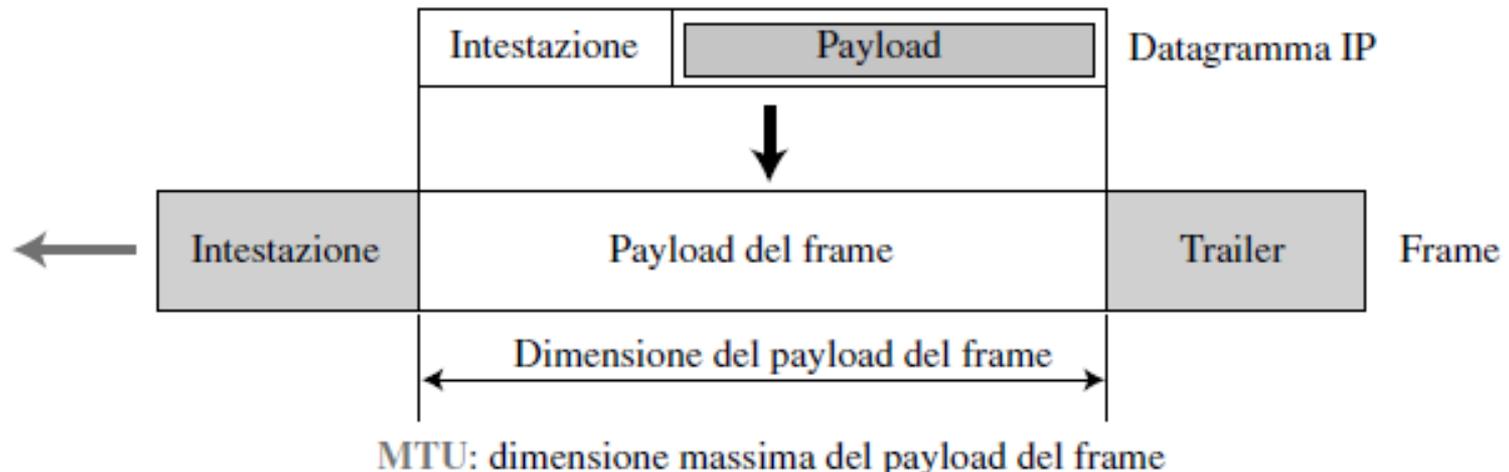
- ❑ **Identificatore, flag e offset di frammentazione:** questi tre campi servono per gestire la frammentazione dei pacchetti (Ipv6 non prevede frammentazione)
- ❑ **Tempo di vita:** o time to live (TTL) è incluso per assicurare che i datagrammi non restino in circolazione per sempre nella rete (in caso per esempio di instradamento ciclico). Il campo viene decrementato a ogni hop e il datagramma viene eliminato in caso il suo TTL = 0
- ❑ **Protocollo:** indica il protocollo a livello di trasporto al quale va passato il datagramma. Questo campo è utilizzato solo quando il datagramma raggiunge la destinazione finale.
  - 6: TCP
  - 17: UDP
  - 1: ICMP
  - 2: IGMP
  - 89: OSPF

# Formato dei datagrammi

- ❑ **Checksum dell'intestazione:** consente ai router di rilevare errori sui datagrammi ricevuti
  - Calcolata solo sull'intestazione
  - Ricalcolata nei router intermedi (TTL e frammentazione)
  - Checksum UDP/TCP è calcolata su intero segmento
- ❑ **Indirizzi IP di origine e destinazione:** inseriti dall'host che crea il datagramma (dopo aver effettuato una ricerca DNS)
- ❑ **Opzioni:** campi che consentono di estendere l'intestazione IP
  - Usate per test o debug della rete
- ❑ **Dati:**
  - il campo dati contiene il segmento di trasporto da consegnare alla destinazione
  - può trasportare altri tipi di dati, quali i messaggi ICMP, IGMP, etc..

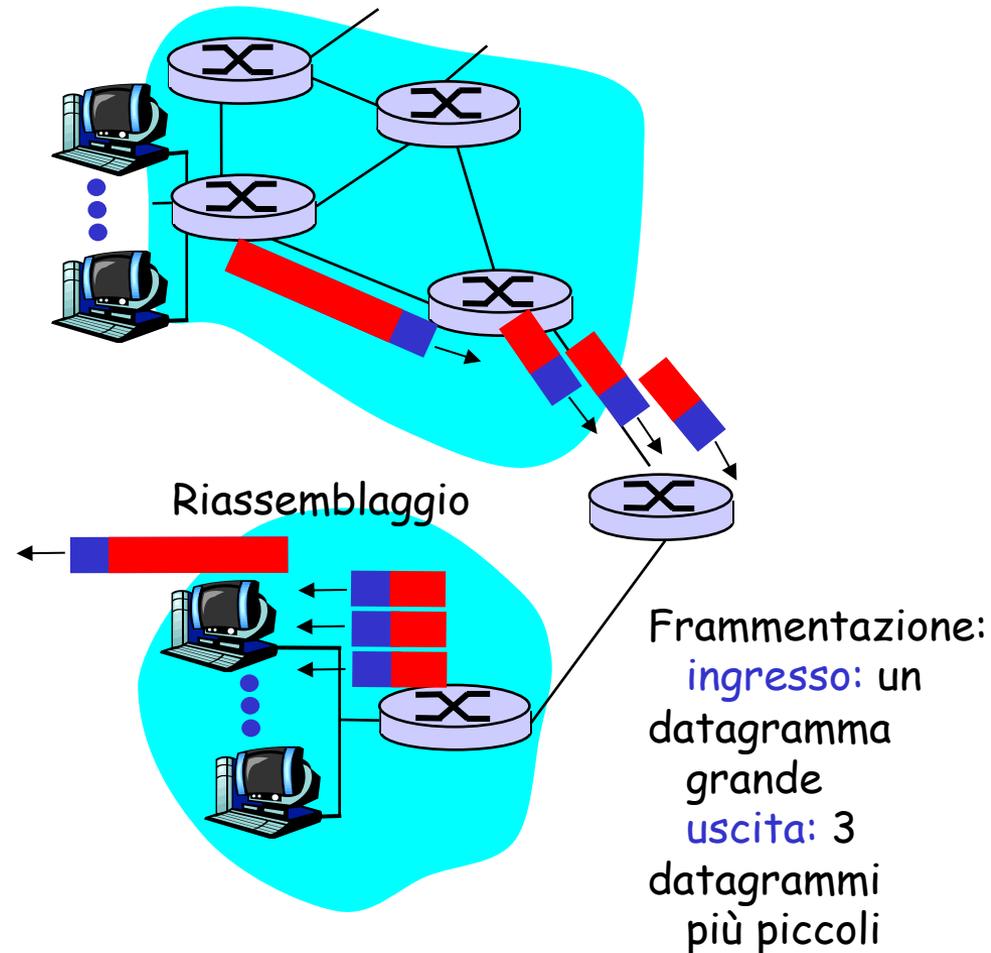
# Frammentazione

- ❑ Un datagramma IP può dover viaggiare attraverso varie reti, ognuna con caratteristiche diverse. Ogni router estrae il datagramma dal frame, lo elabora e lo incapsula in un nuovo frame.
- ❑ La Maximum Transfer Unit (MTU) è la massima quantità di dati che un frame a livello di collegamento può trasportare.
- ❑ La MTU varia in base alla tecnologia



# Frammentazione dei datagrammi IP

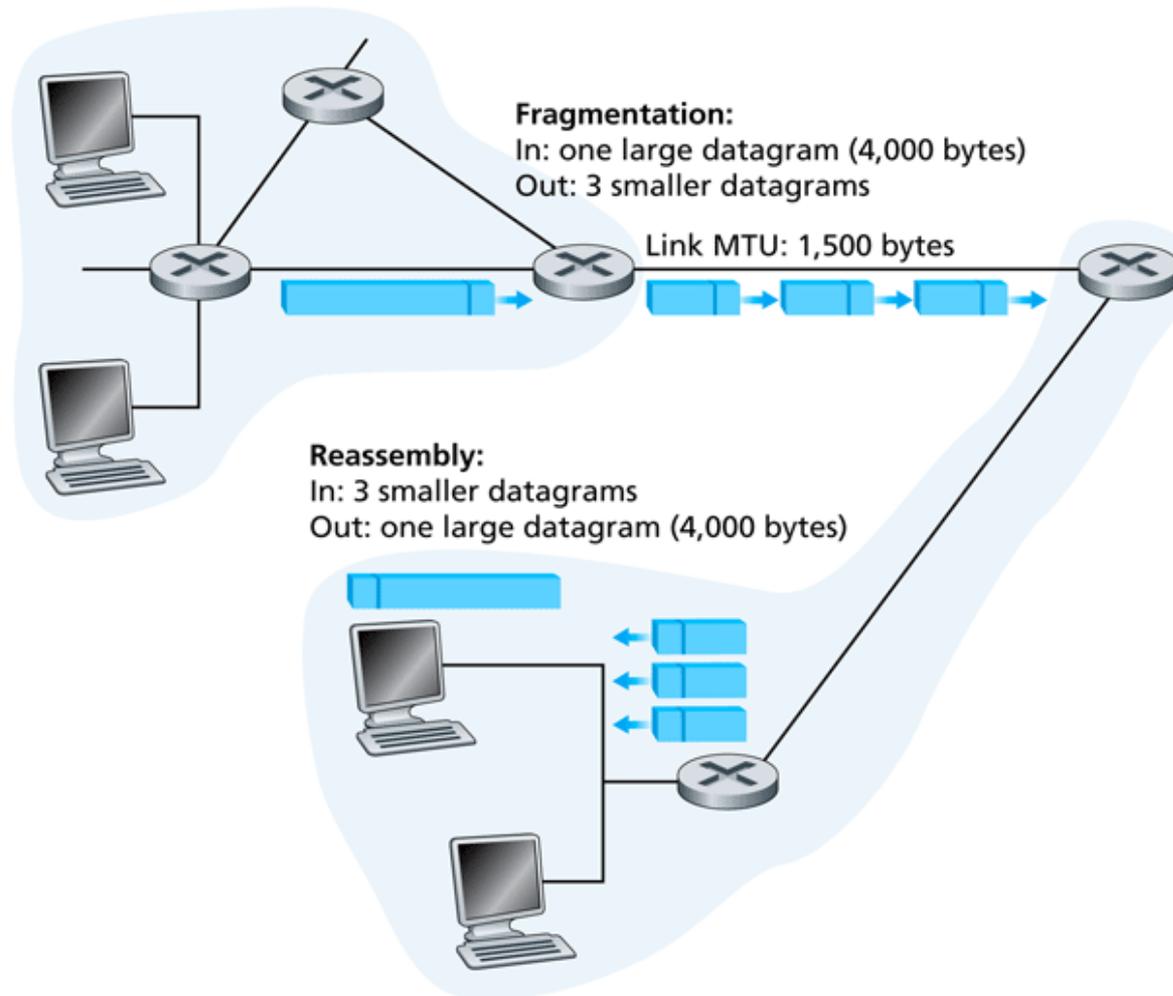
- Differenti tipi di link, differenti MTU (es. 1500b, 1492b).
- Datagrammi IP grandi vengono suddivisi ("frammentati") in datagrammi IP più piccoli.
  - Un datagramma viene frammentato.
  - I frammenti saranno riassemblati solo una volta raggiunta la destinazione
  - I frammenti devono essere riassemblati prima di raggiungere il livello di trasporto
  - I bit dell'intestazione IP sono usati per identificare e ordinare i frammenti



# Bit nell'intestazione per la frammentazione

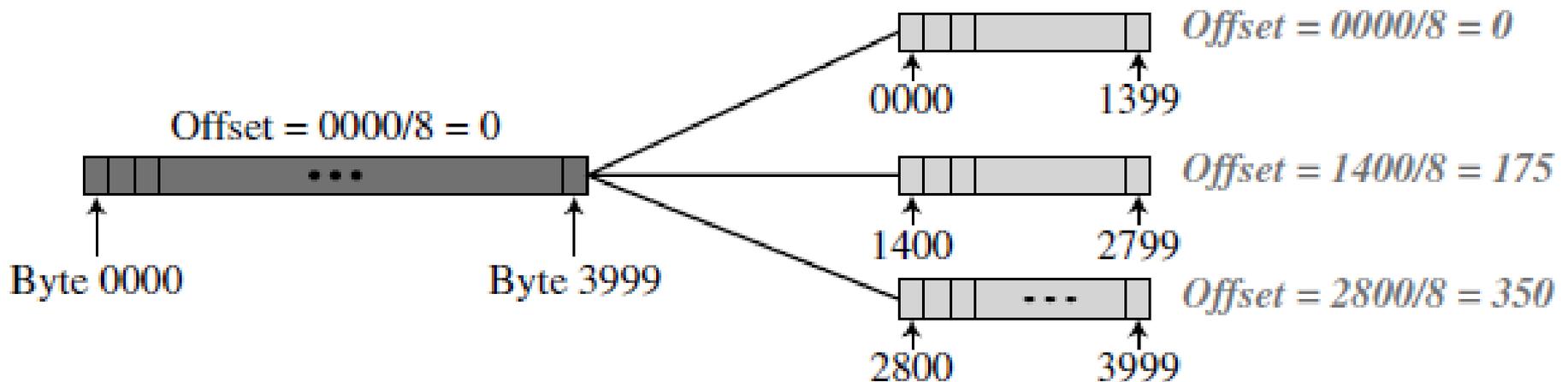
- ❑ Quando un host di destinazione riceve una serie di datagrammi dalla stessa origine deve
  - individuare i frammenti
  - determinare quando ha ricevuto l'ultimo
  - stabilire come debbano essere riassemblati
- ❑ **Identificazione (16 bit)**: identificativo associato a ciascun datagramma al momento della creazione (unico per tutti i frammenti)
  - IP + identificazione identificano in modo univoco un datagramma
- ❑ **Flag**: 3 bit
  1. Riservato
  2. Do not fragment: 1 non frammentare, 0 si può frammentare
  3. More fragments (M): 1 frammenti intermedi, 0 ultimo frammento
- ❑ **Offset (Scostamento laterale)**: specifica l'ordine del frammento all'interno del datagramma originario

# Frammentazione e riassetblaggio



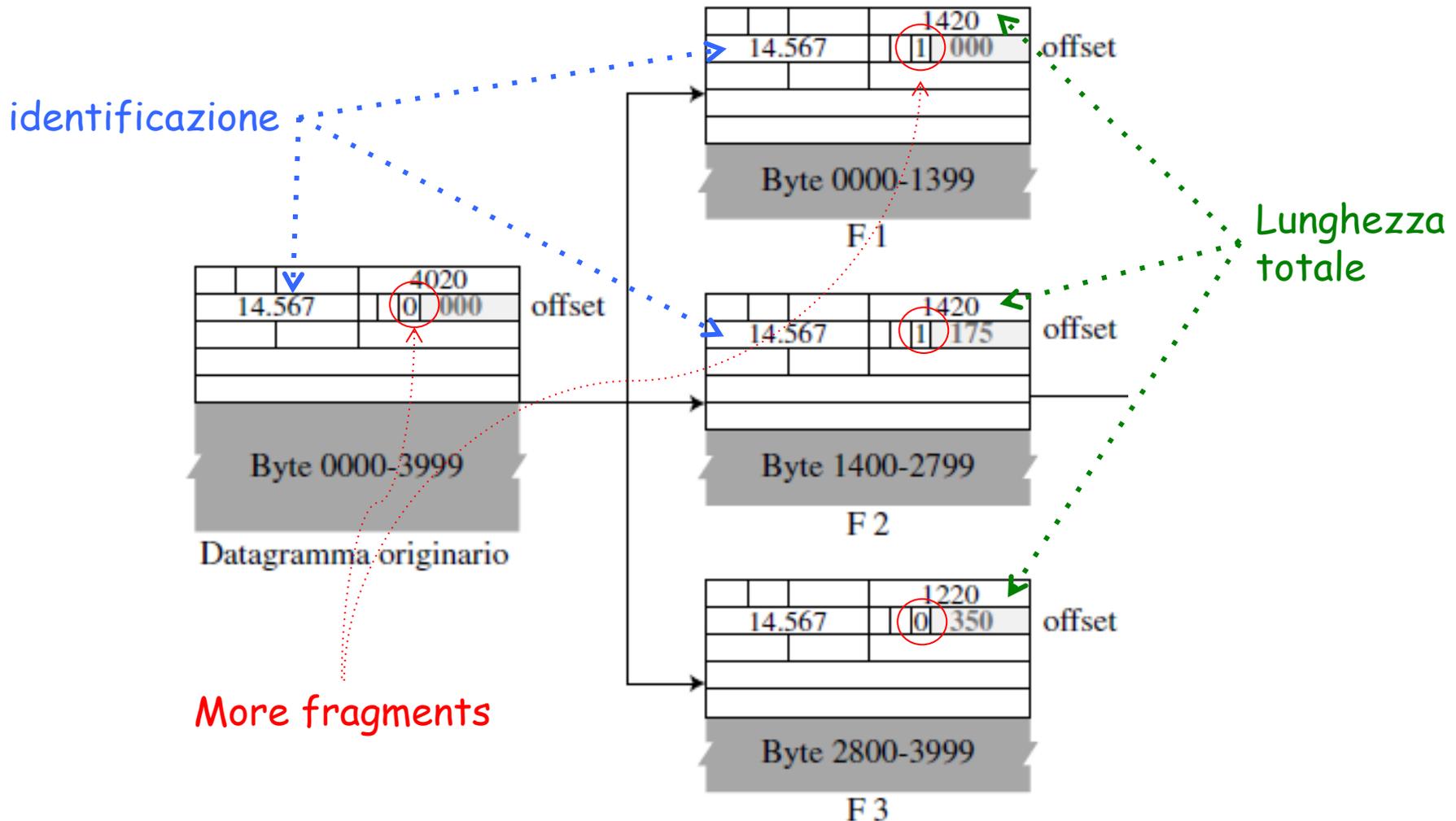
# Frammentazione: offset (13 bit)

Payload di un datagramma con un dimensione di 4000 byte suddiviso in tre frammenti



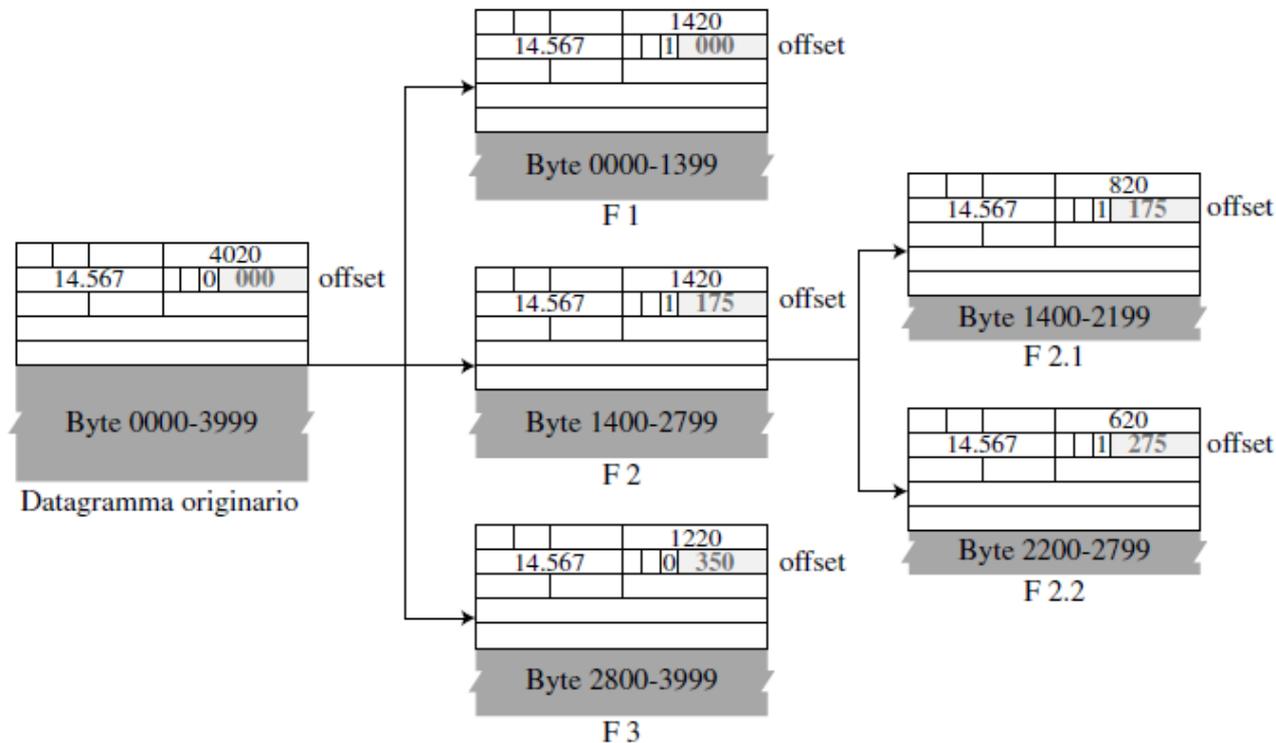
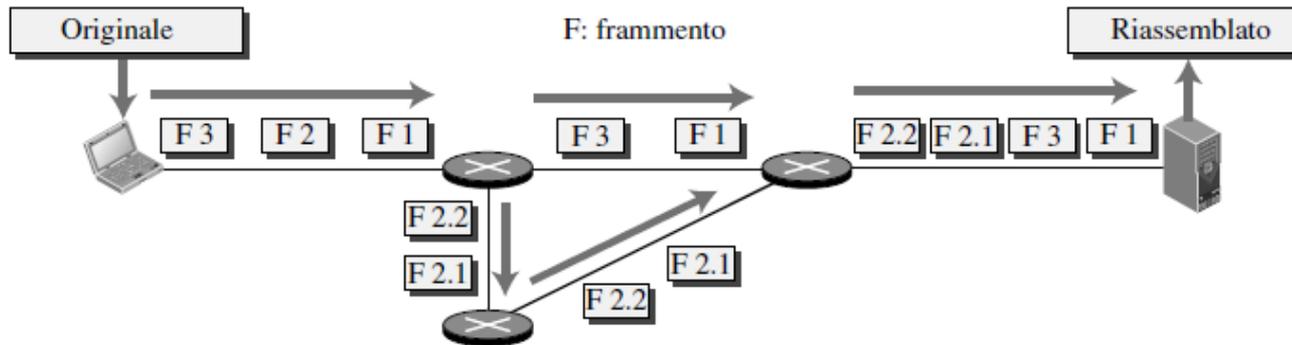
N.B. offset dei dati nel datagramma originale è misurato in unità di 8 byte

# Esempio di frammentazione



In totale vengo trasferiti 40 byte in più: ci sono due header aggiuntivi!!! 4-47

# Frammentazione di un frammento



# Riassemblaggio a destinazione

- ❑ Il primo frammento ha un valore del campo offset pari a 0
- ❑ L'offset del secondo frammento si ottiene dividendo per 8 la lunghezza del primo frammento (esclusa l'intestazione)
- ❑ Il valore del terzo frammento si ottiene dividendo per 8 la somma della lunghezza del primo e del secondo frammento (escluse le intestazioni)
- ❑ ...
- ❑ L'ultimo frammento ha il bit  $M$  impostato a 0<sub>4-49</sub>

# Conclusioni

- ❑ Forwarding (circuito vs pacchetto)
- ❑ Struttura di un router
- ❑ Protocollo IP
  - Intestazione dei datagrammi
  - Frammentazione