



Marco Listanti

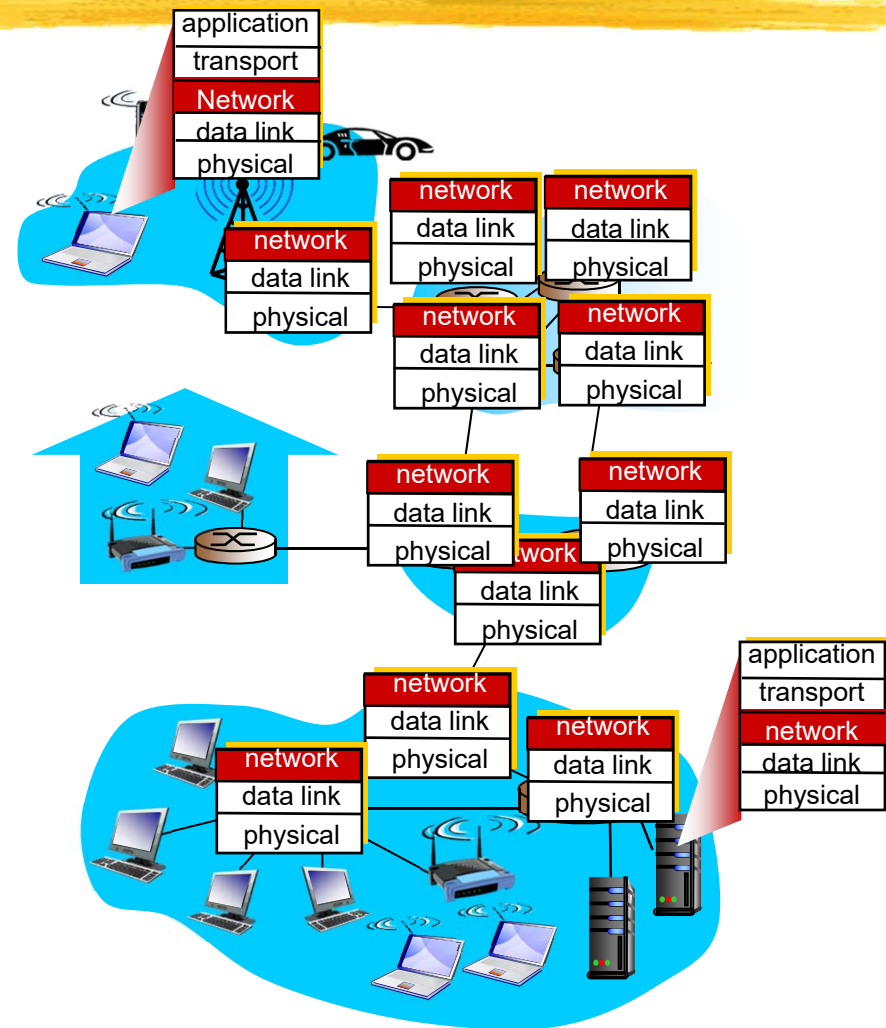
# Strato di rete (parte 1)

## "Reti a pacchetto e Architetture di router"



# Strato di rete (Network Layer)

- Lo stato di rete trasferisce i segmenti, hop to hop, lungo il percorso tra gli host di sorgente e di destinazione
- Le funzioni dello strato di rete sono eseguite da tutti i router intermedi lungo il percorso
  - Incapsula i segmenti in pacchetti





# Funzioni del livello di rete

## ■ Routing (instradamento)

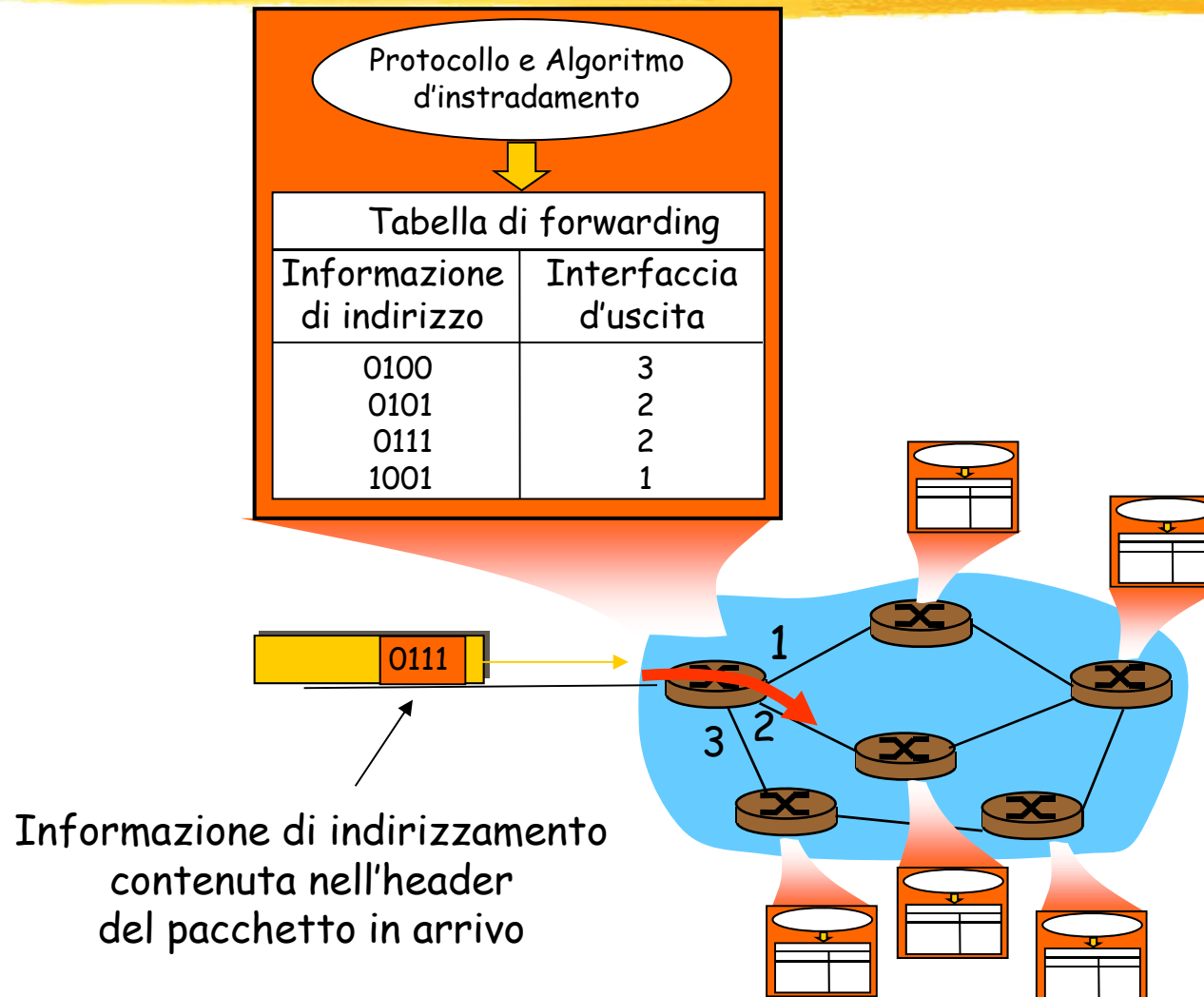
- Funzione decisionale
- Funzione del piano di controllo (**Control Plane**)
- Determina il percorso di rete seguito dai pacchetti tra gli host di sorgente e di destinazione
  - in ciascun router determina l'interfaccia di uscita su cui deve essere rilanciato un pacchetto
- Utilizza protocolli e algoritmi specifici

## ■ Forwarding (rilancio)

- Funzione attuativa
- Funzione del piano dati (**Data Plane**)
- Trasferisce i pacchetti da un interfaccia di ingresso di un router verso l'opportuna interfaccia di uscita individuata dalla funzione di routing



# Routing e forwarding





# Data plane e control plane

## ■ Data plane

- Comprende le funzioni relative al trattamento dei pacchetti
- Determina le modalità di rilancio dei pacchetti (**forwarding**)
- Funzioni eseguite localmente dai router (**router local logic**)

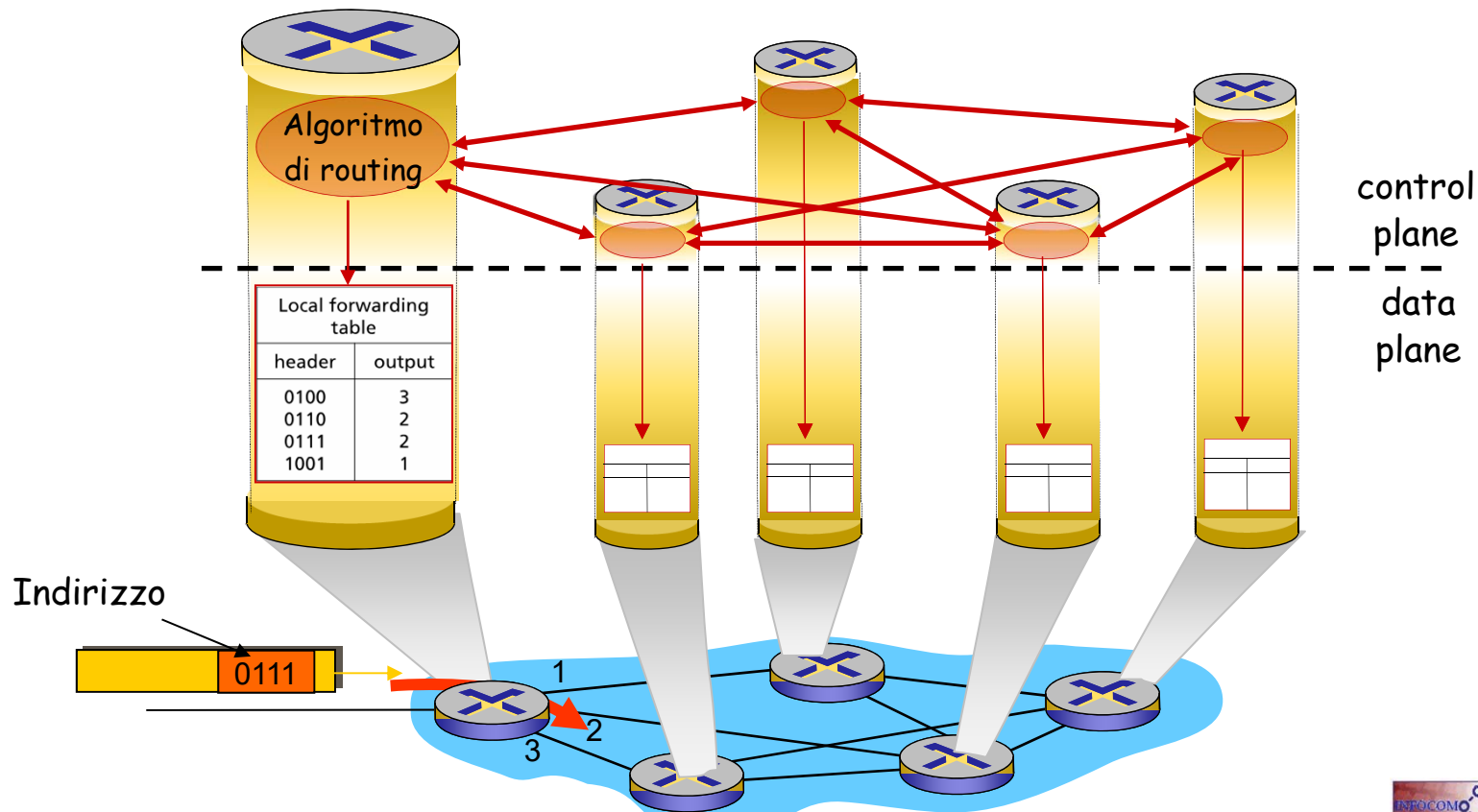
## ■ Control Plane

- Comprende le funzioni logiche necessarie al funzionamento della rete
- Funzioni eseguite dalla rete nel suo complesso (**network-wide logic**)
- Determina il percorso che i pacchetti devono seguire in rete (**instradamento**)
- Due possibili architetture
  - **Distribuita**: algoritmi di routing implementati in ogni router router
  - **Centralizzata**: Software Defined Networking (SDN)



# Control plane distribuito

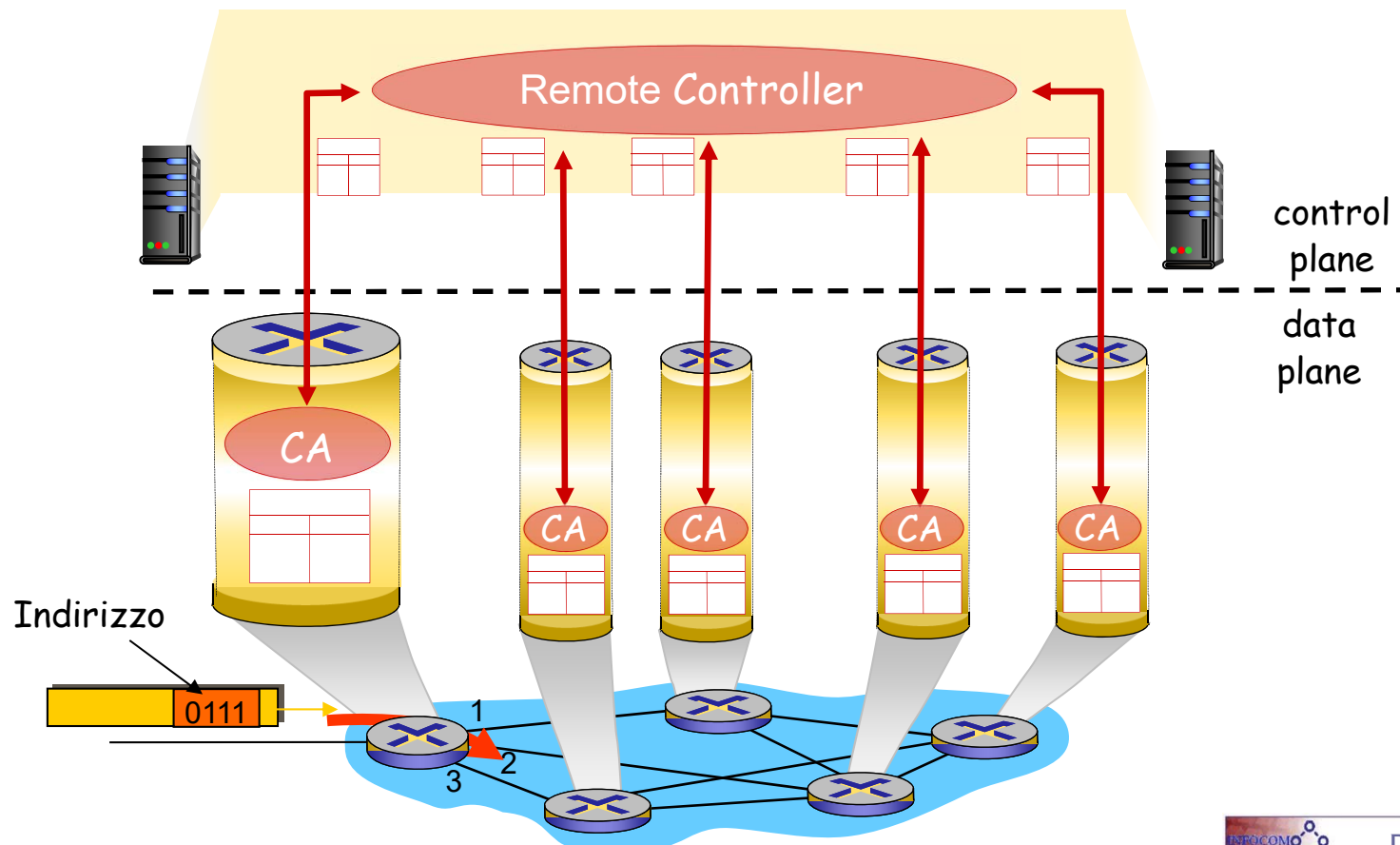
- Le funzioni sono eseguite tramite la cooperazione tra singole componenti implementate in ogni router





# Control plane centralizzato

- Un Controller remoto interagisce con i Control Agent (CA) presenti in ogni router





# Tipologie di servizio di rete

## ■ Servizio senza connessione

- reti a "datagramma"
- i pacchetti sono inviati da una sorgente senza un preventivo accordo sia con la destinazione sia con la rete
- i pacchetti sono trattati dalla rete e da ciascun nodo come entità indipendenti
- l'instradamento è deciso pacchetto per pacchetto
- i router hanno un funzionamento "stateless"

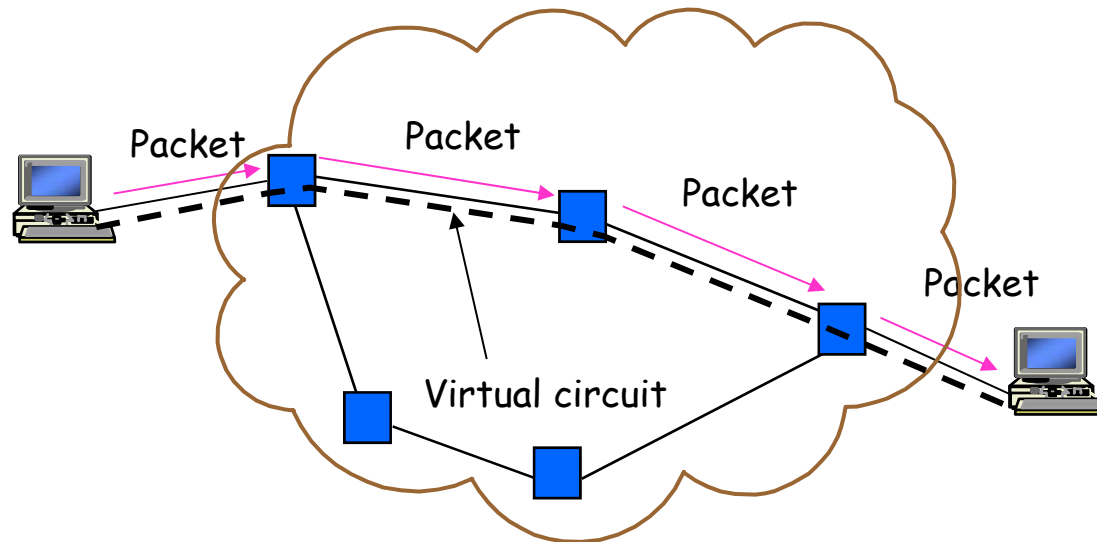
## ■ Servizio connection-oriented

- reti a "circuito virtuale" (VC) o "Label Switching" (es. MPLS)
- prima dell'invio dei pacchetti viene instaurata una **connessione di rete**
- il cammino (path) di instradamento dei pacchetti è deciso al momento dell'instaurazione della connessione
- i nodi hanno un funzionamento "stateful"
  - i nodi mantengono informazioni sullo stato delle connessioni





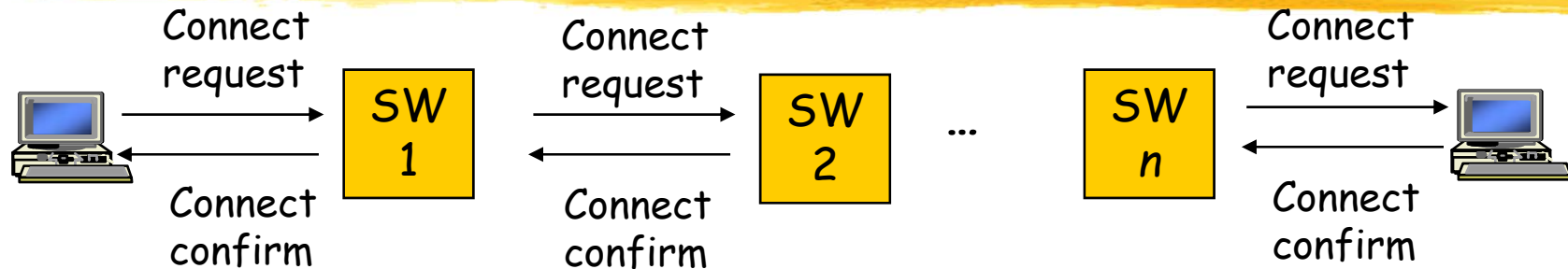
# Packet Switching - Virtual Circuit



- E' necessaria una fase di set-up della connessione
- E' necessario un protocollo di segnalazione
- Tutti i pacchetti seguono lo stesso path in rete
  - Consegna in sequenza dei pacchetti
- L'informazione di indirizzamento contenuta nell'header di ogni pacchetto è l'**identificatore della connessione** a cui appartiene
  - l'identificazione della connessione avviene "per link"



# Connection Setup



## ■ I messaggi di segnalazione

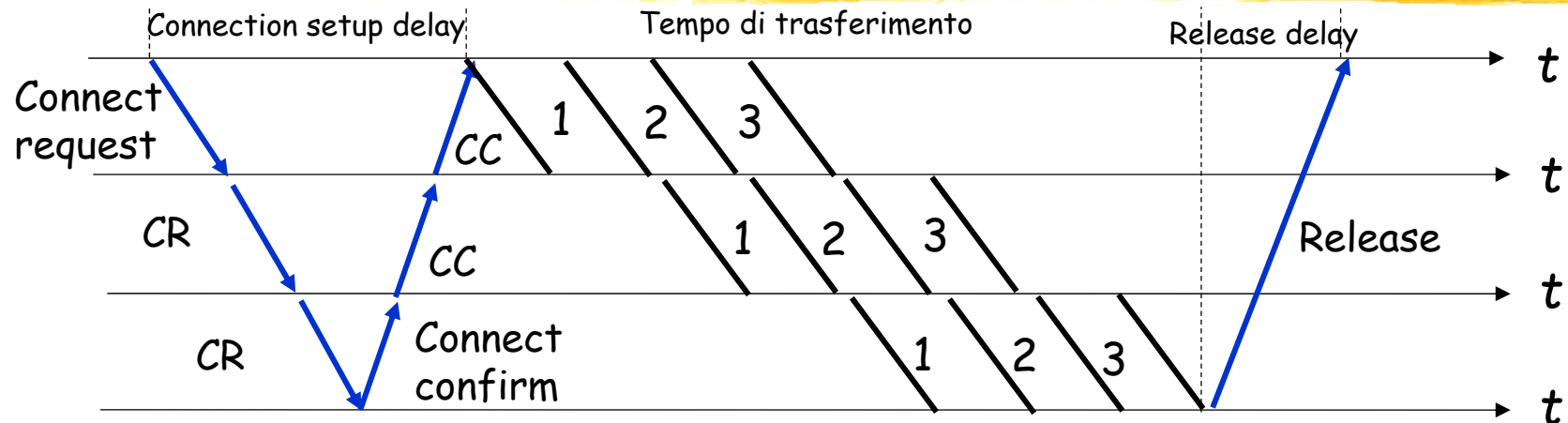
- sono trasmessi lungo il path della connessione
- determinano in ogni nodo l'esecuzione della funzione di routing che identifica il nodo successivo sul path
- inizializzano le tabelle di forwarding nei nodi

## ■ La connessione è identificata su ogni link da un "local tag" (**Virtual Circuit Identifier - VCI**)

- Ogni nodo (**switch**) memorizza la relazione tra input VCI, output VCI e interfaccia di uscita nella tabella di forwarding
- Una volta che le tabelle di forwarding sono inizializzate i pacchetti possono essere trasmessi in rete



# Connection Setup Delay



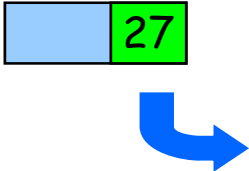
- Il ritardo di instaurazione della connessione (**connection setup delay**) si somma al ritardo di transito dei pacchetti
- Efficienza ( $\rho$ )

$$\rho = \frac{\text{transfer delay}}{\text{setup delay} + \text{transfer delay} + \text{release delay}}$$

- Il ritardo aggiuntivo è
  - tollerabile se è inferiore al ritardo di trasferimento dei dati
  - inaccettabile se devono essere trasferiti pochi pacchetti



# Virtual Circuit Forwarding Table



Input VCI	Output port	Output VCI
12	13	44
15	15	23
27	13	16
58	7	34

- Ogni porta di ingresso di un router ha una propria forwarding table
- Si utilizza il VCI contenuto nell'header del pacchetto come indice di accesso della tabella
- Si individua il record corrispondente al VCI, si legge la porta di uscita e il valore del VCI sul link d'uscita
- Il valore del VCI d'uscita è scritto nell'header del pacchetto



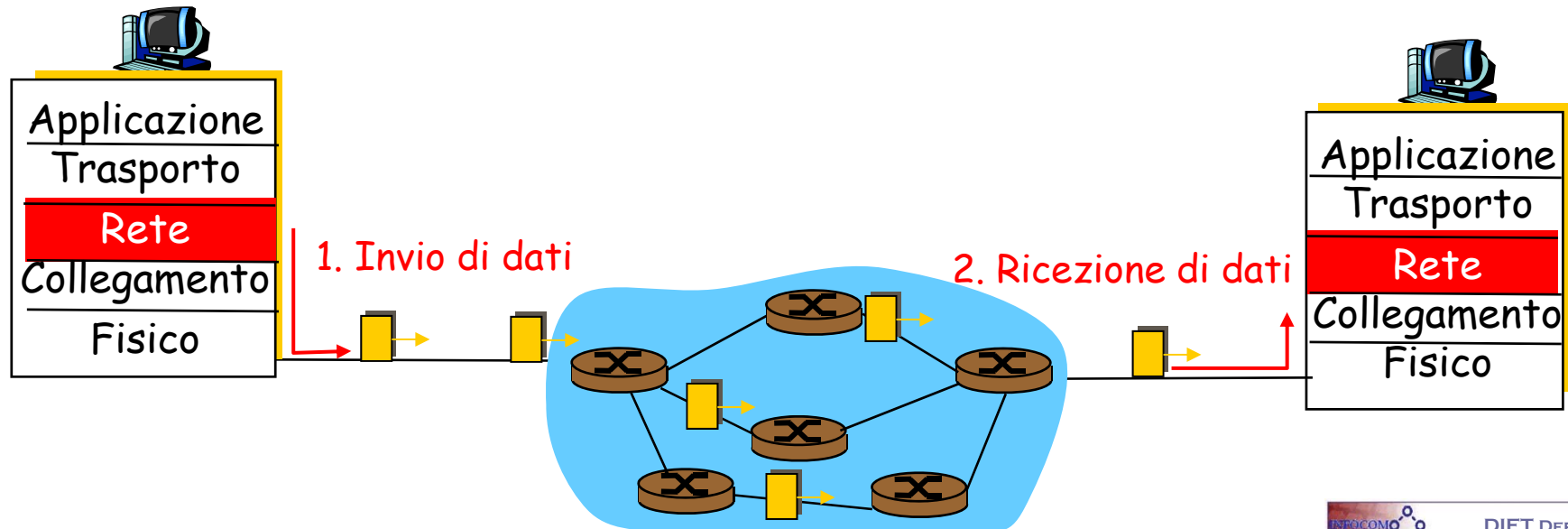
# Riassumendo...

- **Un circuito virtuale consiste in**
  - un percorso tra origine e destinazione
  - identificatori di connessione (VCI) (label), uno per ciascun link
  - righe nella tabella di forwarding in ciascun nodo (switch)
- **Il pacchetto di un circuito virtuale trasporta il VCI (label) nella propria intestazione**
- **Il VCI (label) del pacchetto cambia su tutti i collegamenti lungo un percorso**
  - Un nuovo VCI (label) viene rilevato dalla tabella di forwarding



# Reti a datagramma

- I router della rete sono “**stateless**”
  - Non esiste il concetto di “connessione” a livello di rete
- I router utilizzano gli **indirizzi di destinazione** per effettuare il forwarding
  - I pacchetti possono seguire percorsi diversi in rete
  - La consegna in sequenza non è garantita





# Esempio di Tabella di routing

Intervallo degli indirizzi di destinazione		Interfaccia
da 11001000 00010111 00010000 00000000	➡	0
a 11001000 00010111 00010111 11111111		
da 11001000 00010111 00011000 00000000	➡	1
a 11001000 00010111 00011000 11111111		
da 11001000 00010111 00011001 00000000	➡	2
a 11001000 00010111 00011111 11111111		
altrimenti	➡	3

$2^{32}$  = circa 4 miliardi di possibili indirizzi



# Concetto di prefisso

Prefisso	Interfaccia
11001000 00010111 00010xxx xxxxxxxx →	0
11001000 00010111 00011000 xxxxxxxx →	1
11001000 00010111 00011xxx xxxxxxxx →	2
altrimenti →	3

■ Il pacchetto è instradato sulla porta di uscita corrispondente al prefisso di lunghezza maggiore contenuto nell'indirizzo di destinazione (**longest prefix matching**)

■ E' necessaria una fase di ricerca del prefisso

- Algoritmi di lookup
- Memorie particolari (TCAM)

## Esempi

DA 1: 11001000 00010111 00010110 10100001

DA 2: 11001000 00010111 00011000 10101010

Qual è interfaccia di uscita su cui saranno rilanciati questi pacchetti ?





# Content Addressable Memory (1)

- Una CAM è una memoria specializzata per eseguire operazioni di matching in modo parallelo
  - Es. Longest Prefix Matching
- Rispetto ad una memoria RAM, una CAM, oltre alle celle di memoria, contiene i circuiti di confronto per rilevare un match tra i bit memorizzati e i bit di input
- Elevate prestazioni: oltre  $100 \times 10^6$  lookup/sec
- Alto consumo: circa 10-15 W per chip



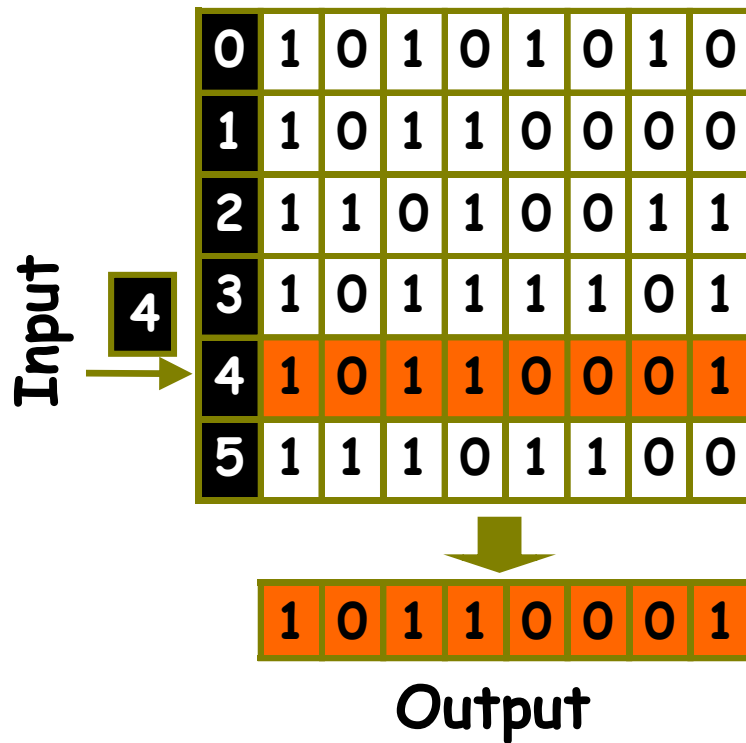
# Content Addressable Memory (2)

- Le **CAM** sono memorie speciali ottimizzate per le operazioni di confronto
- Operazione di lettura in una memoria tradizionale (RAM)
  - Input: indirizzo di una locazione di memoria
  - Output: contenuto della locazione di memoria
- Nelle **CAM** l'operazione di lettura è inversa
  - Input: parola da confrontare
  - Output: indirizzo della locazione che contiene la parola

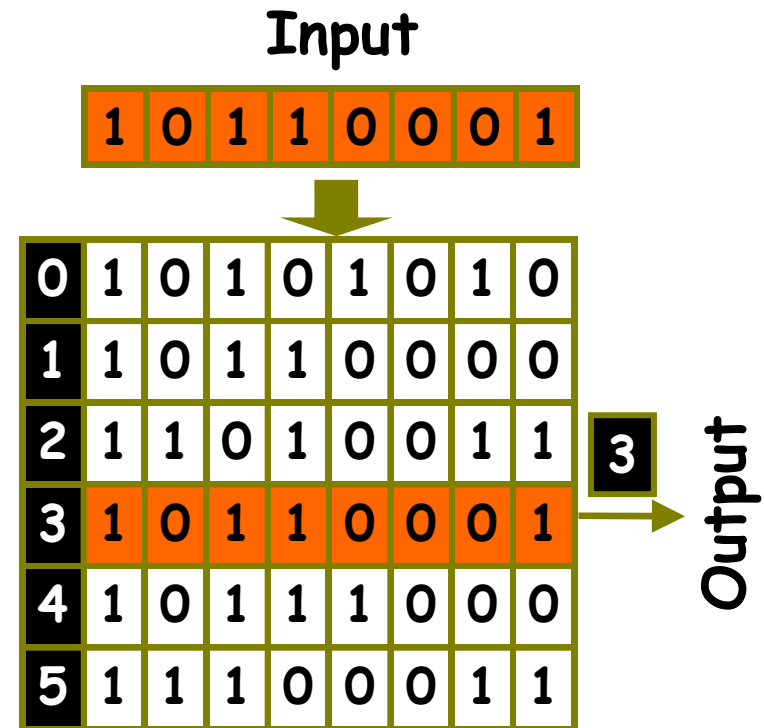


# RAM vs CAM

## RAM



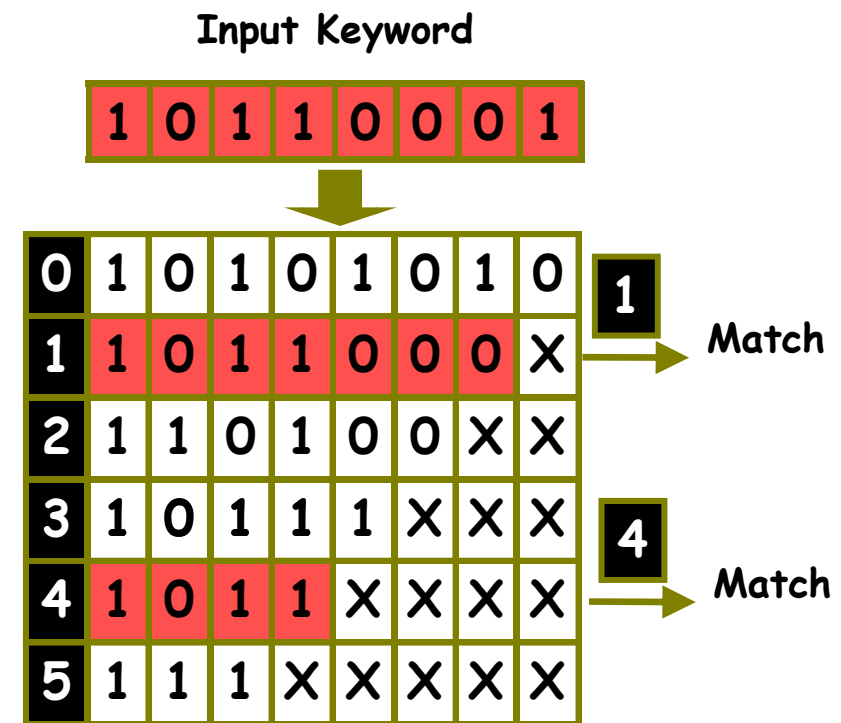
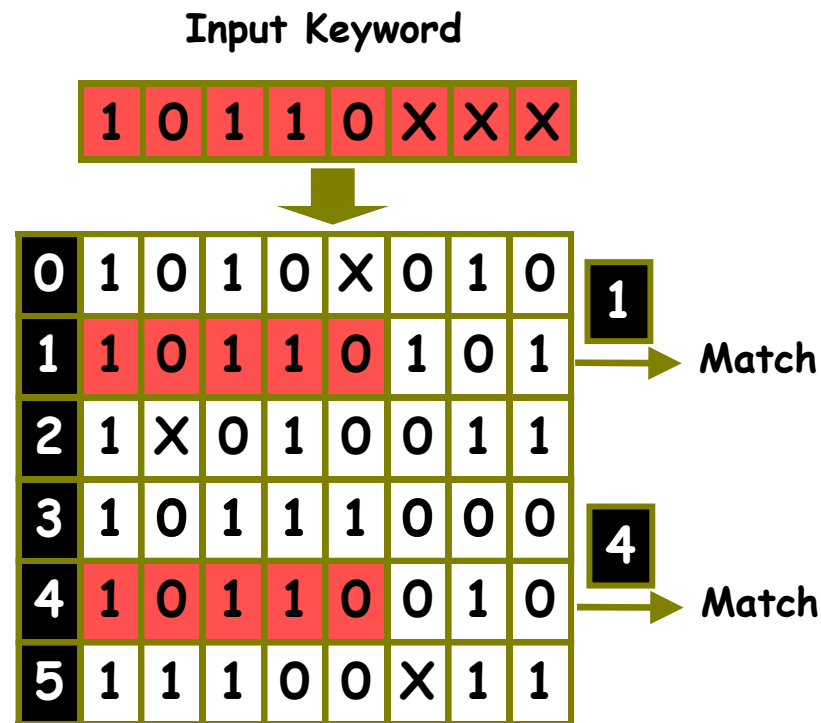
## CAM





# Ternary Content Addressable Memory

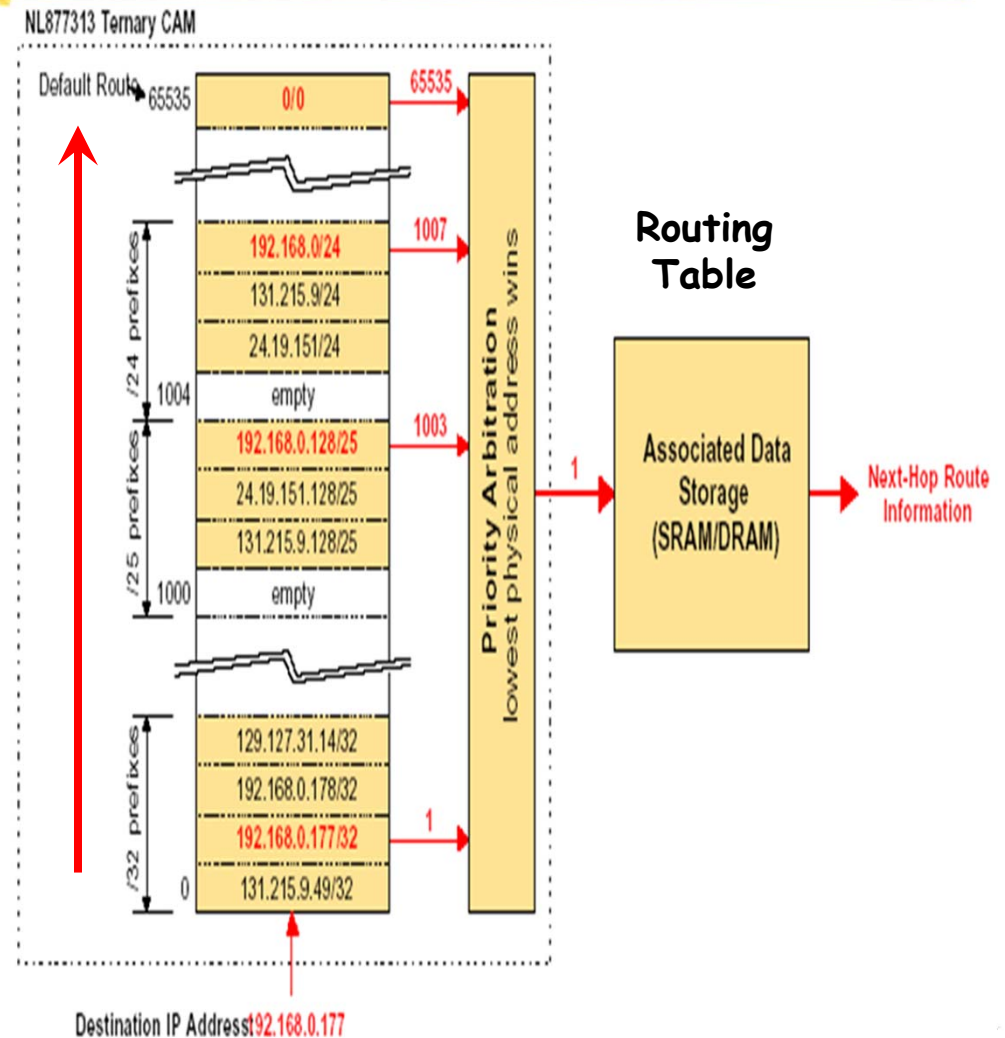
- Rispetto ad un CAM, una TCAM consente anche il confronto con presenza di "don't care" (X)





# Longest Prefix Matching con TCAM

- I prefissi sono ordinati in ordine di lunghezza crescente
- Un Destination Address è confrontato con i prefissi, in ordine crescente di lunghezza
- Il primo match indica l'indirizzo del record della Routing Table che contiene le informazioni di instradamento

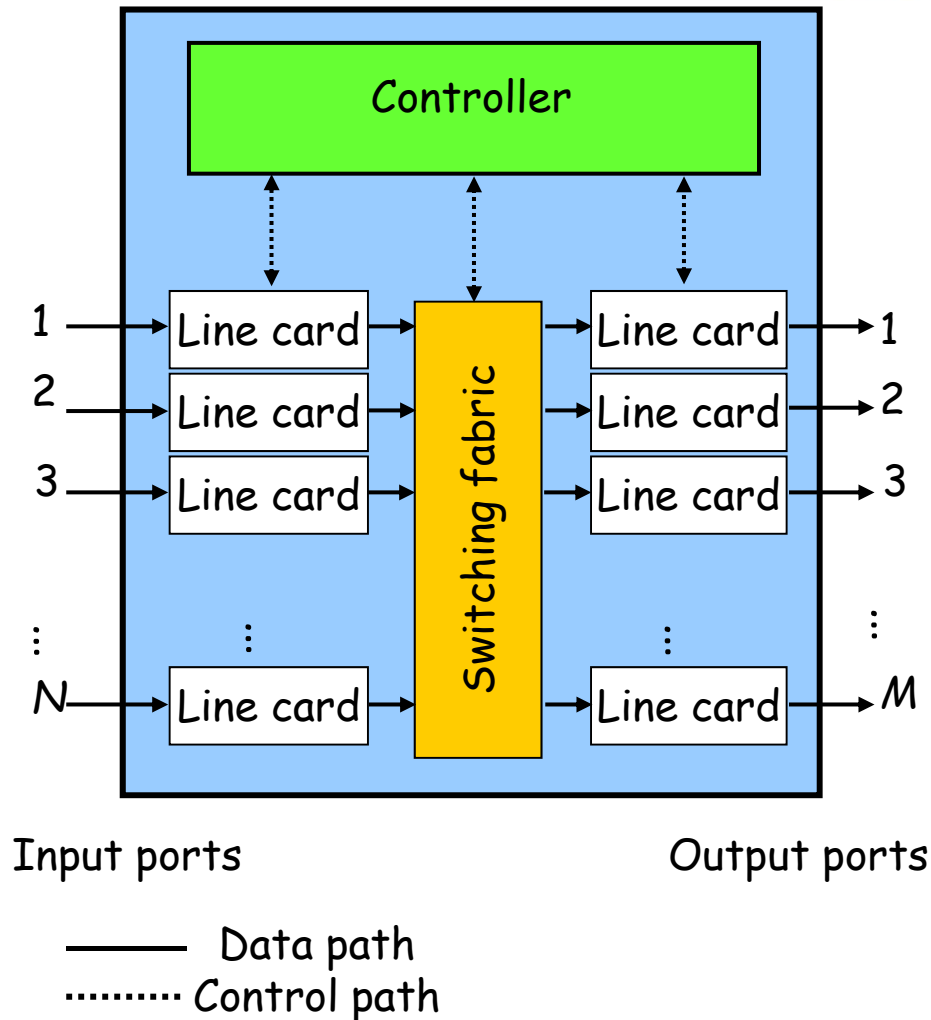




# Architetture di router



# Architettura di un router (1)



## Input Line Card

- Funzioni di strato 1 & 2
- Header processing
- Routing

## Controller

- Funzioni di controllo e di resource allocation

## Switching Fabric

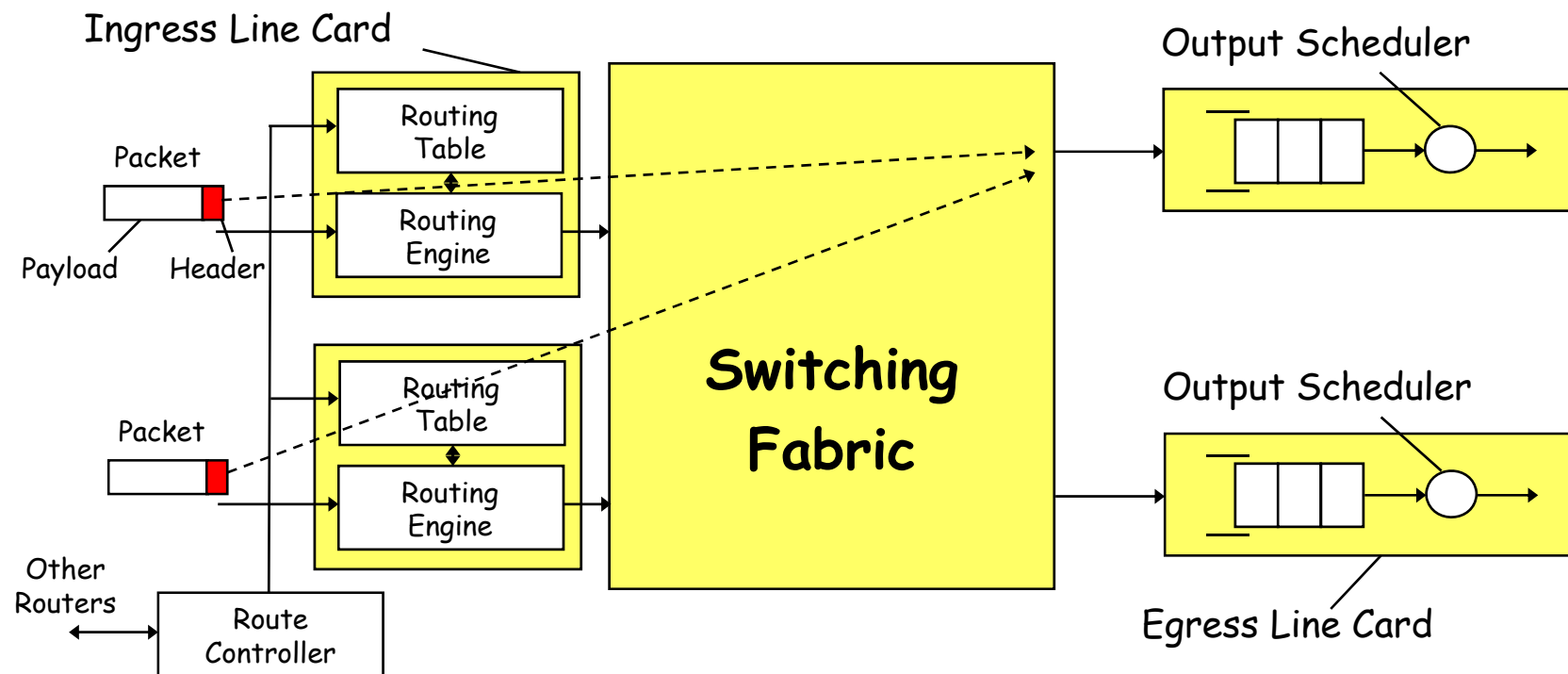
- Funzione di forwarding tra porte di ingresso e di uscita

## Output Line Card

- Scheduling & priority
- Multiplexing



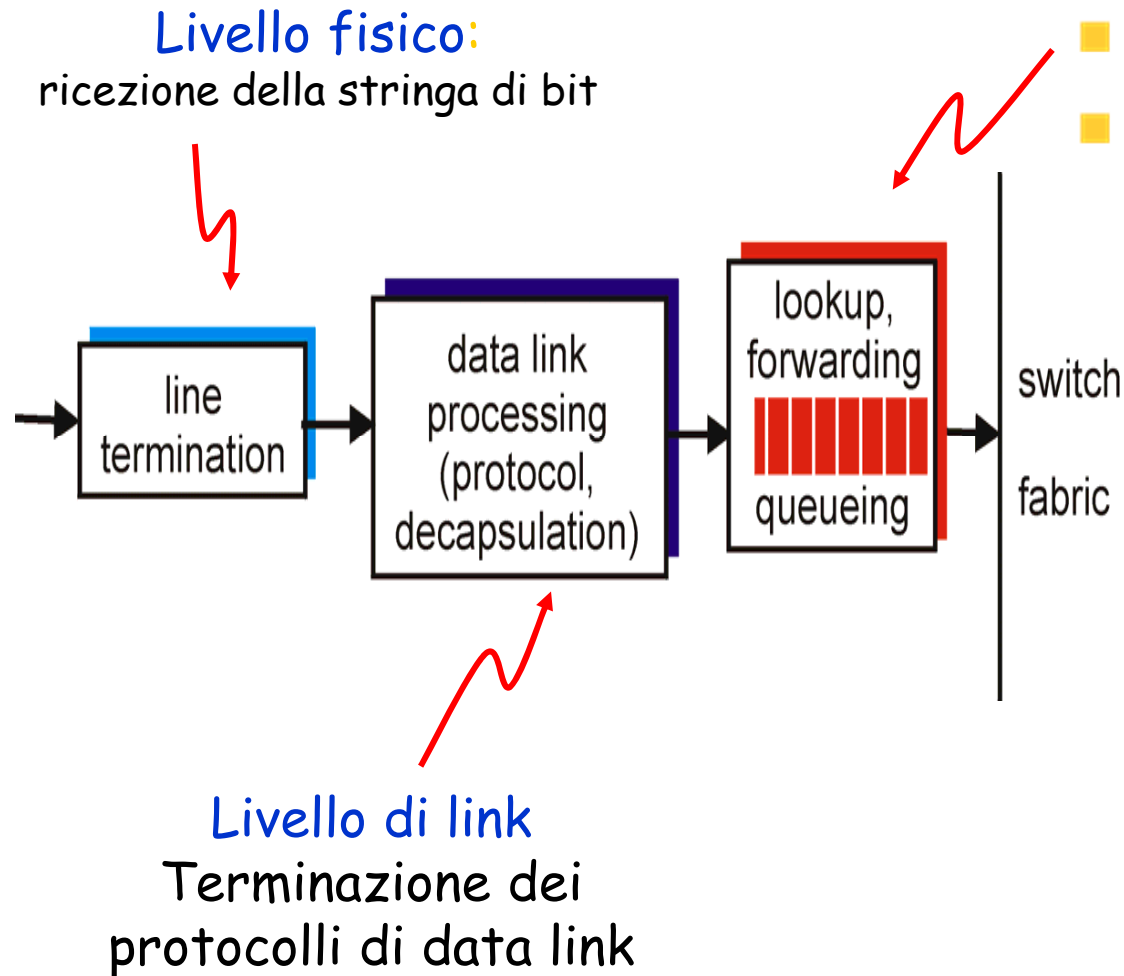
# Architettura di un router (2)







# Porte d'ingresso (Input Line Card)



## Switching distribuito

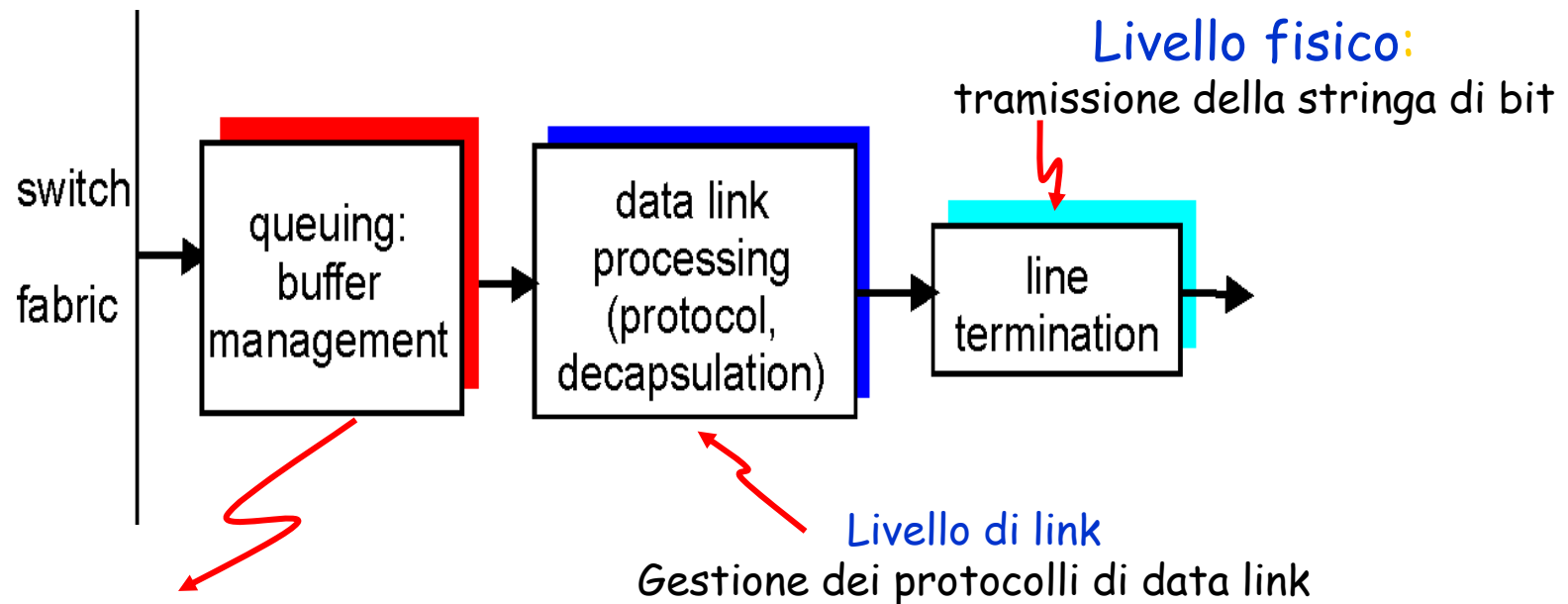
Determina la porta d'uscita dei pacchetti utilizzando le informazioni della tabella di routing

- Obiettivo: completare l'elaborazione allo stesso tasso della linea

Funzione di bufferizzazione se il tasso di arrivo dei pacchetti è superiore a quello di inoltro



# Porte d'uscita (Output Line Card)



## ■ Bufferizzazione

- Utilizzato se il tasso di arrivo dei pacchetti dalle porte di ingresso è superiore al tasso massimo di forwarding sul collegamento uscente

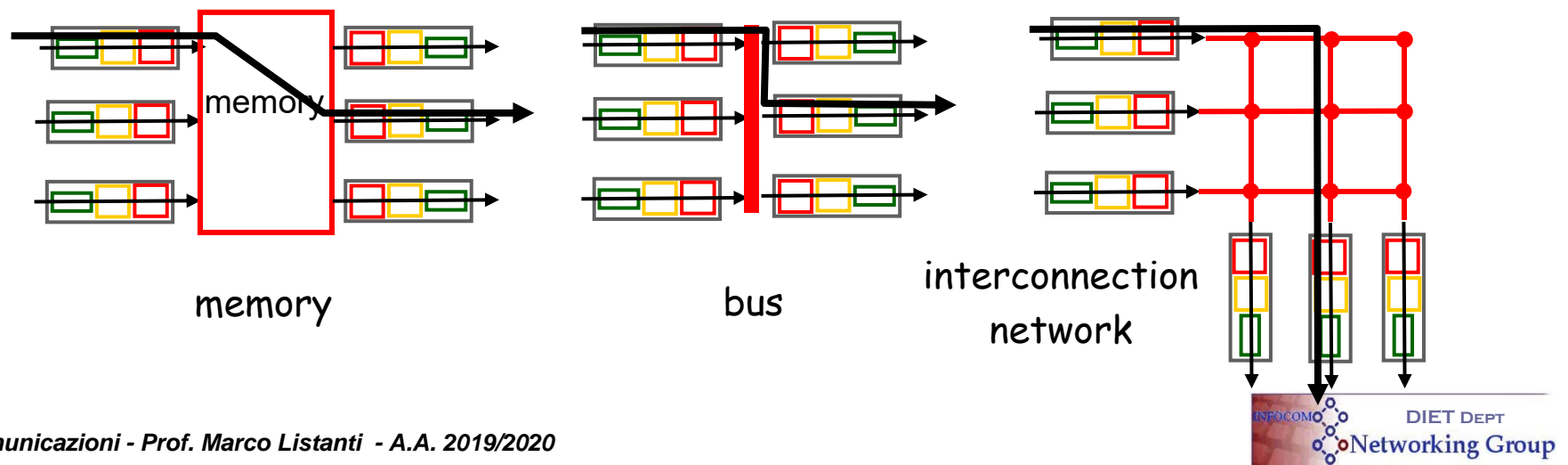
## ■ Packet scheduling

- stabilisce in quale ordine trasmettere i pacchetti accodati



# Switching fabric

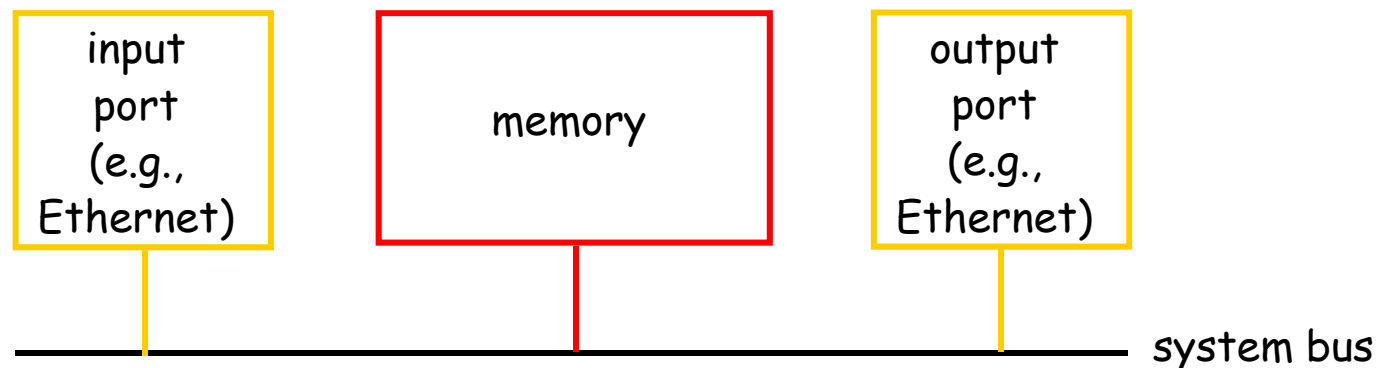
- Esegue il trasferimento dei pacchetti dalla porta di ingresso alla porta di uscita
  - **Switching rate**: rate massimo al quale i pacchetti possono essere trasferiti nella switching fabric
  - **Obiettivo**: N porte di ingress: switching rate N volte il rate di linea
- Tre tipi di switching fabric:





# Architettura memory-based

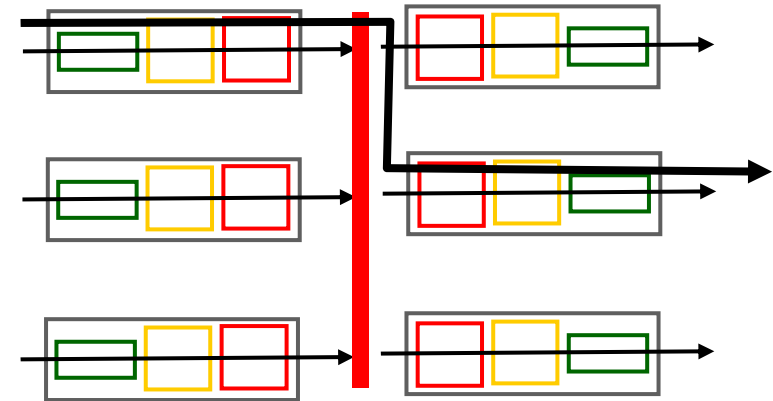
- Prima generazione di router
- I pacchetti sono copiati nella memoria centrale
  - Operazioni di scrittura e di lettura
- Switching rate limitato dalla velocità della memoria
- Ogni pacchetto attraversa due volte il bus





# Architetture a bus

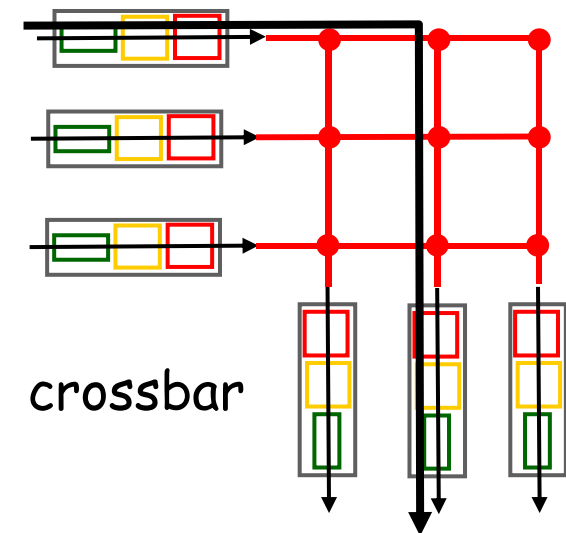
- Le card di input e di uscita sono interconnesse da un bus
- I pacchetti sono trasferiti attraverso il bus
- Switching rate limitato dalla data rate del bus
- Gestione delle contese tra pacchetti per l'uso del bus (**contention**)
- Adatta per router di piccole dimensioni





# Architetture con interconnection network

- Supera i limiti delle architetture a bus
- Architetture derivate dalle reti multiprocessore
  - Es: banyan networks, crossbar, ...
- Le unità dati trasferite all'interno della rete di interconnessione sono celle a lunghezza fissa
  - Necessità di segmentazione e ricostruzione dei pacchetti IP
- Lo switching rate cresce con le dimensioni della rete
- Adatta per router di grandi dimensioni





# Architetture con interconnection network

## Cisco CRS-1 System Configurations

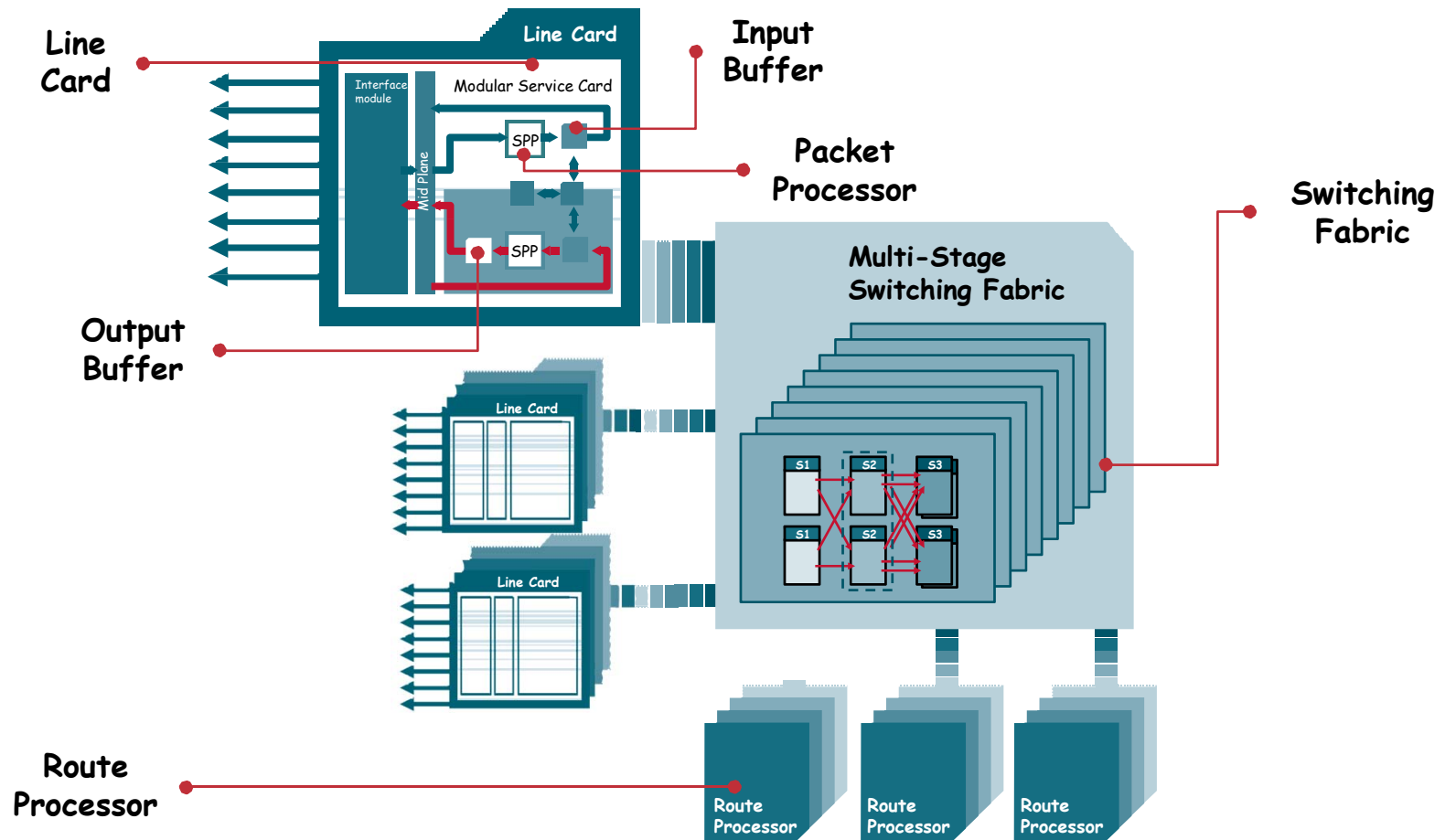
### Single-Shelf System Configuration

- Single 4-, 8-, or 16-slot line card shelf
- Integrated switch-fabric card—no fabric card shelf required
- Switching capacity: 320 Gbps, 640 Gbps, or 1.2 Tbps
- Supports 4, 8, or 16 40-Gbps line cards
  - 4, 8, or 16 OC-768c/STM-256 PoS ports
  - 16, 32, or 64 OC-192c/STM-64c PoS/Dynamic Packet Transport (DPT) ports
  - 32, 64, or 128 10 Gigabit Ethernet ports
  - 64, 128, or 256 OC-48c/STM-16c PoS/DPT ports
  - 4, 8, or 16 OC-768c/STM-256 tunable WDMPOS ports
  - 16, 32, or 64 10 Gigabit Ethernet tunable WDMPHY ports





# Architetture con interconnection network

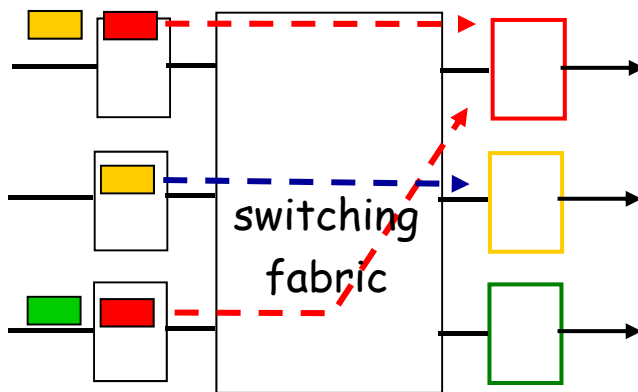






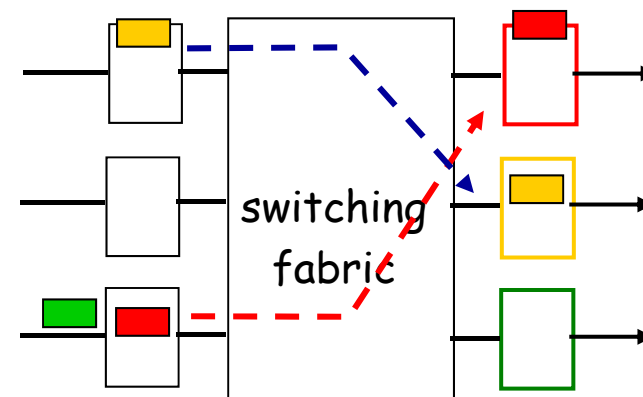
# Input port queuing

- La memorizzazione dei pacchetti in caso di congestion avviene in buffer posti nelle input card
  - Possibilità di ritardi e perdita di pacchetti
- **Problema** : **Head-of-the-Line (HOL) blocking**
  - Il primo pacchetto memorizzato nella coda in input può bloccare i pacchetti successivi



Output port contention

Solo uno dei due pacchetti rossi può essere trasferito, l'altro deve essere memorizzato

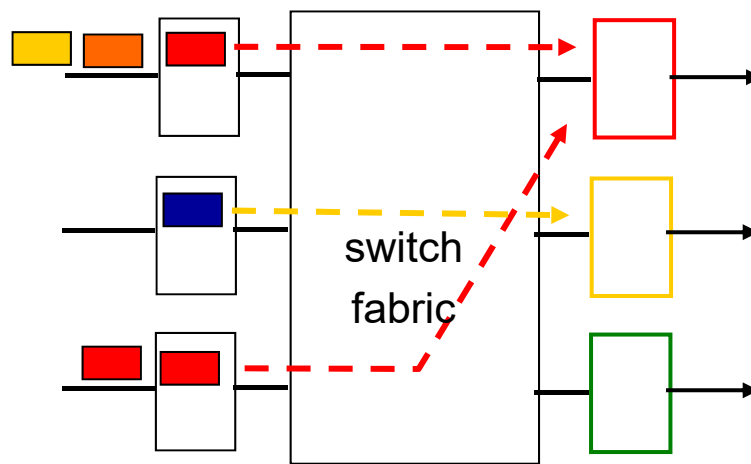


HOL blocking

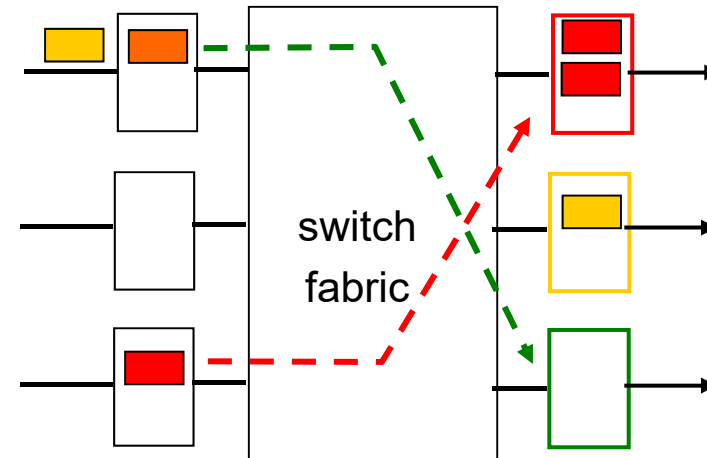
il pacchetto verde subisce un blocco anche se la porta di uscita è libera



# Output port queueing



Contention sulla linea di uscita



Memorizzazione di uno dei pacchetti rossi

- I buffer sono collocati nelle card di output
  - Si ha bufferizzazione quando il tasso di arrivo dei pacchetti verso una singola card supera il rate di trasmissione sul link di uscita
- Non esiste il problema dell' HOL



# Dimensionamento dei buffer

## ■ Regola empirica

- La dimensione dei buffer **B** dipende dal prodotto **RTT·C** (prodotto banda ritardo)

- Dove RTT è il Round Trip Time tipico del router e C è il bit rate del link di uscita
- Esempio: RTT = 250 msec; C = 10 Gpbs; B = 2.5 Gbit

- Se sono presenti N flussi si ha

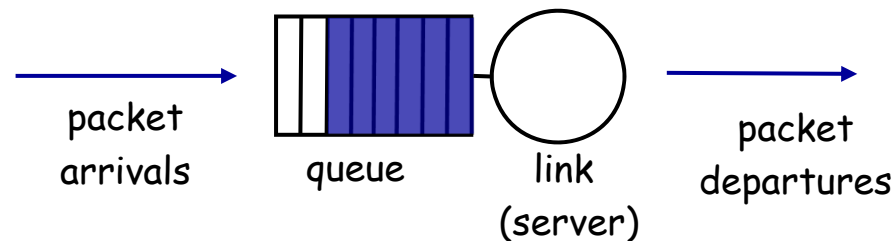
$$B = \frac{RTT \cdot C}{\sqrt{N}}$$

- Es. Se N = 10.000 flussi si ha B = 250 Mbit



# Politiche di scheduling: FIFO

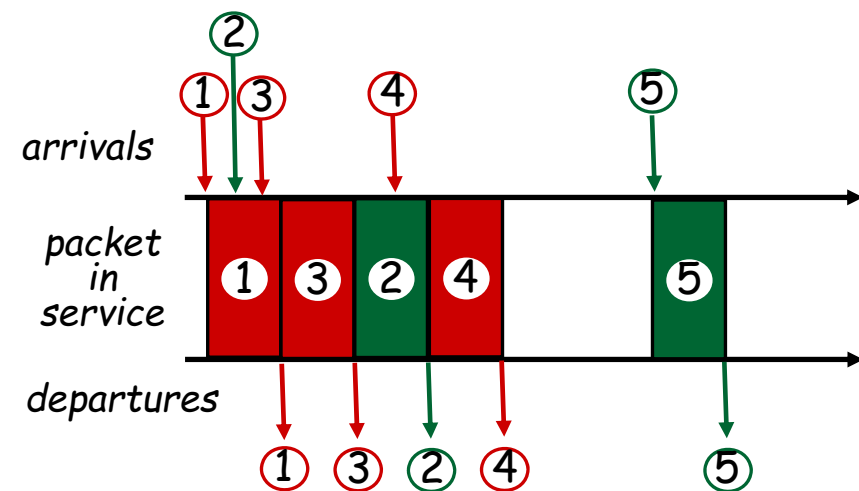
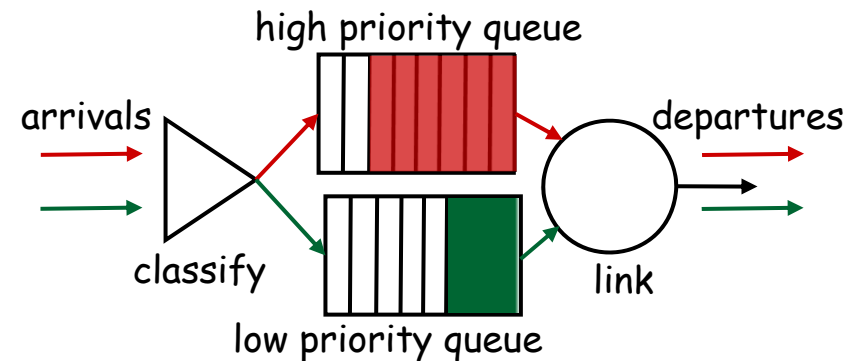
- La funzione di scheduling ha lo scopo di determinare il successivo pacchetto da trasmettere sulla linea di uscita
- **FIFO (first in first out)**
  - Scheduler tradizionale
  - Sceglie i pacchetti da trasmettere secondo l'ordine di arrivo al buffer
- **Politica di scarto dei pacchetti in caso di saturazione del buffer**
  - **tail drop**: si scarta l'ultimo pacchetto arrivato
  - **priority**: scarta i pacchetti secondo al la loro livello di priorità
  - **random**: si sceglie il pacchetto da scartare in modo casuale





# Politiche di scheduling: Priorità

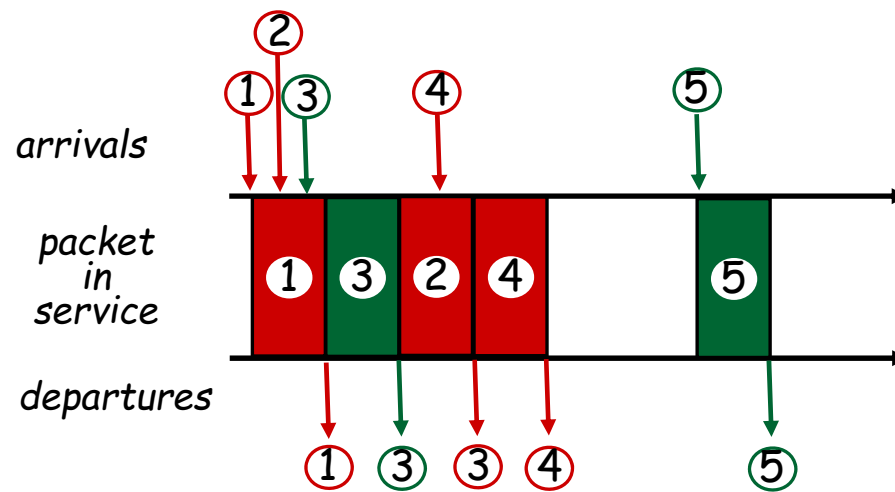
- Il pacchetto da trasmettere è quello a priorità massima
- Classi multiple con livelli di priorità diversi
  - La classe è identificata attraverso un funzione di classificazione
    - Es. IP source/dest, port numbers, etc.





# Round Robin (RR) scheduling

- **Gestione di classi multiple**
  - Ad ogni classe è associato un buffer logico diverso
- **Lo scheduler esamina ciclicamente le code ed emette, se esiste, il primo pacchetto di ciascuna coda**





# Weighted Fair Queuing (WFQ)

- Generalizza il funzionamento dello scheduler RR
- Ad ogni classe è associato un peso che corrisponde alla frequenza con cui viene esplorata la coda

