RETI DI CALCOLATORI

Autunno 2018

docente: Laura Ricci

laura.ricci@unipi.it

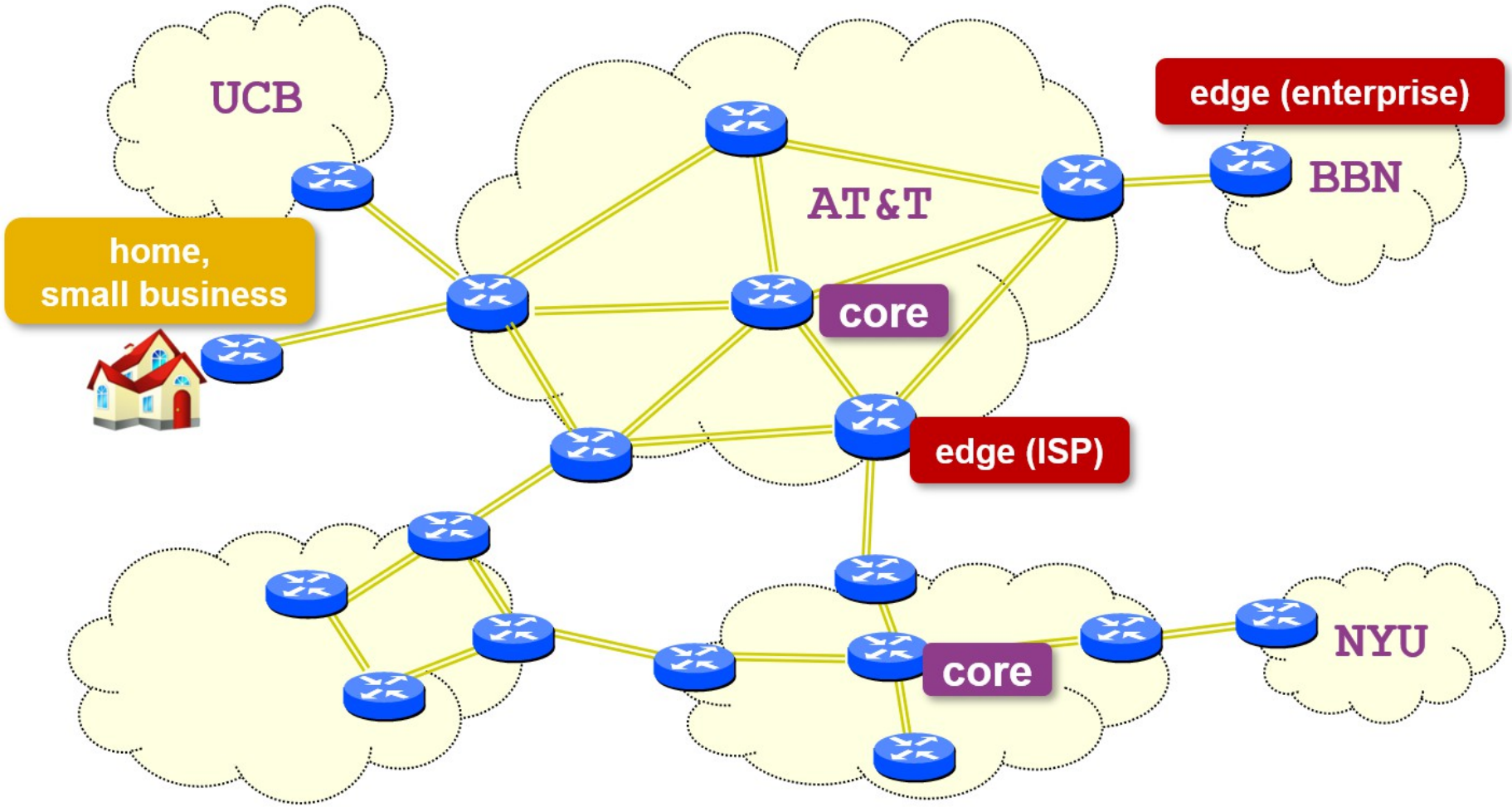
Lezione 8:
IL LIVELLO IP:
STRUTTURA DEI ROUTERS, ICMP
22/10/2018

parte di queste slides sono
ricavati da slides pubblicate in corsi
universitari tenuti da colleghi
italiani e stranieri

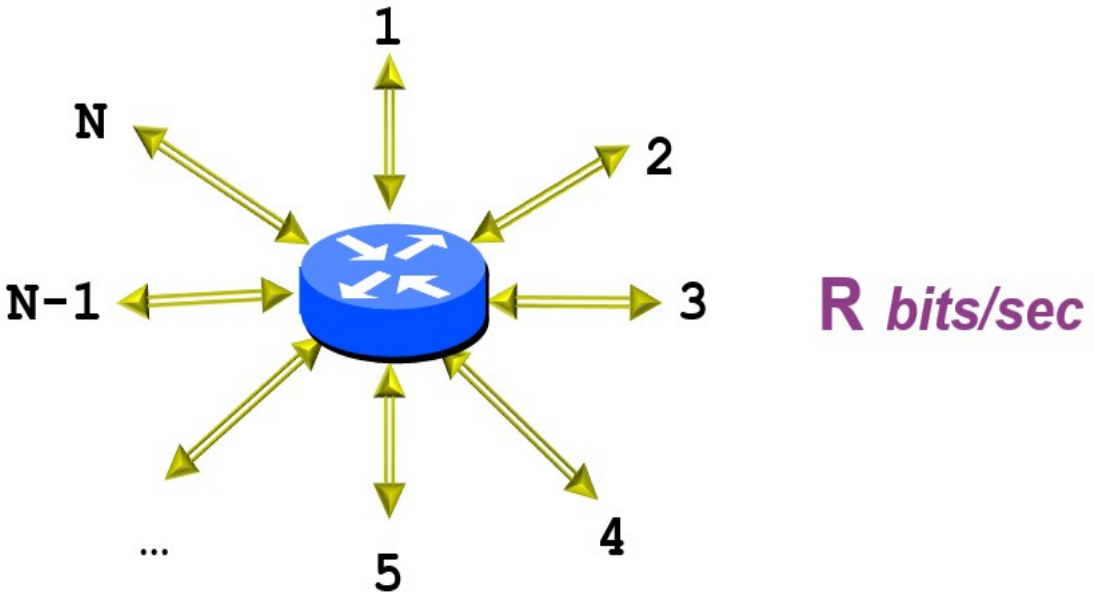
ROUTER: BUILDING BLOCK DI INTERNET

- Core building block dell'infrastruttura di Internet
- Volume di affari: $120B+
- Players: Cisco, Huawei, Juniper, Alcatel-Lucent (>90% del mercato)
  - Cisco
    - CRS (Carrier Router Series)
  - Juniper
    - T-series
  - Alcatel-Lucent (soon to be Nokia)
    - XRS (Extensible Routing System)
  - Huawei
    - Netengine
- Altri produttori per dispositivi per aggregazione/accesso da parte di edge nodes.

NETWORK E ROUTERS



ROUTER: CAPACITA'



- N = numbero di "porte" esterne del router
- R = veocità della singola porta ("line rate")
- capacità del router = N x R

ESEMPI DI ROUTERS: CORE

Juniper T4000

- R= 10/40 Gbps
- NR = 4 Tbps

Cisco CRS

- R=10/40/100 Gbps
- NR = 322 Tbps

72 racks, 1MW

ESEMPI DI ROUTERS: SMALL BUSNESS

Cisco 3945E

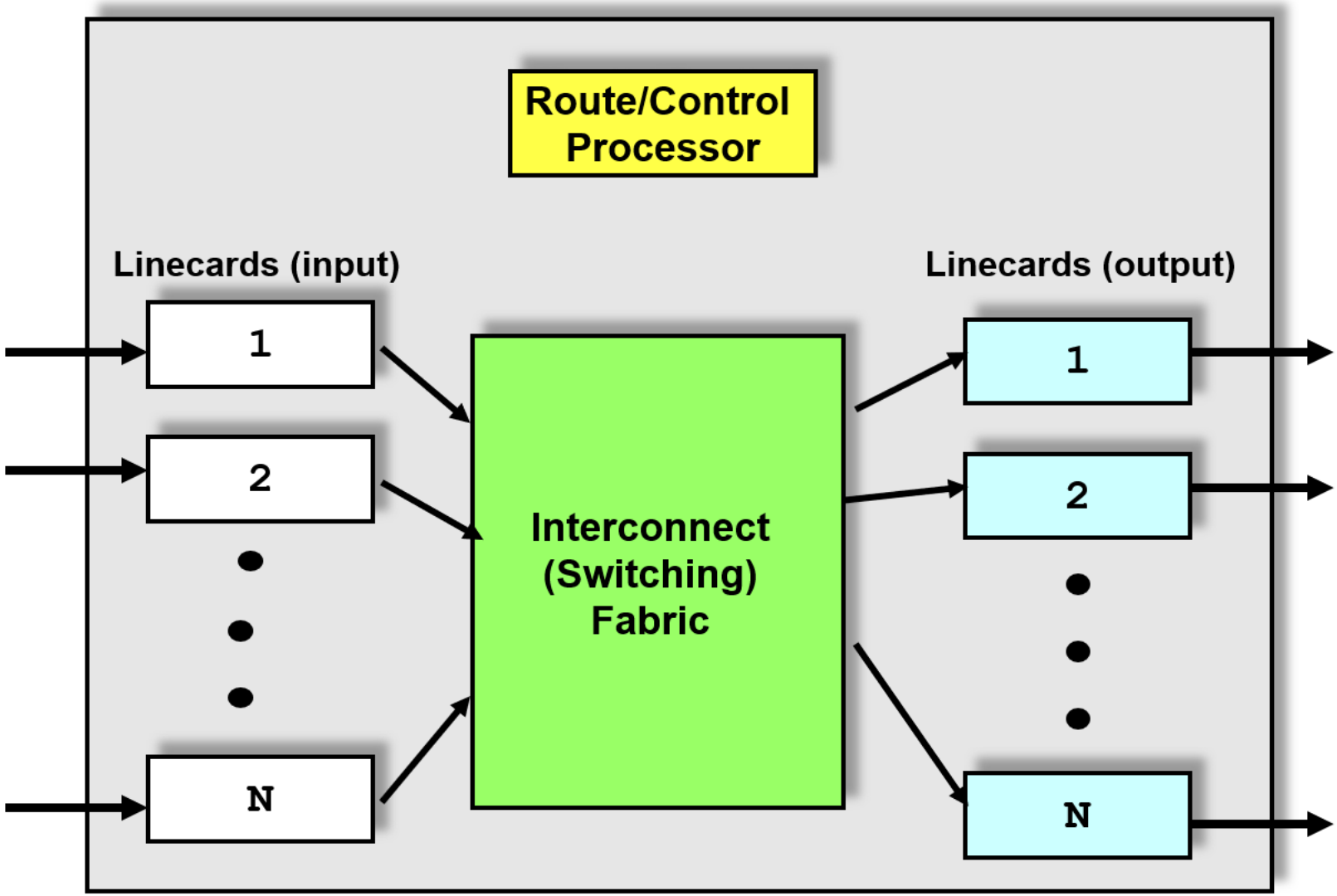
- R = 10/100/1000 Mbps
- NR < 10 Gbps

ROUTER: FUNZIONI PRINCIPALI

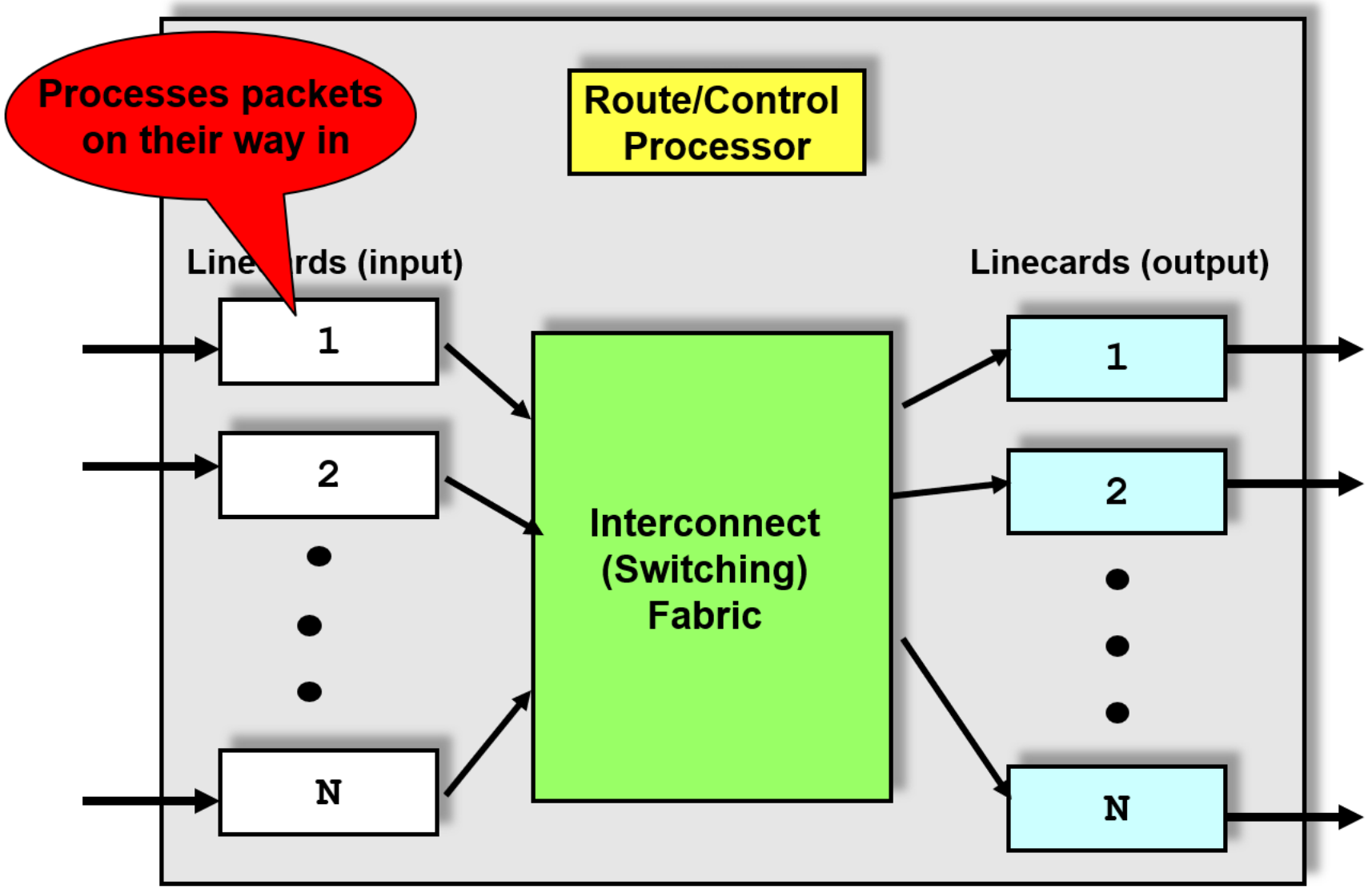
Due funzioni fondamentali:

- eseguire gli algoritmi/protocolli di routing (RIP, OSPF, BGP)
- inoltrare i datagram dai link di ingesso a quelli di uscita

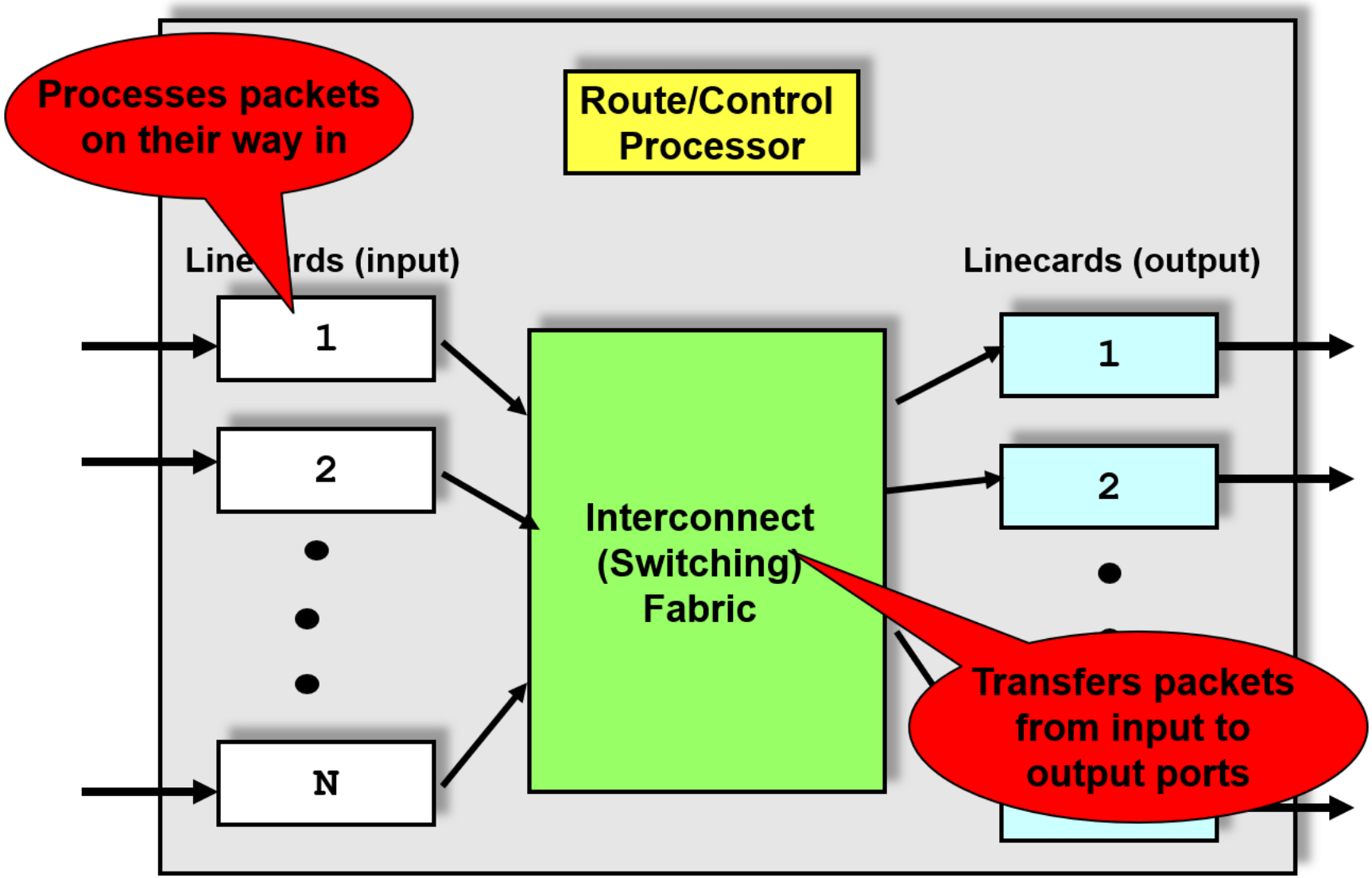
ARCHITETTURA DI UN ROUTER



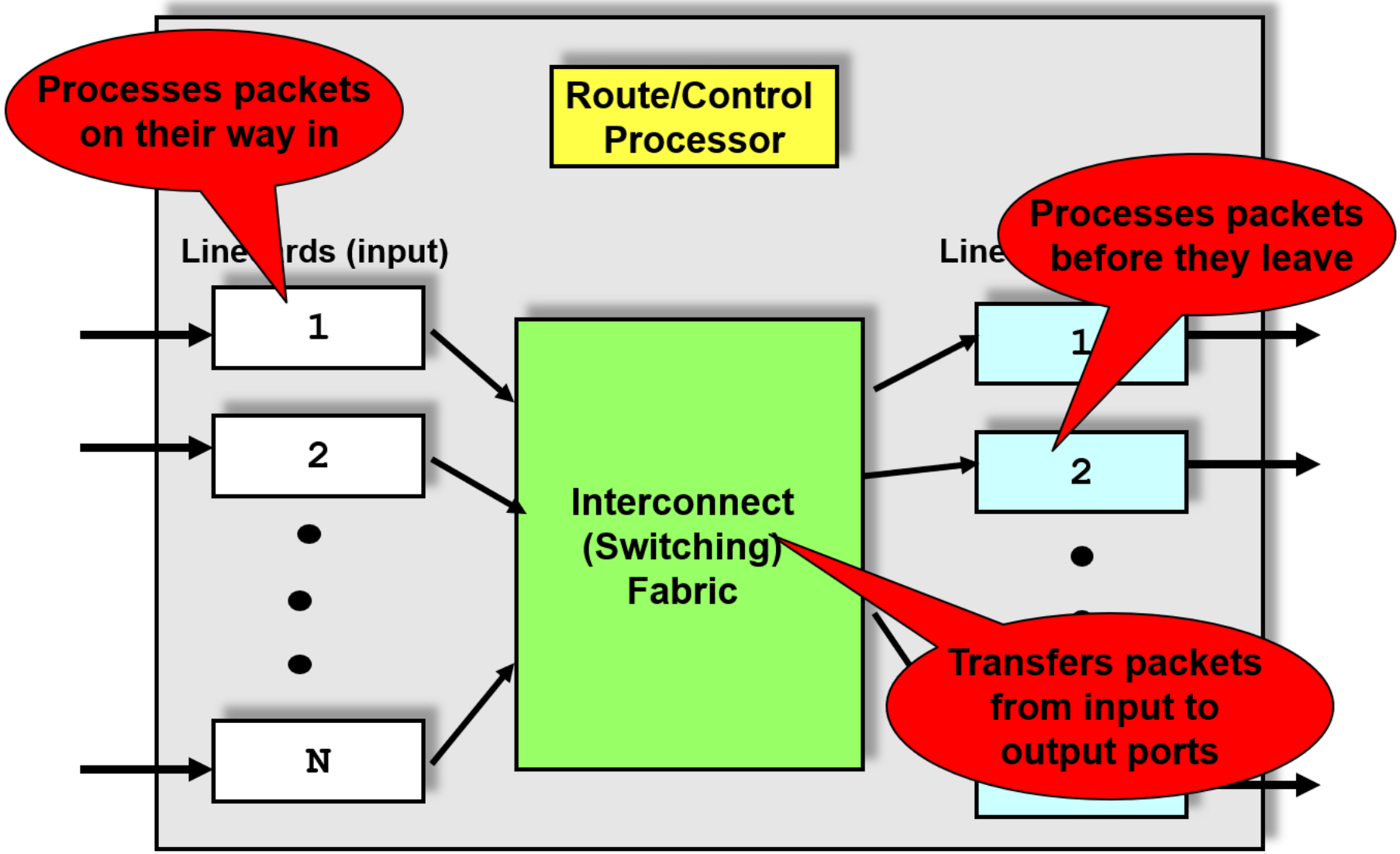
ARCHITETTURA DI UN ROUTER



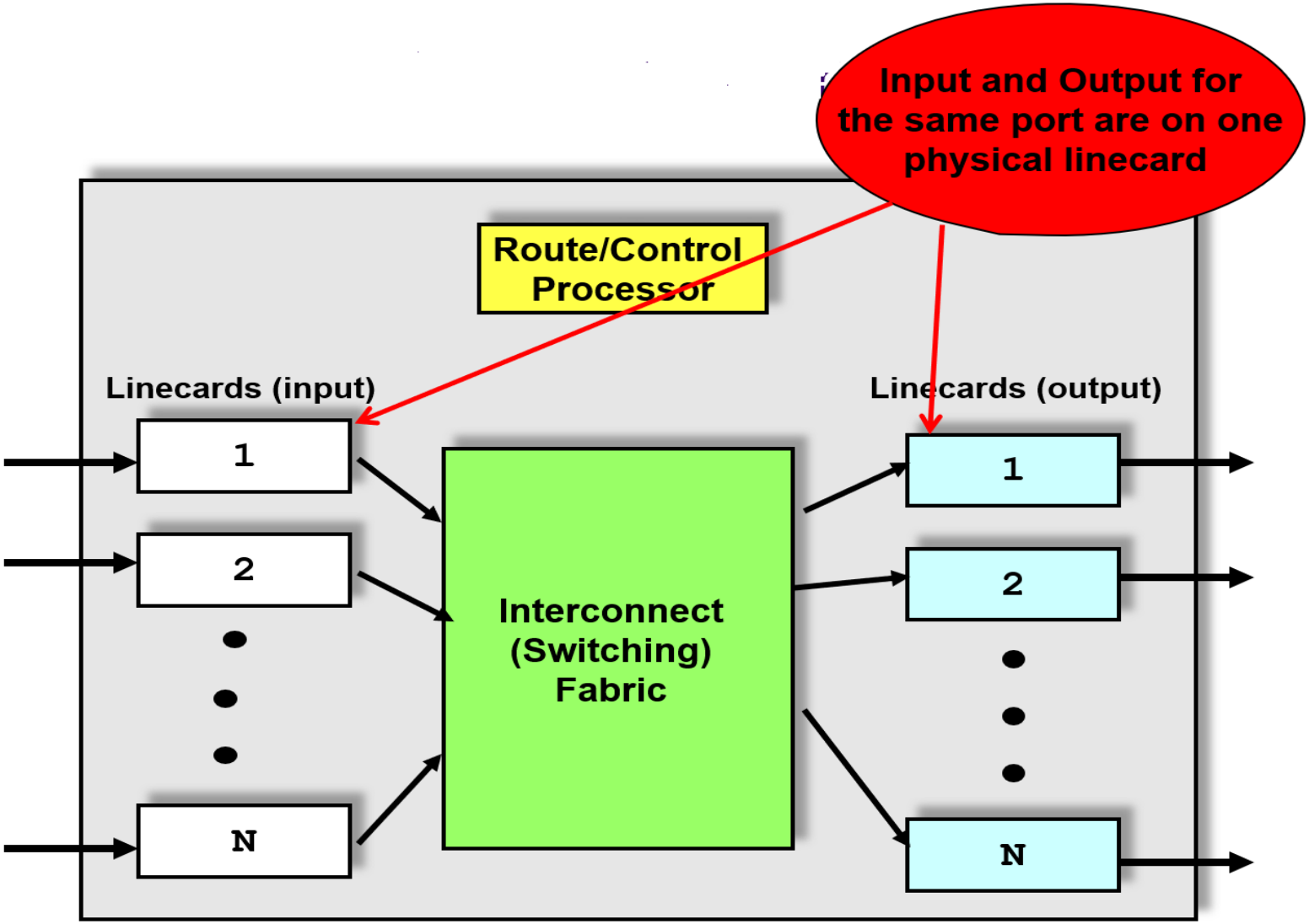
ARCHITETTURA DI UN ROUTER



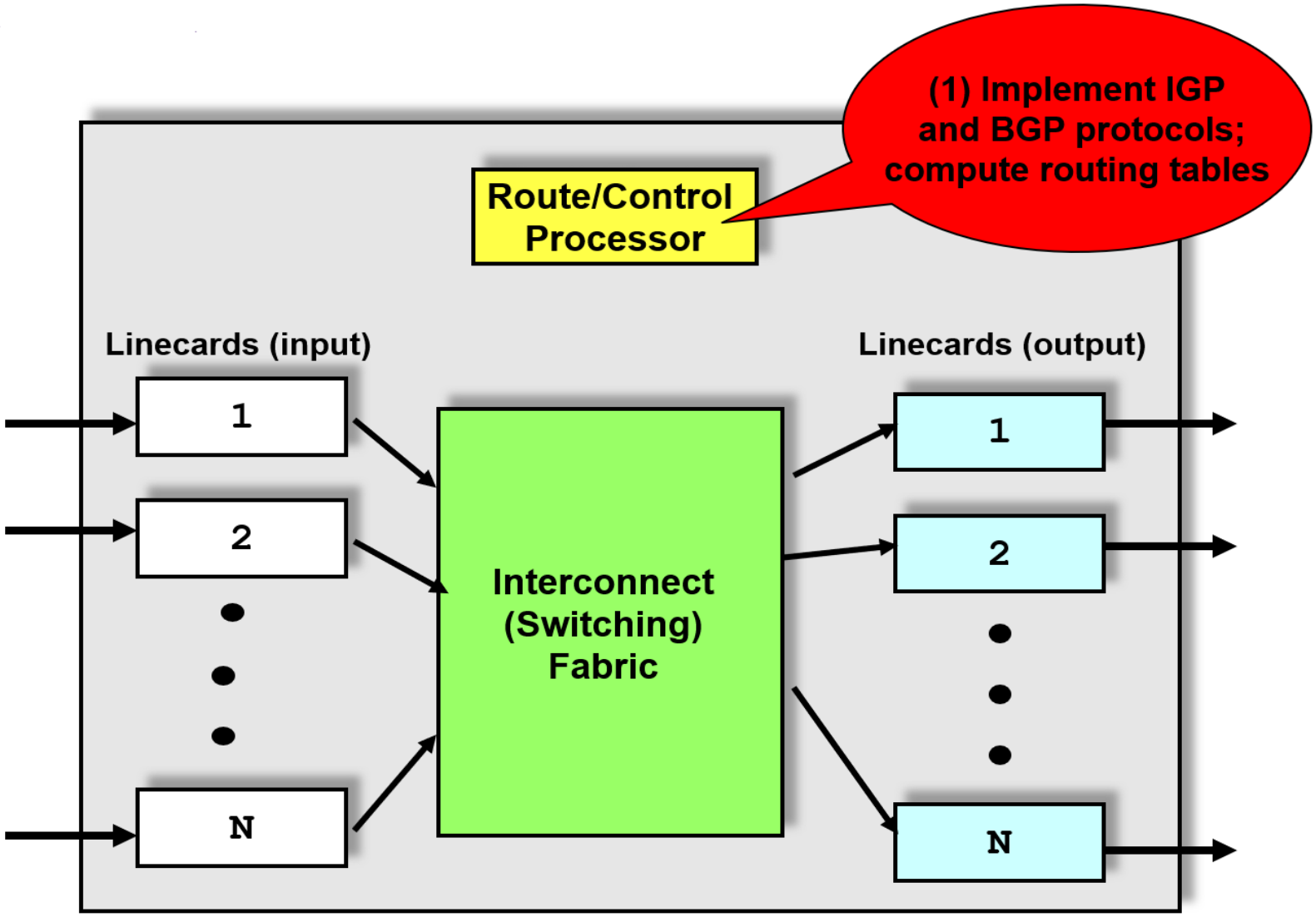
ARCHITETTURA DI UN ROUTER



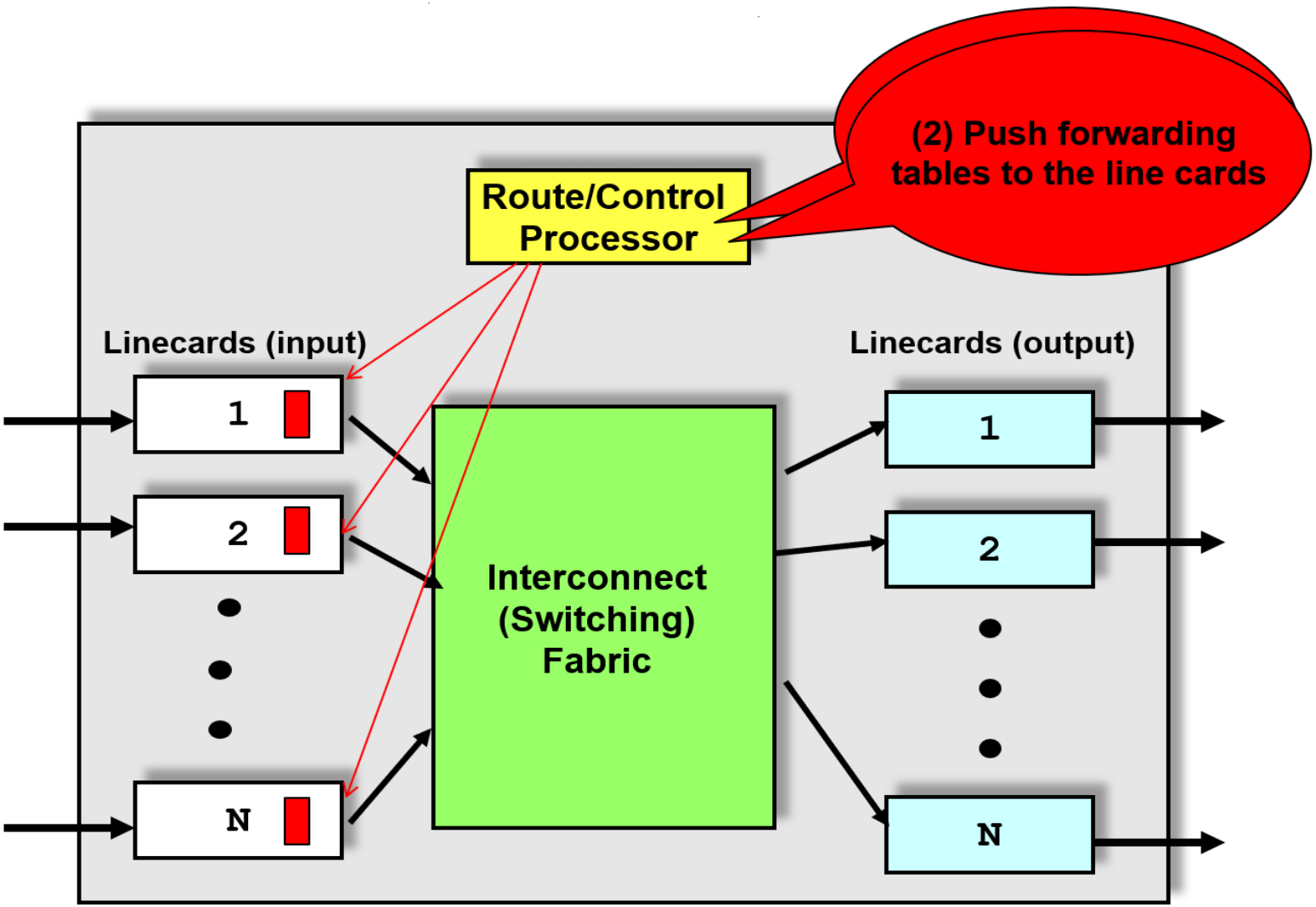
ARCHITETTURA DI UN ROUTER



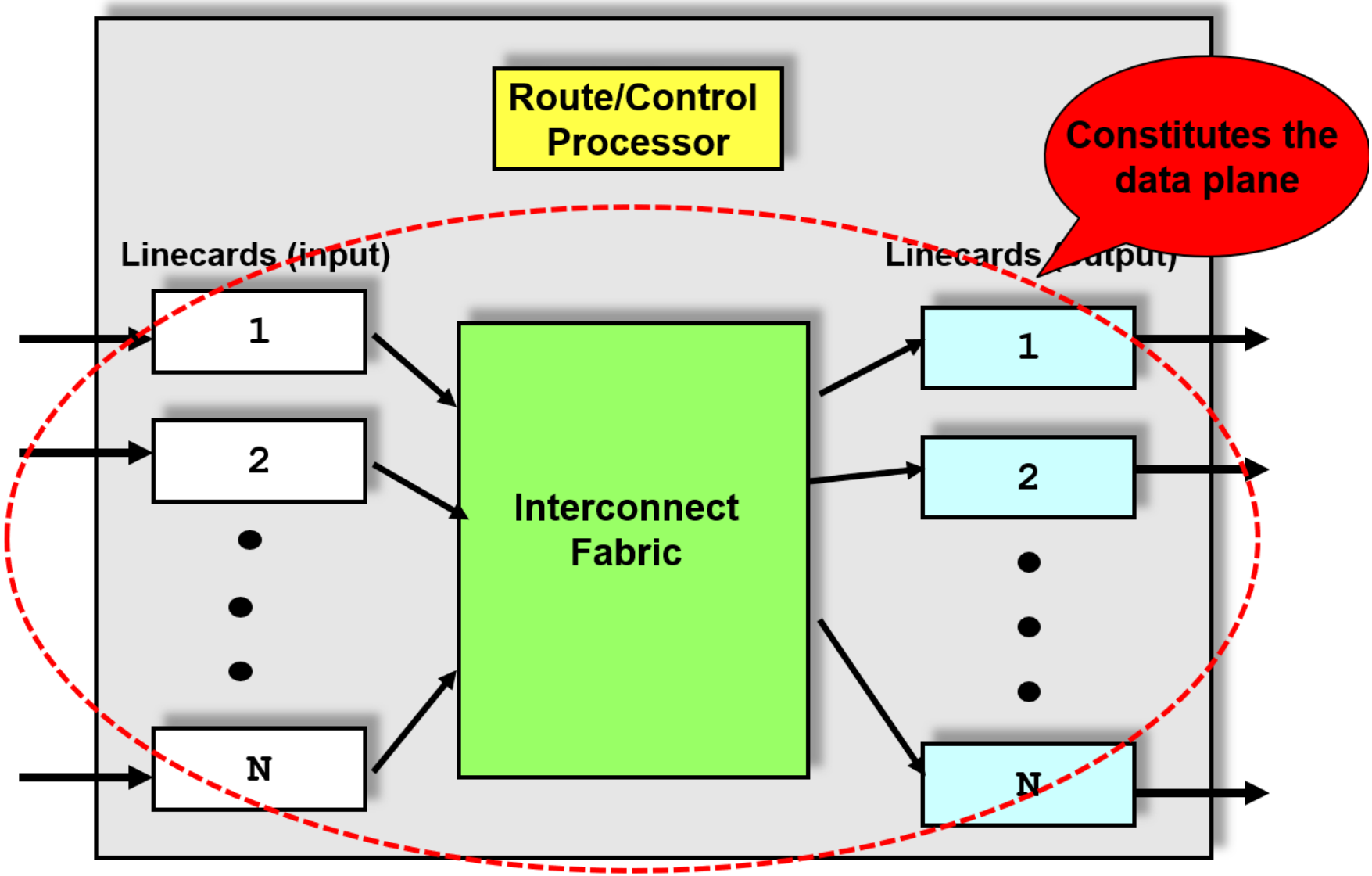
ARCHITETTURA DI UN ROUTER



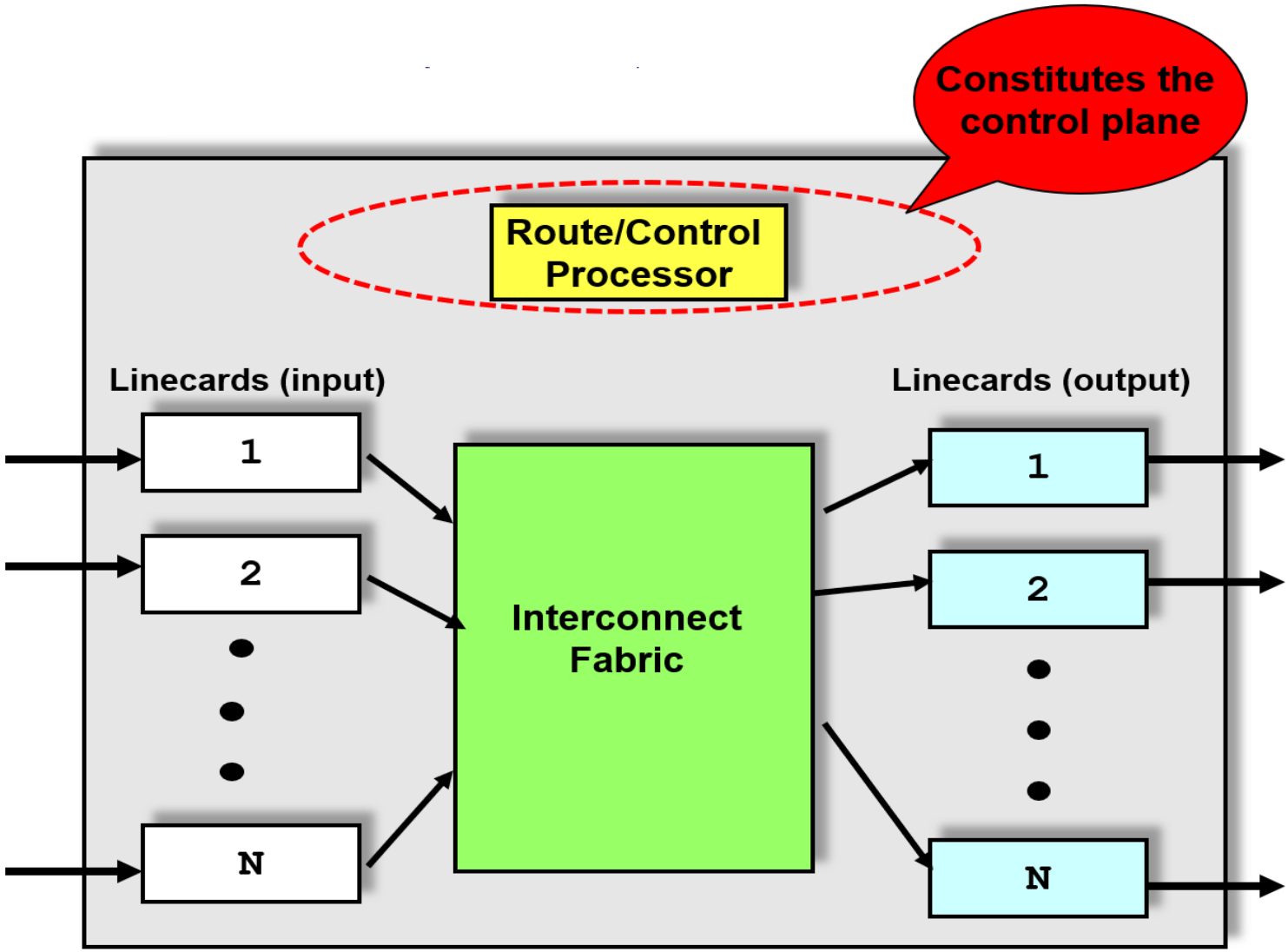
ARCHITETTURA DI UN ROUTER



ARCHITETTURA DI UN ROUTER



ARCHITETTURA DI UN ROUTER



IP FORWARDING

- il router riceve l'indirizzo IP dell'end-host destinazione
  - l'indirizzo non contiene la maschera
- il processo di routing ha già calcolato una forwarding table che contiene tutta l'informazione per il processo di forwarding
- problema: come può un router trovare l'indirizzo destinazione nella routing table nel modo più efficiente possibile?
  - ricerca nella forwarding table dell'indirizzo.
    - in corrispondenza dell'indirizzo, l'interfaccia di uscita del router su cui inoltrare il pacchetto
  - address look-up
    - uno dei principali bottleneck del processo di forwarding
    - come rendere efficiente questo processo?

FORWARDING TABLE (INTERCONNECT FABRIC)

- interfaccia tra control e data plane
  - concettualmente una map/dictionary/hashtable che mappa indirizzi IP destinazione sulla una porta di uscita del router che lo connette al nodo prossimo hop verso la destinazione
  - creata dal control plane, letta dal data plane
- un accesso per ogni pacchetto ricevuto
  - meccanismi di accesso devono essere molto efficienti
- high speed router 1Tbit/s (1 T = 10^{12})
  - con pacchetti di 1000 bit, un pacchetto ogni 1 ns
  - necessaria una forwarding table che richiede 1B lookups al secondo
  - non realizzabile con CPU off-the-shelf
  - richiede circuiti dedicati (switching chps), specializzati per l'istradamento.

IP FORWARDING: EXACT MATCHING

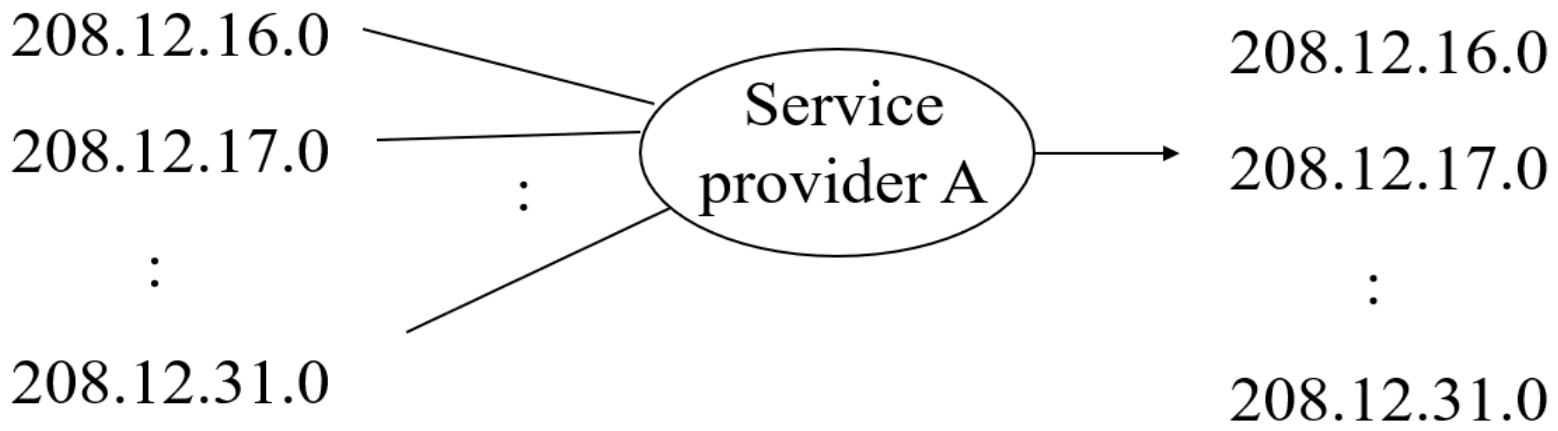
- tabella di forwarding contiene indirizzi IP completi, non maschere.
- l'indirizzo IP della destinazione viene confrontato con ogni indirizzo IP nella tabella
  - un unico match risultante, se la tabella di forwarding non contiene duplicati
  - non è una soluzione efficiente a livello IP
  - soluzione possibile a livello di switch, a livello data link
- indirizzi a 32 bit (IPv4): 2^{32} entrate nella tabella di routing, circa 4 miliardi di entrate, diversi Gbytes di memoria.
  - realizzabile con memorie off-the-shelf
    - esempio DRAM presenti nelle memorie dei processori possono supportare 1 milione di lookup al secondo
  - tuttavia queste memorie risultano troppo lente per supportare 1 miliardo di look-ups al secondo
  - inoltre: ogni router deve conoscere ogni altro router/end host
    - non compatibile con strutturazione gerarchica degli indirizzi

IP FORWARDING: PREFIX MATCHING

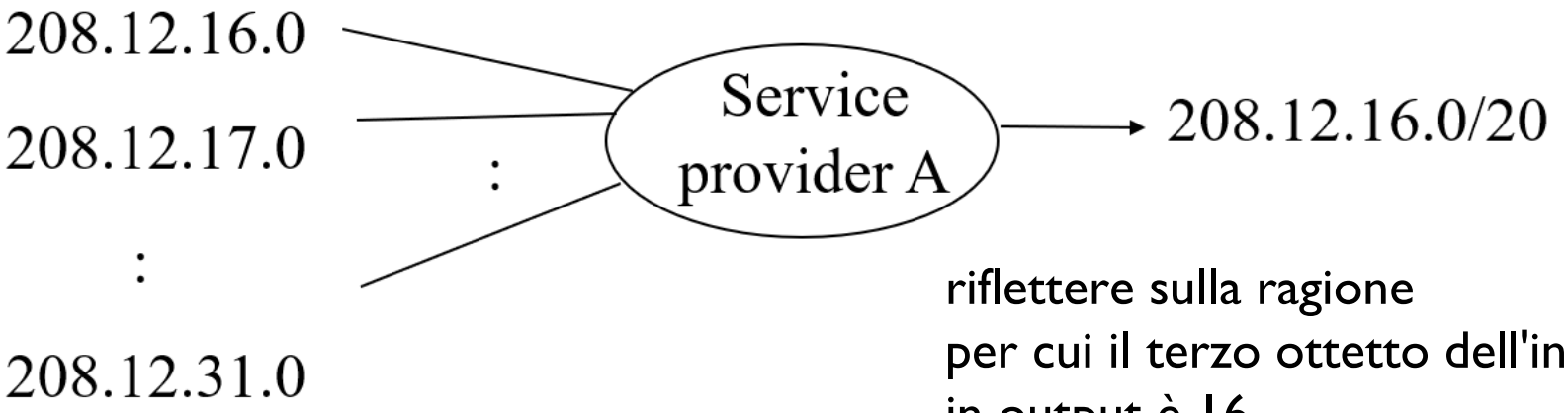
- Prefix matching
  - processo con sui si confronta un indirizzo IP con un insieme di indirizzi di rete
  - scopo: controllare se l'indirizzo IP condivide un prefisso comune con alcune reti
- Introduce la possibilità di avere più matching per ogni indirizzo IP
  - l'indirizzo IP 8.8.8.4 condivide un prefisso sia con la rete 8.8.*.* che con la rete 8.8.8.*
  - notare che gli indirizzi denotano due reti diverse
- l' Internet Protocol risolve questa situazione definendo il look up in modo che restituisca il prefisso più lungo (longest prefix match)
  - più probabile che sia vicino alla destinazione

PERCHE' LONGEST PREFIX MATCH?

- Inter-domain routing senza CIDR



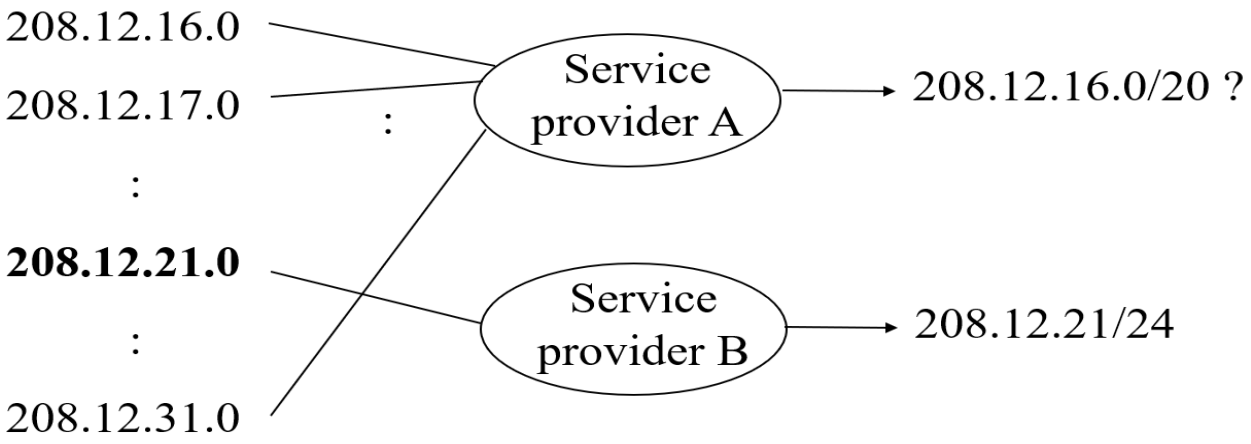
- Inter-domain routing con CIDR



riflettere sulla ragione
per cui il terzo ottetto dell'indirizzo
in output è 16

PERCHE' LONGEST PREFIX MATCH?

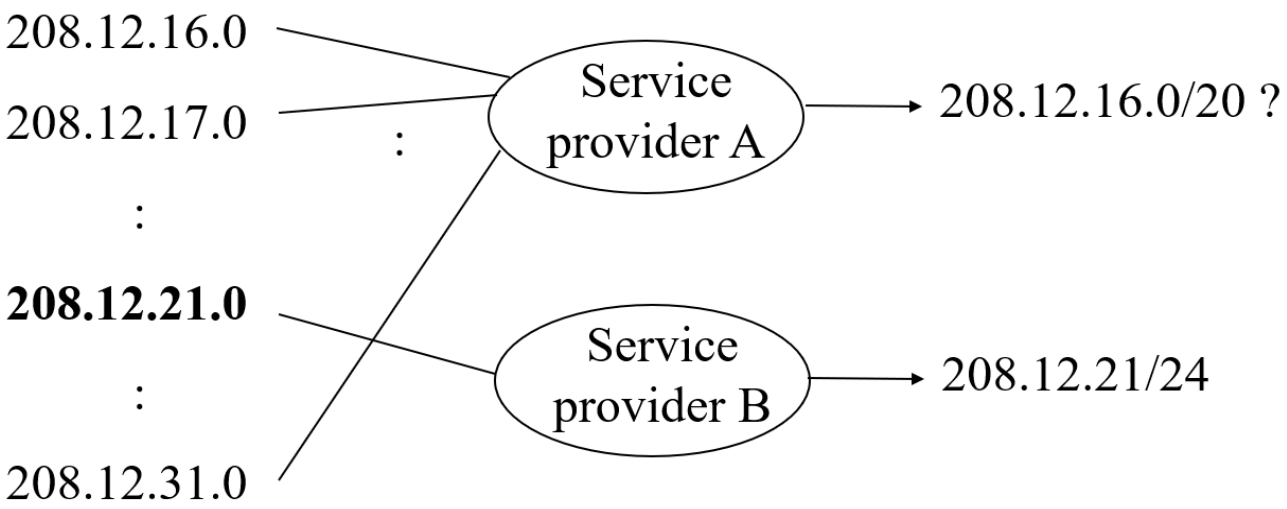
- con CIDR, è possibile prefix overlap
- un indirizzo può corrispondere a più entrate
  - gli indirizzi da 208.12.16.0/24 a 208.12.31.0/24 sono aggregati in 208.12.16.0/20.
  - successivamente, una delle reti con indirizzo 208.12.21.0/24 cambia, ma chiede di mantenere i vecchi inirizzi
  - gli indirizzi precedenti non possono essere più aggregati in una singola rete di indirizzo 208.12.16.0/20.



PERCHE' LONGEST PREFIX MATCH?

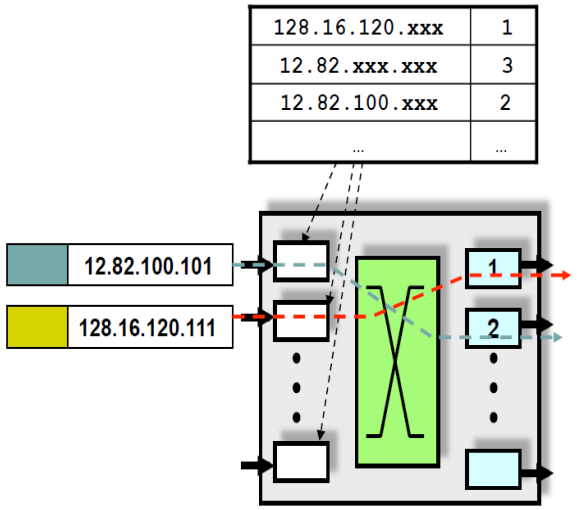
Soluzione

- mantenere la rotta 208.12.16.0/20 ed aggiungere una route separata a 208.12.21.0/24.
- 208.12.21.0/24 viene indicata come eccezione a 208.12.16.0/20.
- usare il longest prefix match affinchè i pacchetti siano inoltrati correttamente a 208.12.21.0/24.



IP LOOKUP

- IP addressing
  - più IP addresses sono aggregati, per aumentare la scalabilità: notazione /n
- Tabella di forwarding: mapping da prefissi IP ad interfacce di output
- Route lookup:
  - individuare il prefisso più lungo nella tabella a comune con l'indirizzo destinazione contenuto nel pacchetto
  - longest Prefix Match (LPM) lookup
  - prefisso più lungo equivale a rotta più "specifica"
    - rete più piccola, informazioni di
    - routing più precise e dettagliate



- Il pacchetto con indirizzo destinazione 12.82.100.101 è inviato all'interfaccia 2, poichè 12.82.100.xxx è il prefisso più lungo a comune con l'indirizzo destinazione del pacchetto

LONGEST PREFIX MATCHING

101.xx.xxx.xxx (/3)	1
1**.***.xxx.xxx(/1)	3
…	. .
…	…

a quale porta inviare il pacchetto con indirizzo destinazione 100.5.6.7?

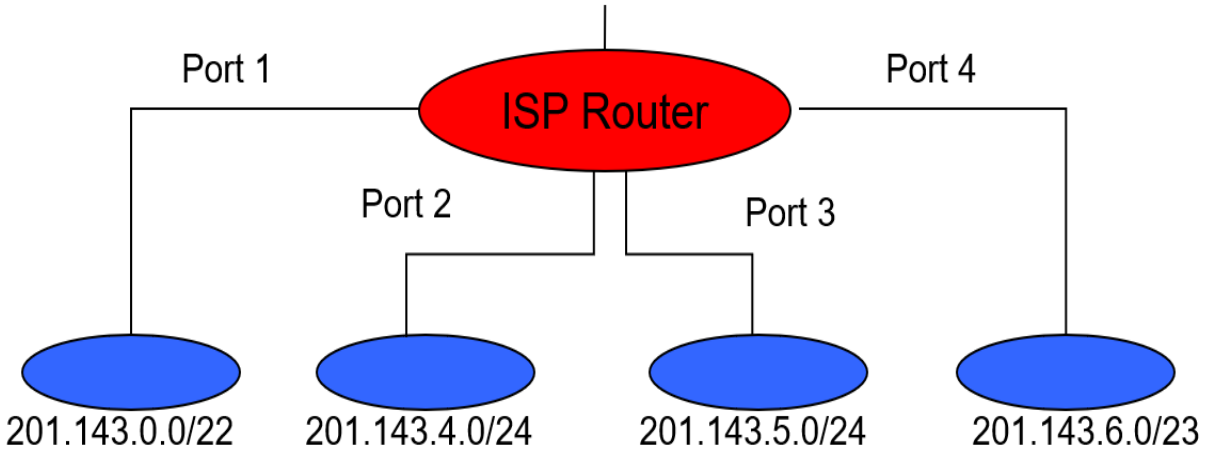
Longest prefix matching

- un semplice algoritmo (non corretto)
  - una wildcard (asterisco) combacia con qualsiasi valore
  - restituire il prefisso con il più alto numero di bit coincidenti
    - non è l'indirizzo più specifico
- se il confronto dà esito positivo per più righe della tabella occorre selezionare l'indirizzo coincidente e con la netmask caratterizzata dal maggior numero di 1

LONGEST PREFIX MATCHING

- Problema diffuso anche in altri contesti:
  - problema dell'autocompletamento in search engine come Google
    - la query di un utente viene confrontata con un insieme di query candidate
    - determinare la query che condivide il prefisso più lungo con la query dell'utente.
    - Le query con il prefisso più lungo vengono mostrate come suggerimenti per l'utente
  - Kademlia Distributed Hash Table (DHT):
    - peer organizzati in una overlay network secondo i loro prefissi logici
    - overlay e ricerche guidate dal prefisso
    - implementata in Bittorrent
  - Patricia trie in Ethereum
- Diverse soluzioni algoritmiche per rendere efficiente la ricerca dei possibili matchings

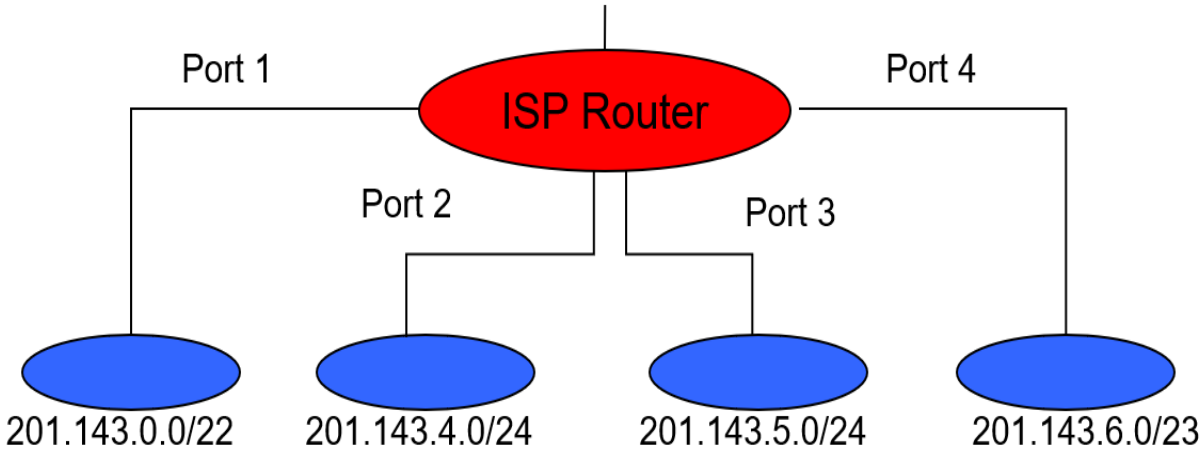
LONGEST PREFIX MATCH



201.143.0.0/22	11001001	10001111	000000––	————————
201.143.4.0/24	11001001	10001111	00000100	————————
201.143.5.0/24	11001001	10001111	00000101	————————
201.143.6.0/23	11001001	10001111	0000011–	————————

- Consider 11001001100011110000010111010010

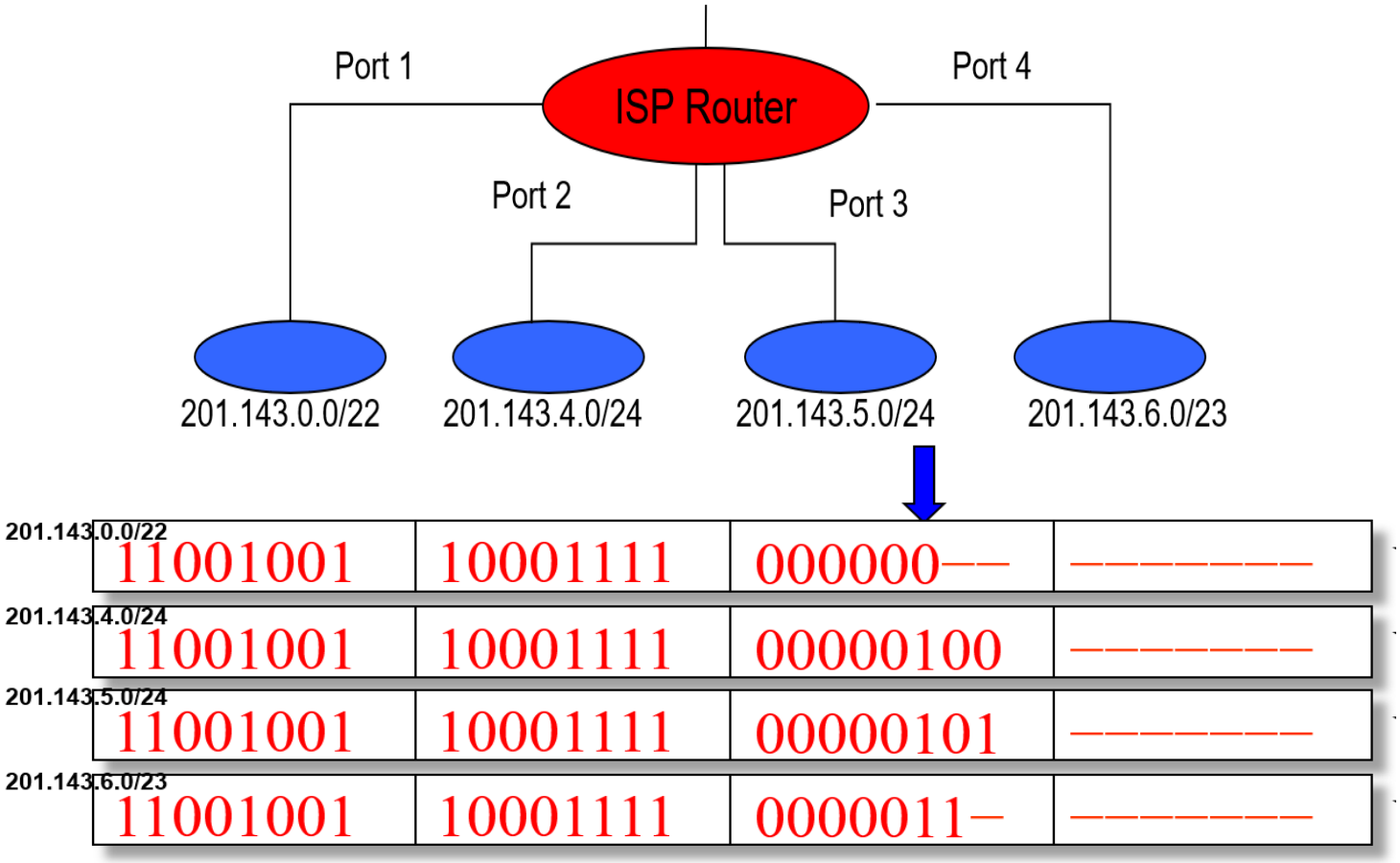
LONGEST PREFIX MATCH



201.143.0.0/22	11001001	10001111	000000––	––––––––
201.143.4.0/24	11001001	10001111	00000100	––––––––
201.143.5.0/24	11001001	10001111	00000101	––––––––
201.143.6.0/23	11001001	10001111	0000011–	––––––––

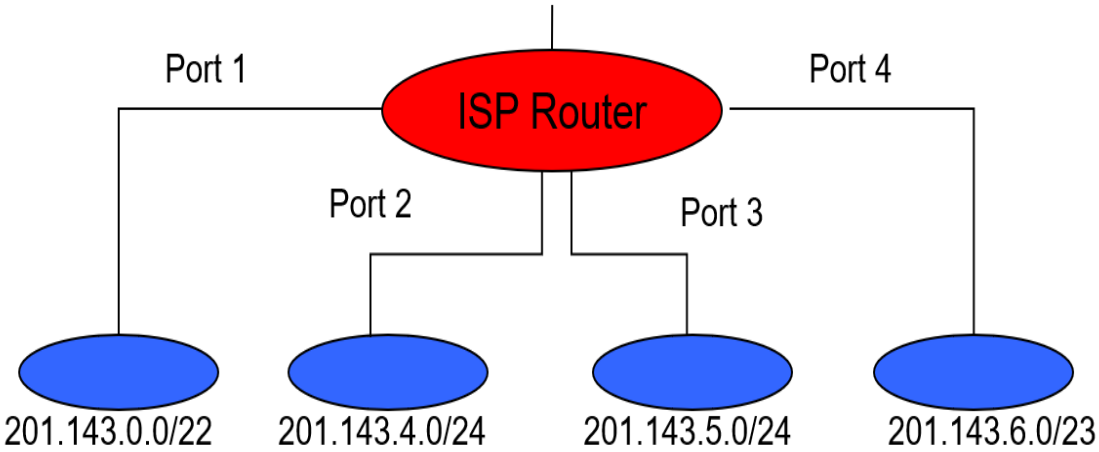
- Consider 11001001100011110000010111010010
  - First 21 bits match 4 partial prefixes

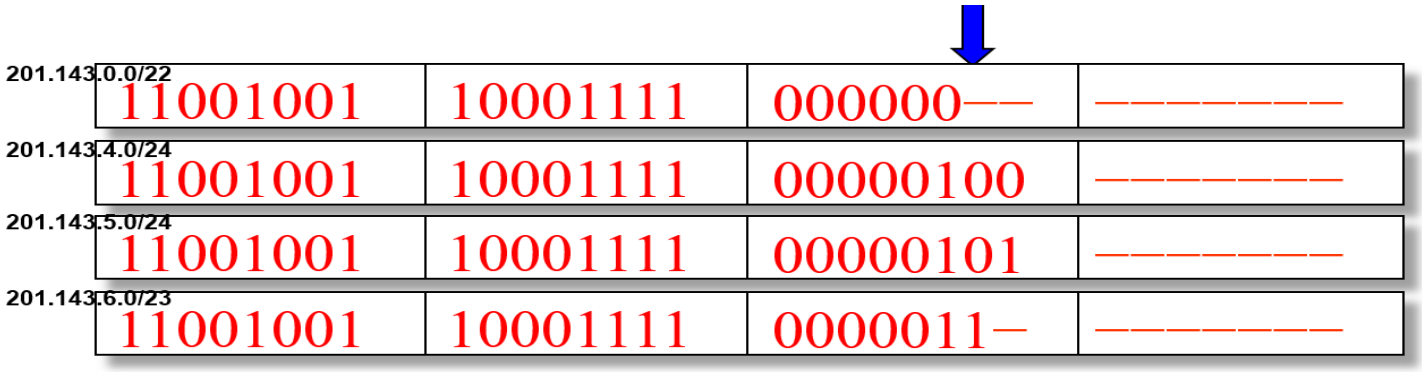
LONGEST PREFIX MATCH



- Consider 11001001100011110000010111010010
  - First 21 bits match 4 partial prefixes
  - First 22 bits match 3 partial prefixes

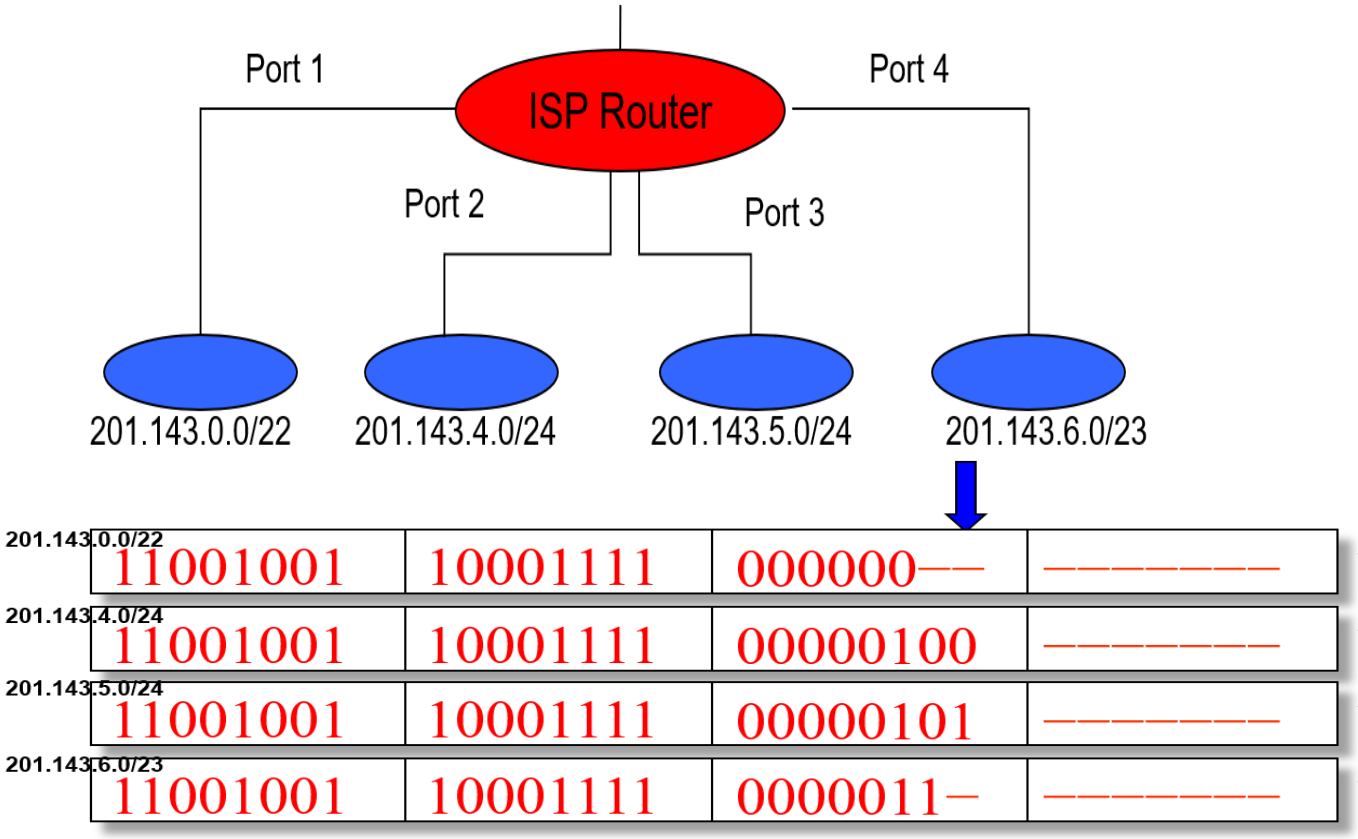
LONGEST PREFIX MATCH





- Consider 11001001100011110000010111010010
  - First 21 bits match 4 partial prefixes
  - First 22 bits match 3 partial prefixes
  - First 23 bits match 2 partial prefixes

LONGEST PREFIX MATCH



- Consider 11001001100011110000010111010010
  - First 21 bits match 4 partial prefixes
  - First 22 bits match 3 partial prefixes
  - First 23 bits match 2 partial prefixes
  - First 24 bits match exactly one full prefix

LOOK UP: IMPLEMENTAZIONE BASE

	Destinazione	Netmask	Etc.
1	0.0.0.0	0.0.0.0	…
2	192.168.2.0	255.255.255.0	…
3	192.168.2.18	255.255.255.255	…

- Datagramma con IP dest. = 192.168.2.22

```
192.168.002.022
255.255.255.255
192.168.002.022 != 192.168.002.018

192.168.002.022
255.255.255.000
192.168.002.000 == 192.168.002.000
```

- La riga 2 è quella giusta (network specific)

LOOK UP: IMPLEMENTAZIONE BASE

- dati in input:
  - indirizzo IP di destinazione del datagramma
  - indirizzo di destinazione e la netmask specificati in ciascuna riga della tabella

- procedura:

  D= Destination IP Address

  considerare le entrate della tabella in ordine decrescente di prefisso (prima le entrate le cui netmask hanno più 1)

  entry= (Network/subnet ID, subnet mask, next-host)

  D1 = subnet mask **&** D (AND bit a bit)

  if (D1 == Network/subnet ID)

  if (next-hop) è una interfaccia

  consegna il datagram alla destinazione

  altrimenti

  consegna il datagram al next-hop

TABLE LOOK UP: TRIE

- problema: alcuni router possono avere tabelle contenenti 100.000 prefissi
- testare sequenzialmente ogni entrata della tabella di routing per trovare una corrispondenza: poco scalabile
  - in media: O(numero delle entrate)
  - soluzione alternativa: usare un trie, ovvero un albero di prefissi
    - memorizza stringhe in una struttura ad alberi
    - radice associata con la stringa vuota
    - una stringa associata ad ogni nodo
    - tutti i discendenti di un nodo condividono un prefisso, che è la stringa associata a quel nodo

TABLE LOOK UP: TRIE

- Un trie binario usato per rappresentare un insieme di prefissi
  - nodo a:"0", node c:"011", ed il nodo i:"1111"
- I nodi ombreggiati sono i prefissi che vengono memorizzati nella forwarding table del nodo
- I nodi b e s sono "eccezioni" al prefisso 0 (nodo a)

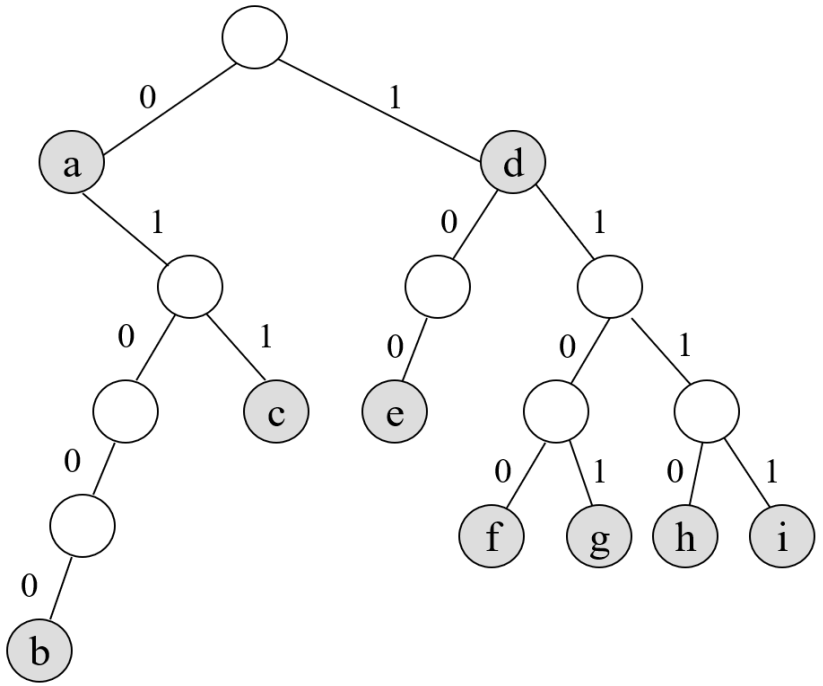
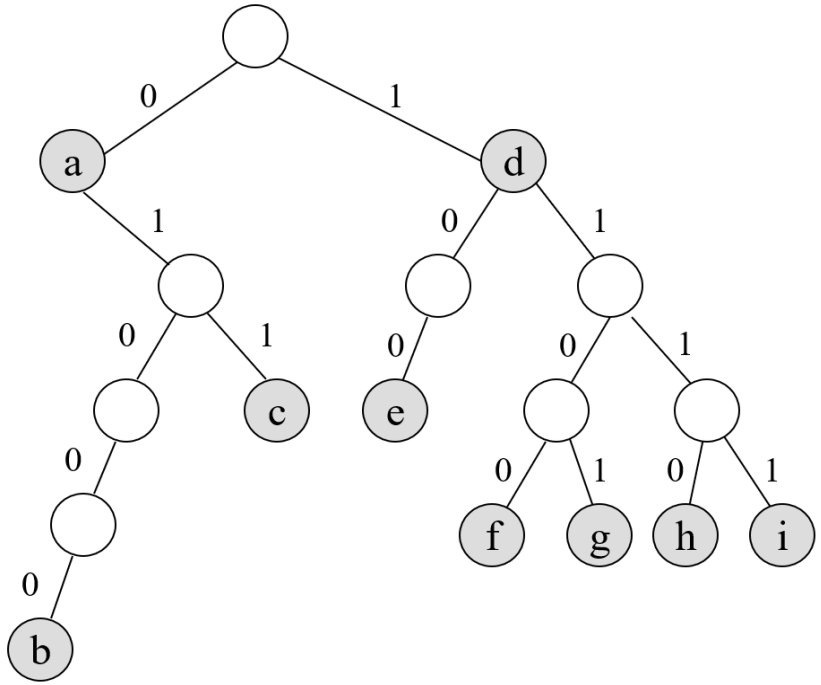


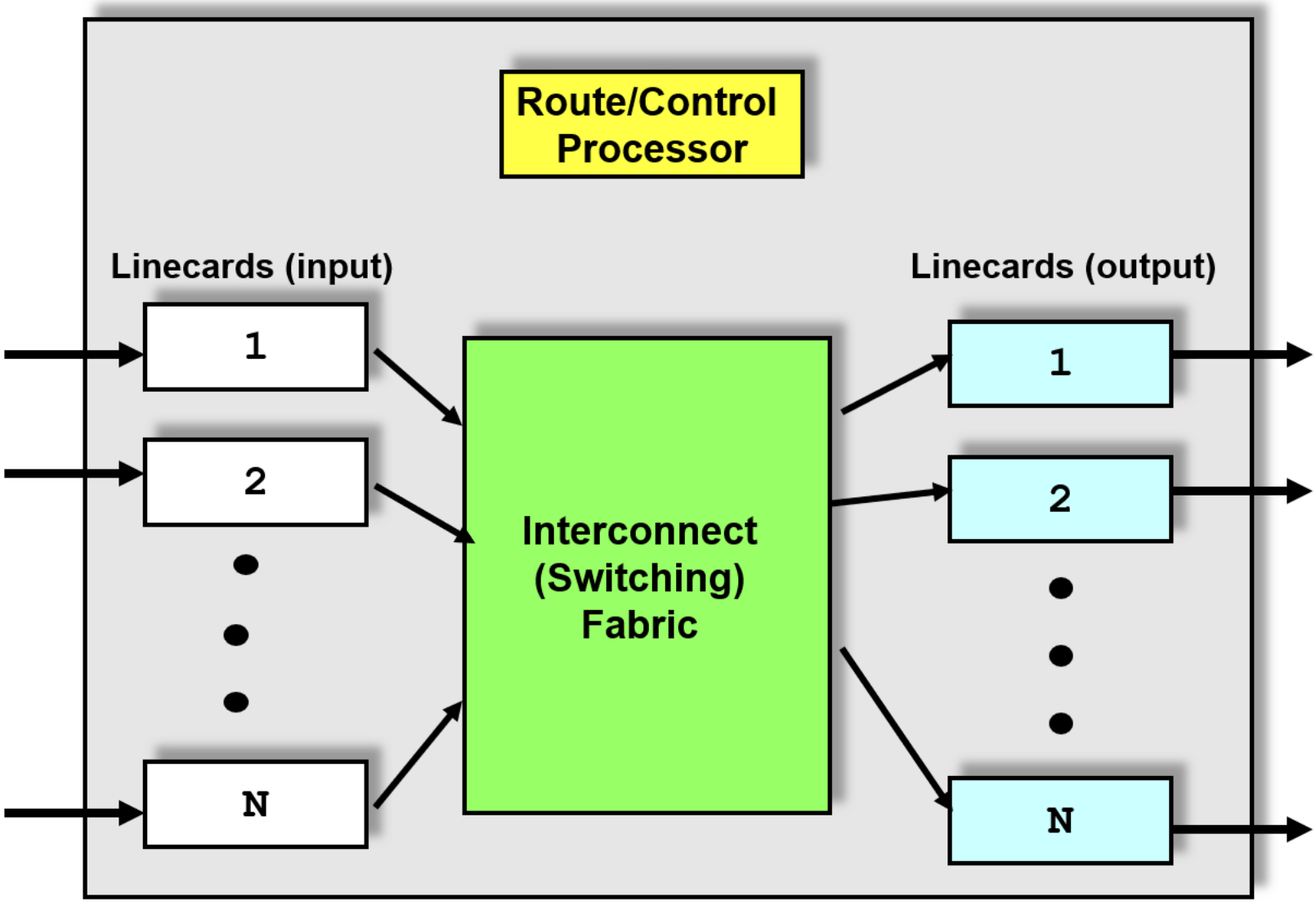
TABLE LOOK UP: TRIE

Data una certa destinazione:

- attraversare l'albero guidati dai bit dell'indirizzo e memorizzare l'ultimo prefisso visitato
- terminazione quando non ci sono più rami percorribili



ARCHITETTURA DI UN ROUTER

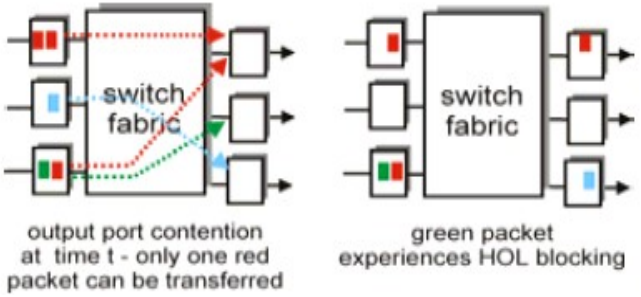


INPUT LINECARDS

- tasks:
  - ricevere i pacchetti in ingresso
    - elaborare i bit dal livello fisico
    - ricostruire il datagram, controllare correttezza ed eventualmente scartarlo (funzioni del livello data link)
  - aggiornare l'header IP
    - TTL, Checksum, Opzioni (possibile), Fragmento (possibile)
  - decentralized switching:
    - IP lookup (longest prefix lookup), integrato nelle porte di input per rendere più efficiente il processo
  - accodare il pacchetto nelle code della switching fabric
- sfida: velocità!
  - elaborare i dati in arrivo alla "velocità della linea"
  - 100B packets @ 40Gbps: un nuovo pacchetto ogni 20 nano secs!
- implementato tipicamente con hardware specializzato
  - ASICs, "network processors" specializzati

INPUT LINECARDS: QUEUEING

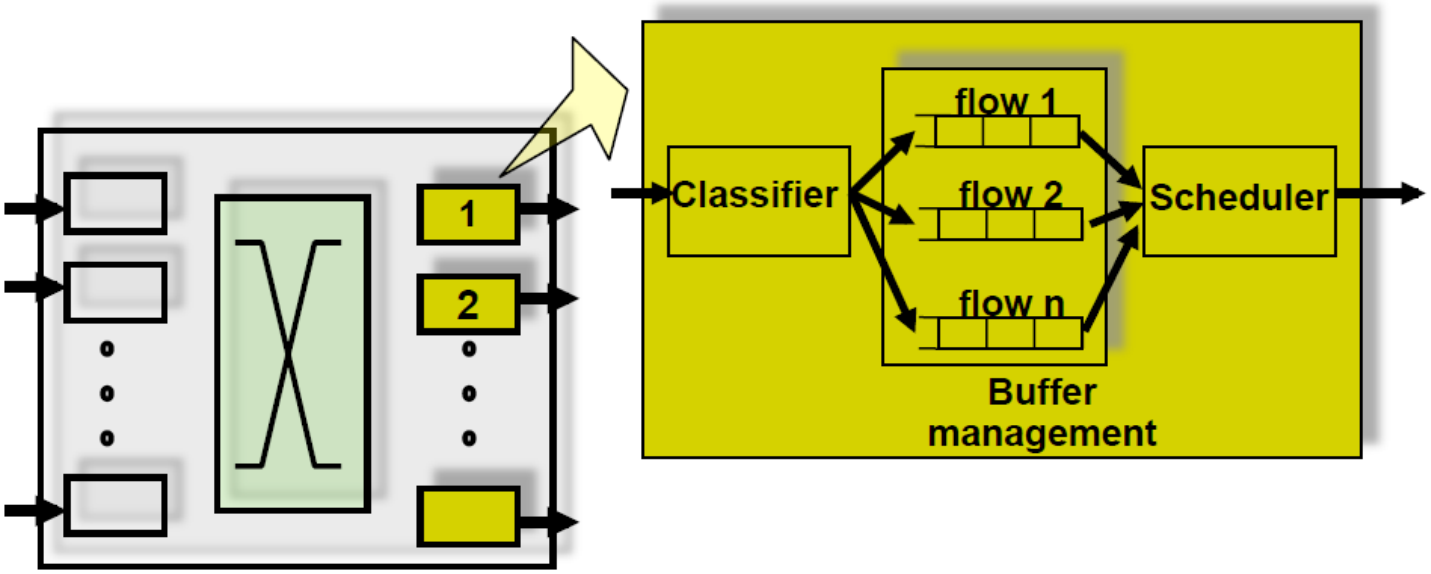
- switching fabric può essere più lenta della capacità di elaborazione combinata di tutte le line cards
- necessario integrare code nelle porte di input
- head-of-the-line(HOL) blocking: i datagram incima alla coda impediscono agli altri di avanzare nelle code



OUTPUT LINECARDS

Funzioni delle output linecards

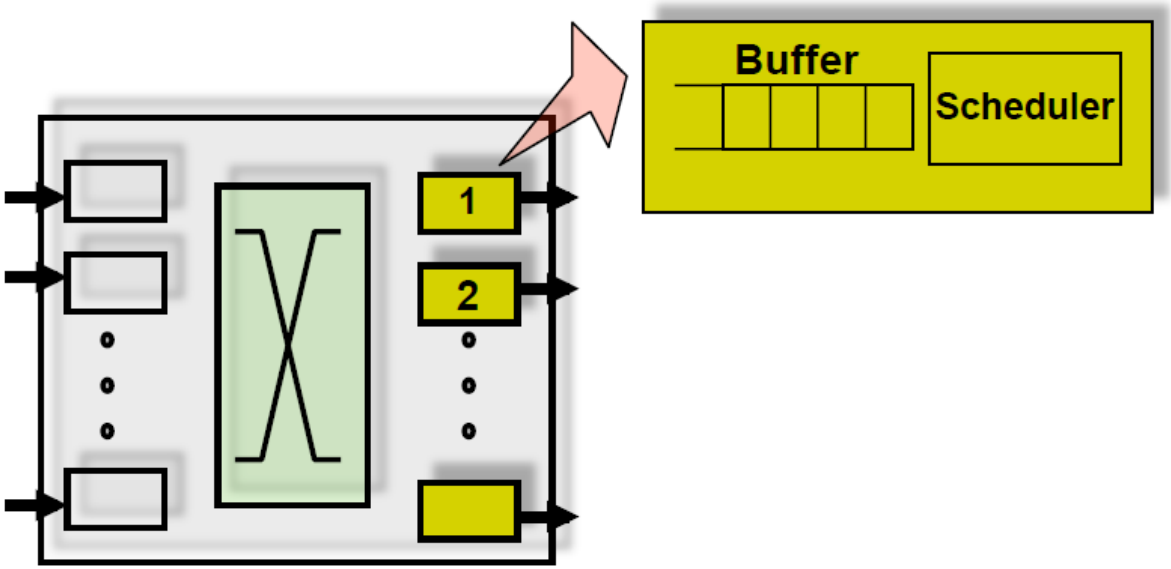
- packet classification: mappare ogni pacchetto su un flusso distinto
- buffer management: decidere quando e quale pacchetto eliminare, nel caso di congestione
- scheduler: decidere quando e quale pacchetto trasmettere
  - In base a queste strategie possibile definire diverse politiche



LA SOLUZIONE PIU' SEMPLICE

Nessuna classificazione

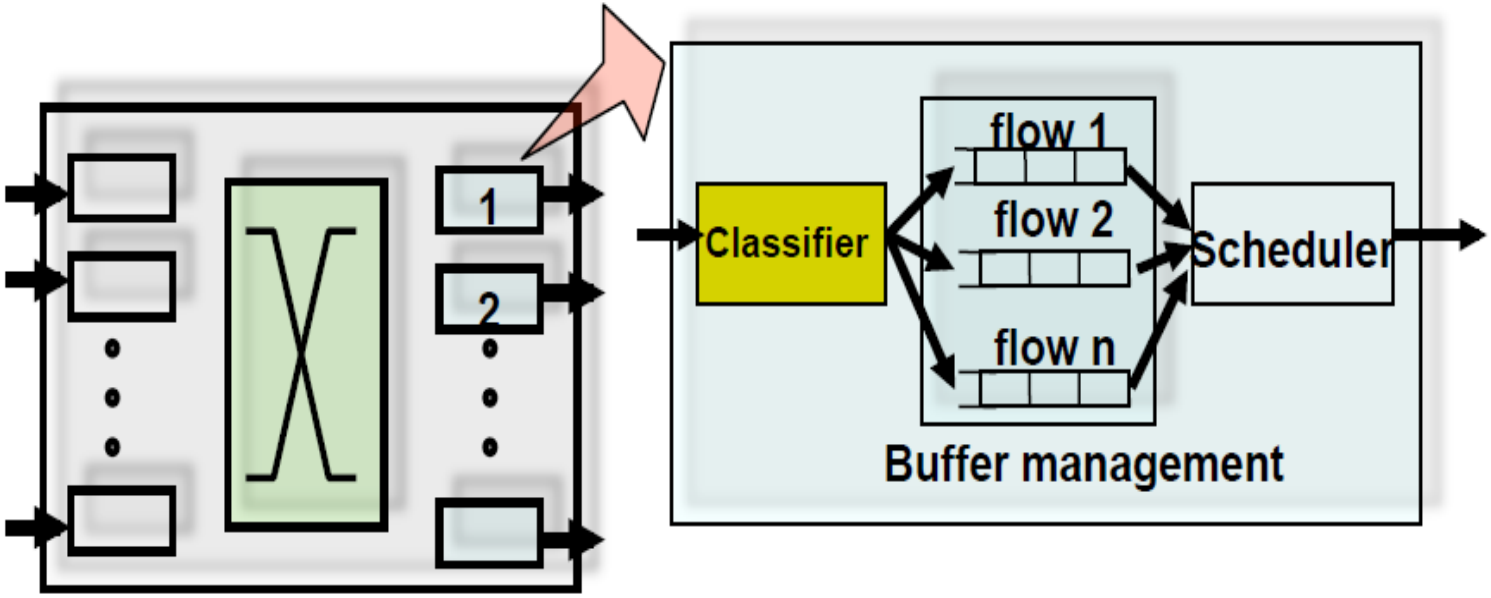
- drop-tail buffer management: quando il buffer è pieno eliminare il pacchetto in arrivo
- First-In-First-Out (FIFO) Scheduling: i pacchetti vengono schedulati nell'ordine con cui arrivano sulla output linecard



CLASSIFICAZIONE DEI PACCHETTI

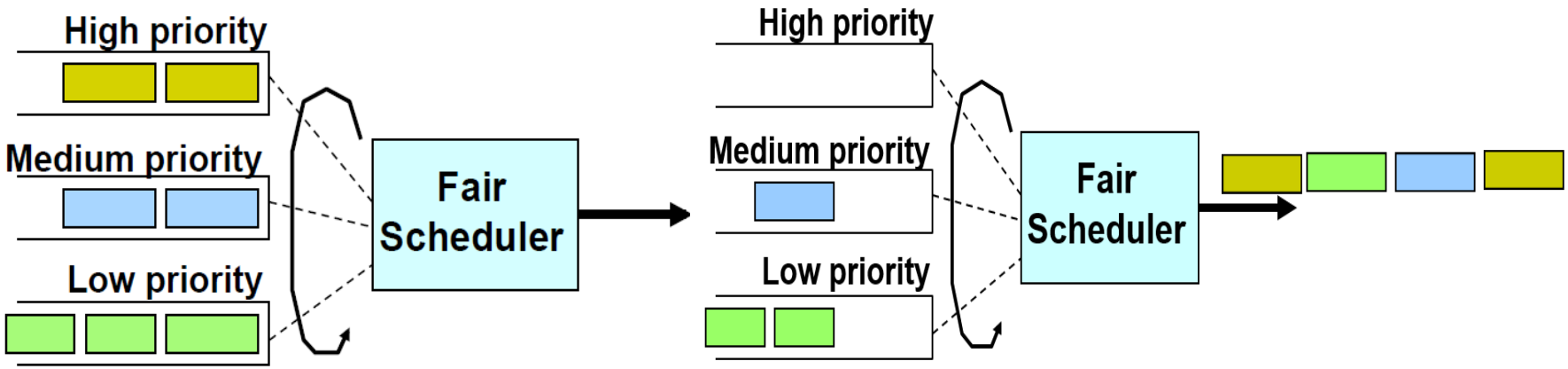
Classificazione dei pacchetti IP sulla base del valore di alcuni campi nell'header del pacchetto

- indirizzo IP sorgente/destinazione (32 bits)
- numero di porta TCP sorgente/destinazione (16 bits)
- tipo del servizio (TOS) byte (8 bits)
- tipo del protocollo (8 bits)



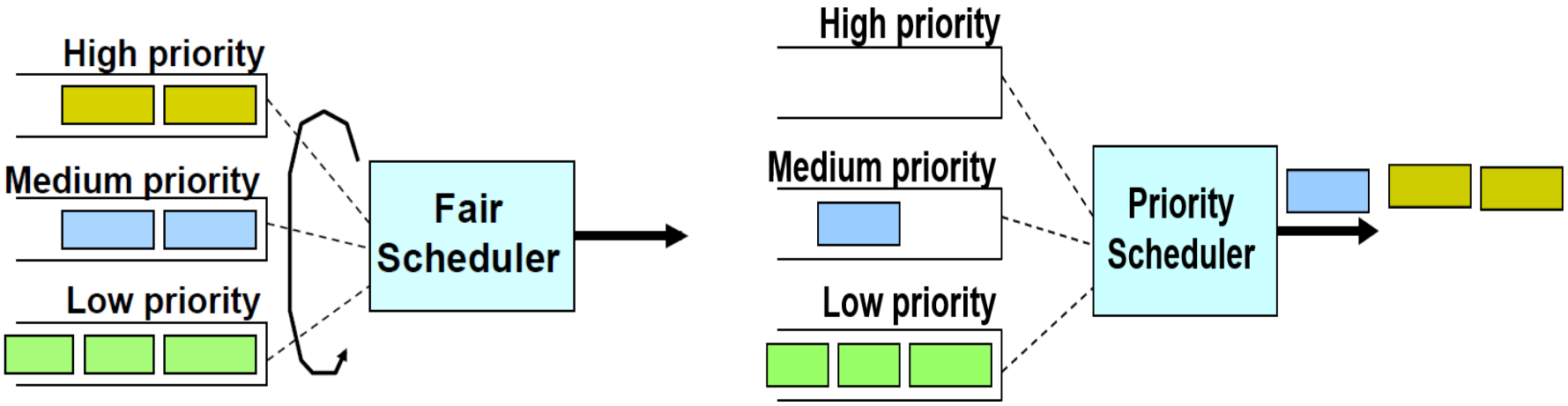
SCHEDULING

- una coda per ogni flusso
  - lo scheduler decide quando e da quale coda deve essere prelevato un valore per spedirlo
- obiettivi dell'algoritmo di scheduling:
  - veloce!
  - dipende dalla politica implementata (fairness, priorità, etc.)
- Round robin: pacchetti serviti da ogni coda, a turno.

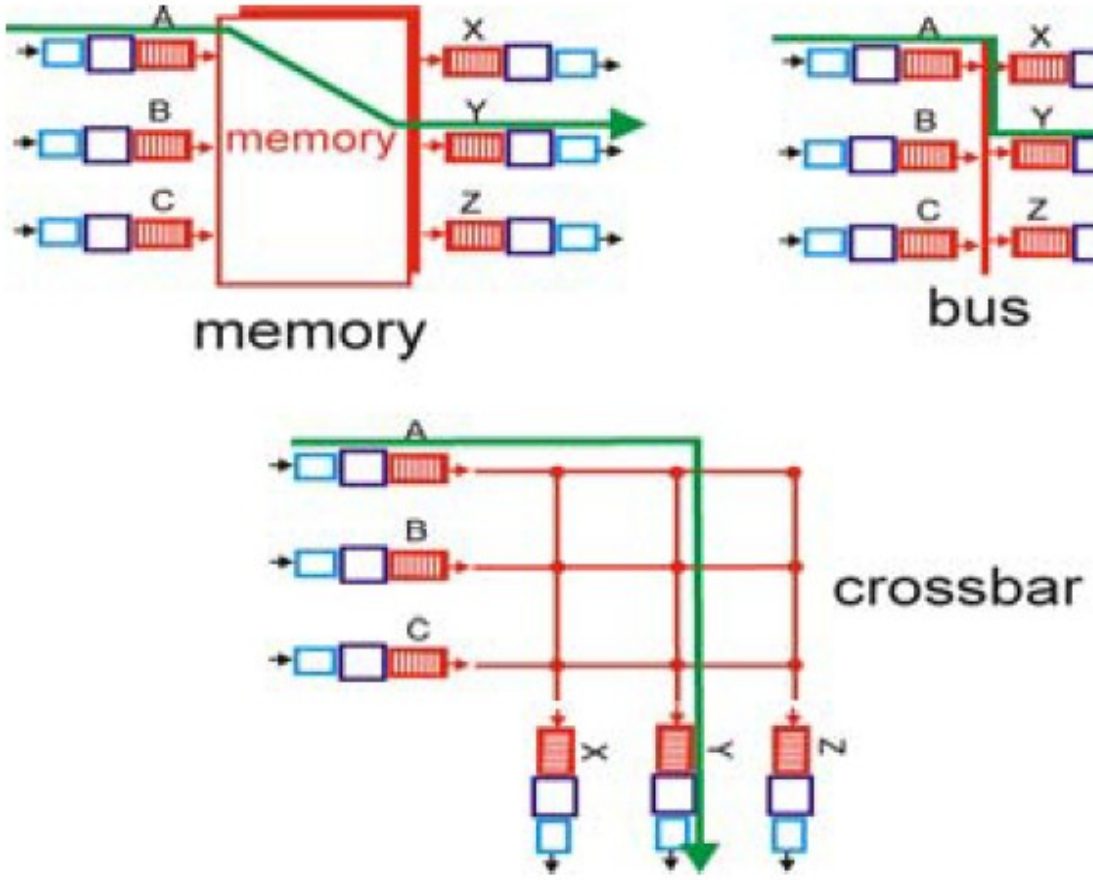


SCHEDULING

- una coda per ogni flusso
  - lo scheduler decide quando e da quale coda deve essere prelevato un valore per spedirlo
- obiettivi dell'algoritmo di scheduling:
  - veloce!
  - dipende dalla politica implementata (fairness, priorità, etc.)
- Priority scheduler: pacchetti nella coda a più alta priorità sono inviati prima dei pacchetti nelle code a priorità minore



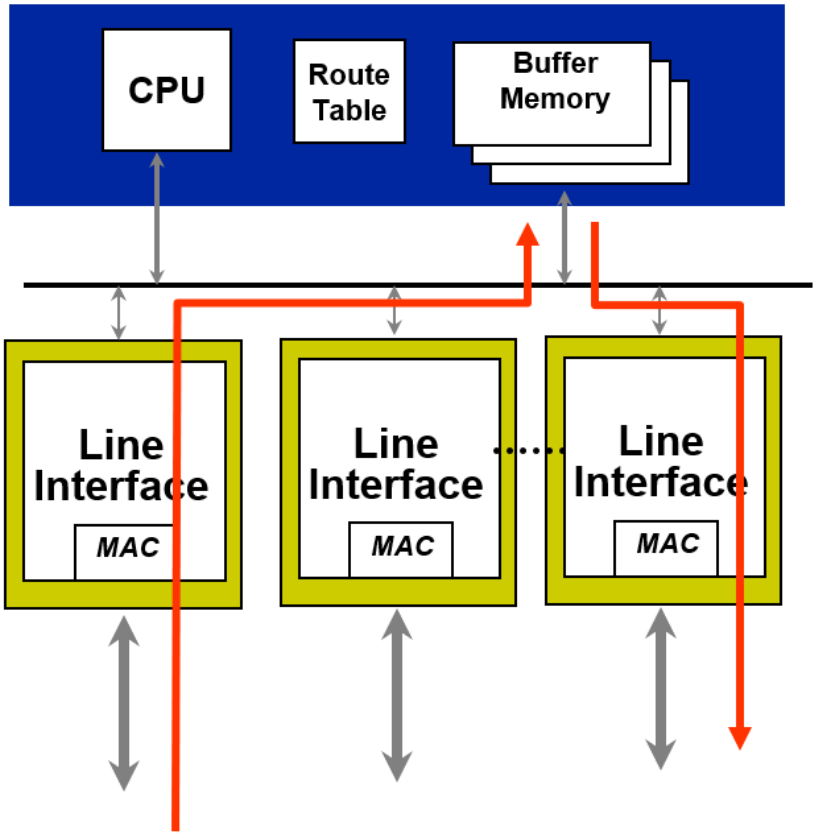
SWITCHING FABRIC: TIPI



SWITCHING VIA SHARED MEMORY

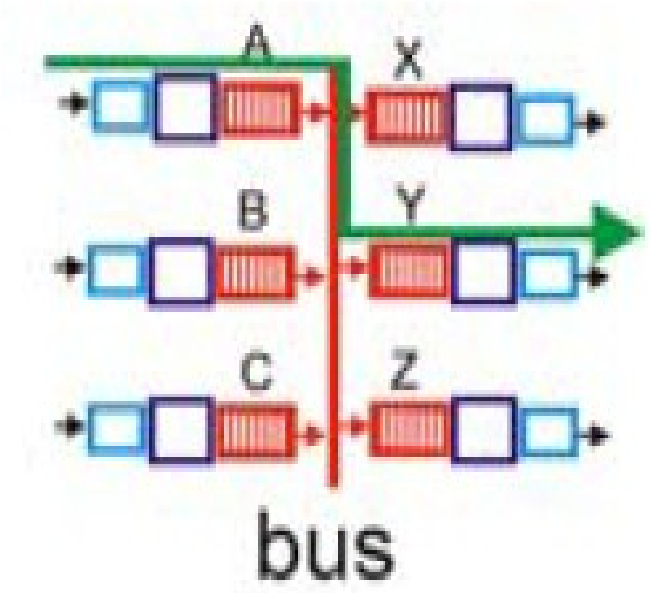
Routers di prima generazione

- una CPU sul chip del router
- il pacchetto è copiato dalla CPU dalla input alla output linecard
- velocità limitata dalla banda di memoria (attraversamento di due bus per ogni datagram)
- poco efficiente: limitato dalla banda della memoria condivisa



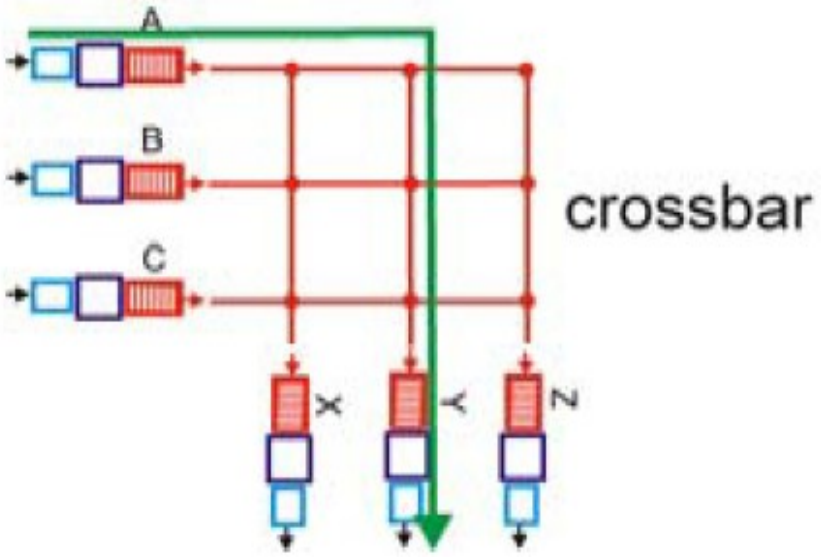
SWITCHING VIA BUS

- trasferimento del datagram dalla input memory verso la ouptput memory via shared bus
- bus contention: switching speed limitato dalla banda del bus
- 1 Gbps bus, Cisco 1900: velocità sufficiente per access ed enterprise routers (non regional o backbone)



PINT-TO-POINT SWITCH

- Per superare la limitazione di banda di un solo bus
- Banyan networks, altre reti di interconnessione inizialmente proposte per interconnettere i processori in un multiprocessor
- Cisco 12000: switches Gbps attraverso la rete di interconnessione



IL PROTOCOLLO IP....

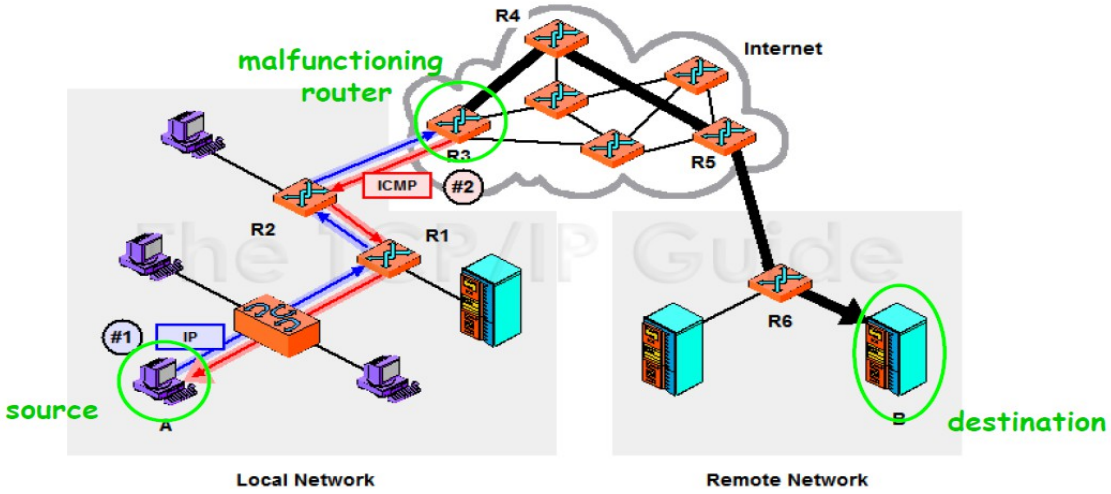
- offre un servizio di tipo **best effort**
  - non garantisce la corretta consegna dei datagrammi
  - se necessario si affida a protocolli affidabili di livello superiore (TCP)
- è quindi necessario un protocollo di controllo
  - gestione di situazioni anomale
  - notifica di errori o di irraggiungibilità della destinazione
  - gestione della rete

ICMP (Internet Control Message Protocol)

- ICMP segnala solamente errori e malfunzionamenti, ma non esegue alcuna correzione
- ...ICMP **non rende affidabile** IP

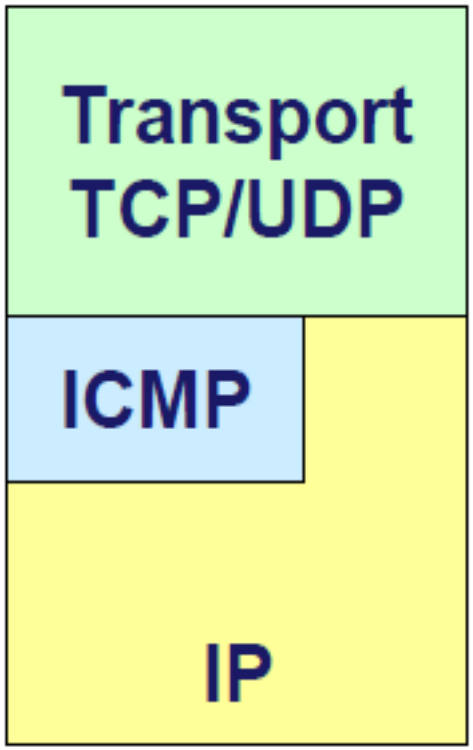
IL PROTOCOLLO IP....

- offre un servizio di tipo best effort
  - non garantisce la corretta consegna dei datagrammi
  - se necessario si affida a protocolli affidabili di livello superiore (TCP)
- cosa fare
  - se un router scarta un datagram perchè non riesce a trovare una rotta verso l'host finale o perchè il TTL è 0?
  - se l'host finale deve scartare un certo numero di pacchetti perchè non ha ricevuto tutti i pacchetti entro un certo intervallod i tempo?
  - come fare a verificare se un altro host/router è "vivo"?



ICMP: INTERNET CONTROL MESSAGE PROTOCOL

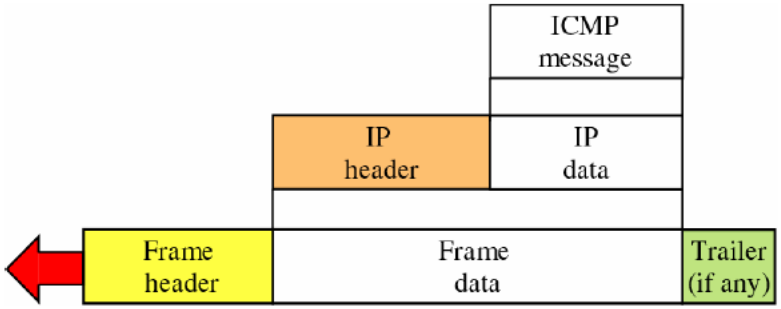
- protocollo utilizzato dai routers per scambiare informazione di controllo
  - errori
  - messaggi di controllo
- segnala solamente errori e malfunzionamenti, ma non esegue alcuna correzione
  - ….non rende affidabile IP
- dal punto di vista della strutturazione a livelli, ICMP è un protocollo separato
  - usa IP per trasportare i messaggi
  - si può collocare a livello superiore ad IP
- in pratica, però, è integrato ad IP
  - "companion protocol" di IP



ICMP: INTERNET CONTROL MESSAGE PROTOCOL

i messaggi ICMP

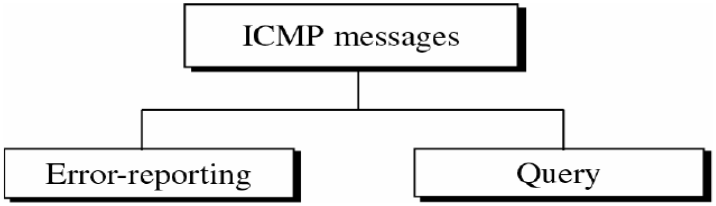
- sono incapsulati in datagrammi IP
  - non passati direttamente al livello data-link
- sono identificati nell'header del pacchetto IP, dal valore 1 del campo Protocol
- il payload del messaggio IP è il messaggio ICMP
- ed il payload contiene, a sua volta, alcune informazioni sul datagram IP ricevuto
  - ad esempio il datagram che ha generato l'errore...



The ultimate destination of an ICMP message is not an application program or user on the destination machine, but the Internet Protocol software of that machine!

TIPI DI MESSAGGI ICMP

- Error Reporting
  - riportano problemi che un router oppure un host destinazione possono incontrare nell'elaborazione di uno specifico pacchetto IP
- Query
  - supportano un host o il network manager nel reperimento di informazioni specifiche su un router o un altro host
    - coppie richiesta/risposta



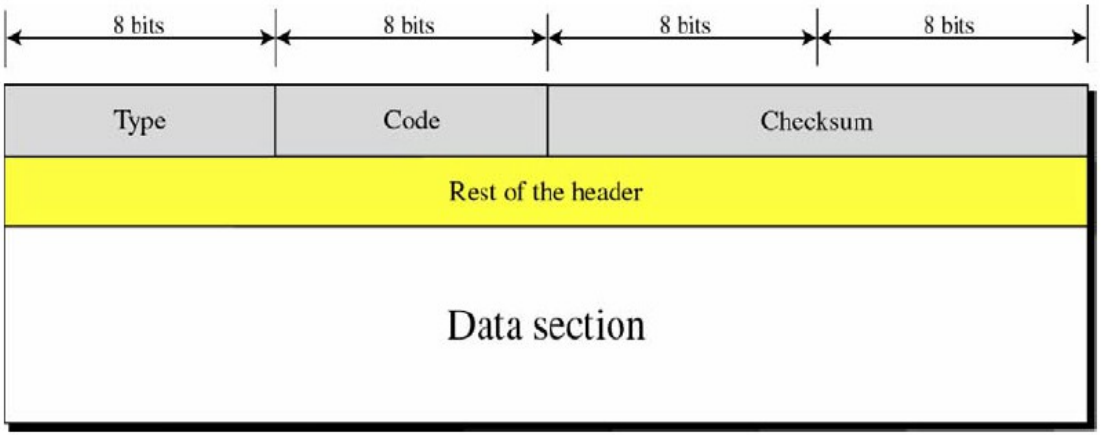
Category	Type	Message
Error-reporting Messages	3	Destination Unreachable
	4	Source Quench
	11	Time Exceeded
	12	Parameter Problem
	5	Redirection

Category	Type	Message
Query Messages	8 / 10	Echo Request / Reply
	13 / 14	Timestamp Request / Reply
	17 / 18	Address Mask Request / Reply
	10 / 9	Router Solicitation or Advertisement

FORMATO DEI MESSAGGI

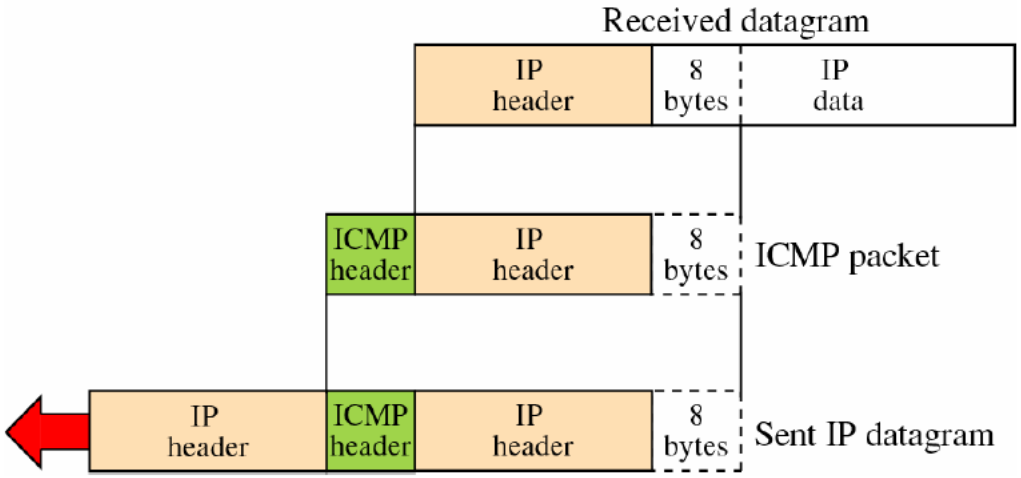
Primi 4 byte dell'header uguali per tutti i messaggi, gli ultimi 4 differiscono

- Type definisce il tipo di messaggio ICMP
- Code specifica la ragione per cui si invia quel tipo di messaggio
- Checksum nell'header, ma calcolata sull'intero messaggio
- Data section
  - in messaggi di errore riporta informazione per individuare il pacchetto originale che ha generato l'errore
  - in query messages riporta informazioni in base al tipo della query



ICMP ERROR REPORTING

- ICMP non corregge errori, semplicemente li segnala
  - correzione lasciata ai protocolli di più alto livello
- i messaggi di errore sono sempre spediti al mittente originale
- la sezione dati (payload) di tutti i messaggi di errore contiene l'header IP del datagram originale + 8 bytes del payload di quel datagram
  - nel caso dei protocolli UDP e TCP i primi 8 byte forniscono informazioni su porta e numero di sequenza
  - il mittente può informare UDP o TCP dell'errore



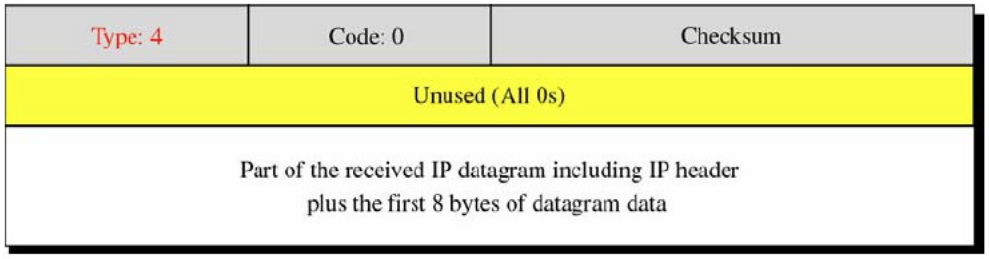
ICMP ERROR REPORTING

Source Quench (type 4)

- inviato quando un router scarta un datagram a causa delle congestione nelle sue code
- due scopi
  - informare il mittente che il datagram è stato scartato
  - informare il mittente che c'è congestione nella rete
- Il mittente può ridurre la frequenza con cui spedisce i datagram implementando una forma di controllo della congestione.

why optional?!

	Implementation
Host	Optional.
Router	Optional.



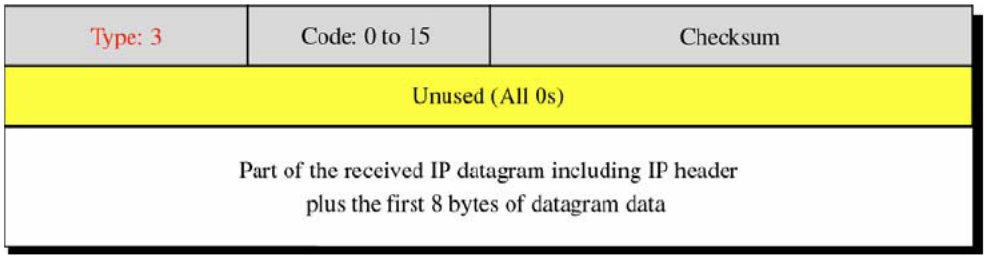
ICMP ERROR REPORTING

Destination Unreachable (tipo 3)

- un router non è in grado di inoltrare un datagram
  - non possiede una rotta appropriata nella sua tabella di routing
  - non può fragmentare un datagram, perchè il bit no-fragment è settato
- scarta il datagram e restituisce un messaggio ICMP "destination unreachable" al mittente

In questo caso usato il campo code per indicare la causa

	Implementation
Host	Mandatory.
Router	Mandatory.



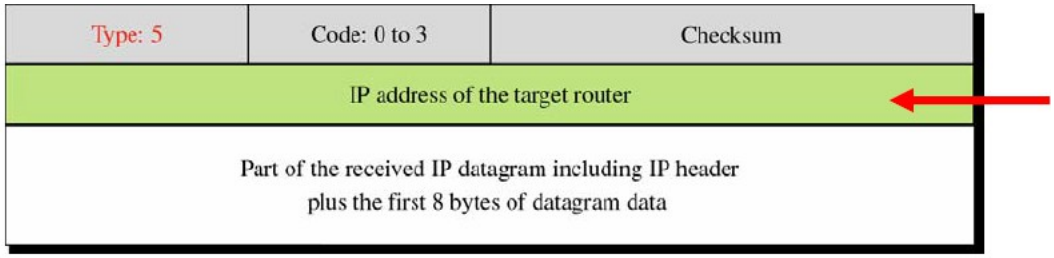
Code 2: (destination host) **protocol is unreachable**
Code 3: (destination host) **port is unreachable – application program is not running presently**
Code 7: **destination host is unknown**
Code 9: **communication with destination network is administratively prohibited**

ICMP ERROR REPORTING

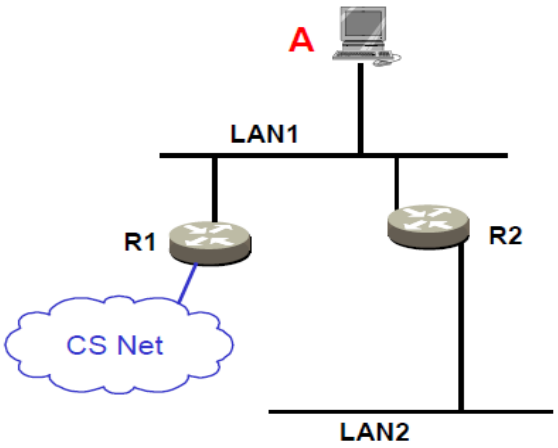
Redirection (type 5)

- un host in genere al momento del bootstrap possiede una tabella di routing piccola, che viene quindi aggiornata con il temo
- questo messaggio supporta l'aggiornamento della tabella di routing
  - all'inizio poche entrate nella tabella di routing
  - possibilità di inviare un datagram al router meno opportuno.
  - questo router invia il datagram al router corretto
  - per segnalare l'aggiornamento della routing table, il router invia un messaggio all'host per informarlo che esiste una rotta migliore verso la destinazione

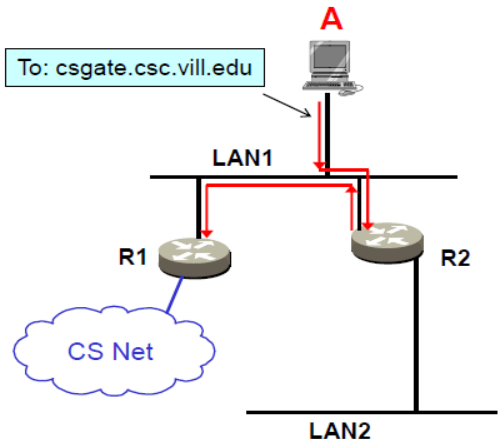
	Implementation
Host	
Router	Mandatory.



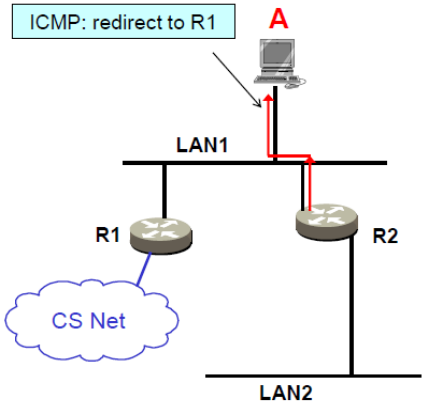
ICMP ERROR REPORTING: RIDIREZIONE



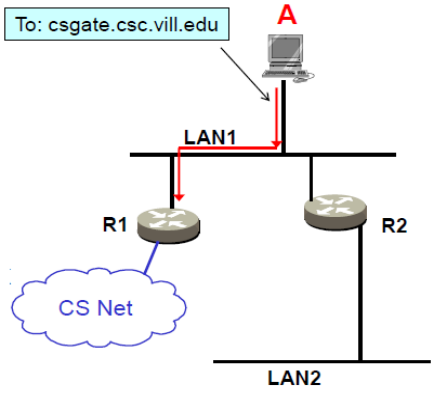
R2 è il default router dell'host A



Quando A manda un messaggio a Campus Net lo manda al default router (R2), che poi lo inoltra ad R1



R2 invia anche un messaggio *ICMP Redirect* ad A, notificandogli di usare R1 per le connessioni a CS Net



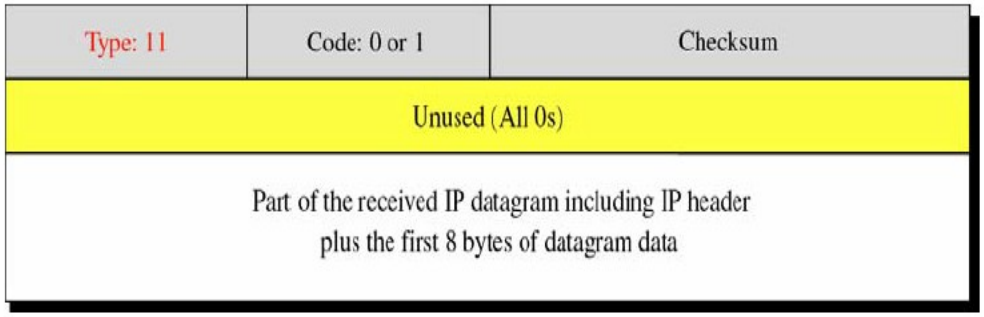
A quindi invia pacchetti direttamente ad R1

ICMP ERROR REPORTING

Time Exceeded (type 11)

- messaggio inviato al mittente quando il valore del TTL è 0 (code 0)
- usato da un host destinazione per mostrare che non tutti i frammenti sono arrivati a destinazione entro un intervallo di tempo (code 1)

	Implementation
Host	Optional.
Router	Mandatory.

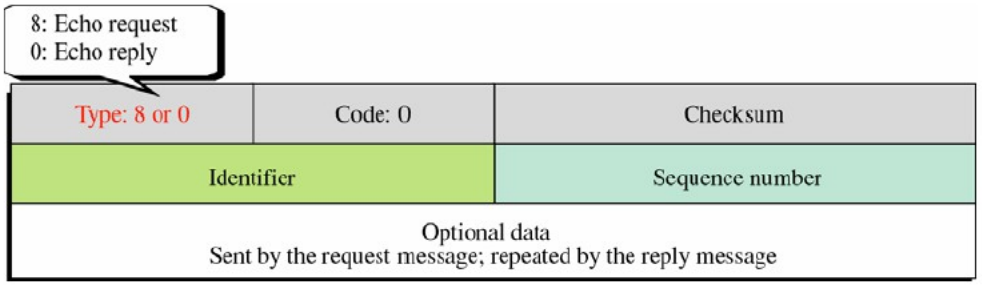


MESSAGGIO DI QUERY

Echo request e Reply

- usati come supporto alla diagnosi della rete
- messaggi combinati determinano se due sistemi possono comunicare
  - si invia un messaggio di "echo request" al nodo che si vuole testare.
  - il campo opzionale contiene un campo dati che contiene un messaggio che deve essere rimandato esattamente uguale al mittente in un messaggio di "echo reply"
- 'echo request'and 'echo reply' usati come base per implementare il messaggio ping
  - controllare se un host è raggiungibile e funzionante

	Implementation
Host	Mandatory.
Router	Mandatory.



MESSAGGIO DI QUERY

Timestamp Request and Reply

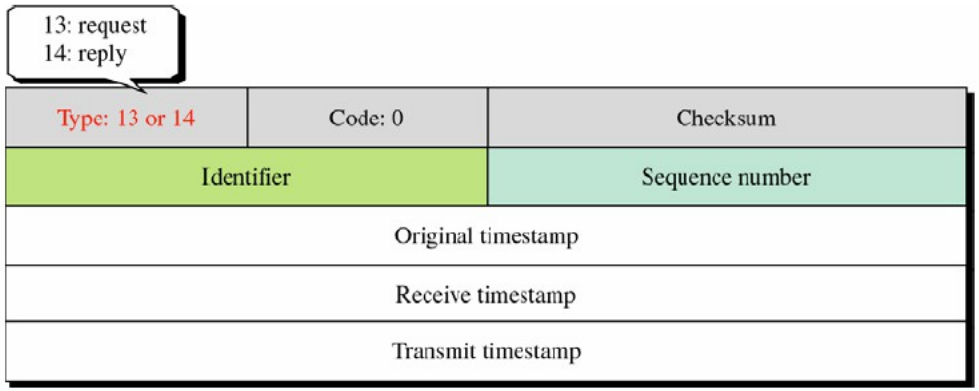
- per determinare il RTT per scambiare un datagram tra due host o routers
  - original timestamp riempito dal mittente al momento dell'invio del messaggio
  - receive timestamp riempito dal destinatario quando la richiesta viene ricevuta
  - transmit timestamp riempito dal destinatario all'invio della 'timestamp reply'
  - calcolo del roundtrip tiene di conto che i due clock non sono sincronizzati

ending time= value of receive timestamp-value of original timestamp

receiving time= time the packet returned -value of transmit timestamp

RTT = sending time+ receiving time

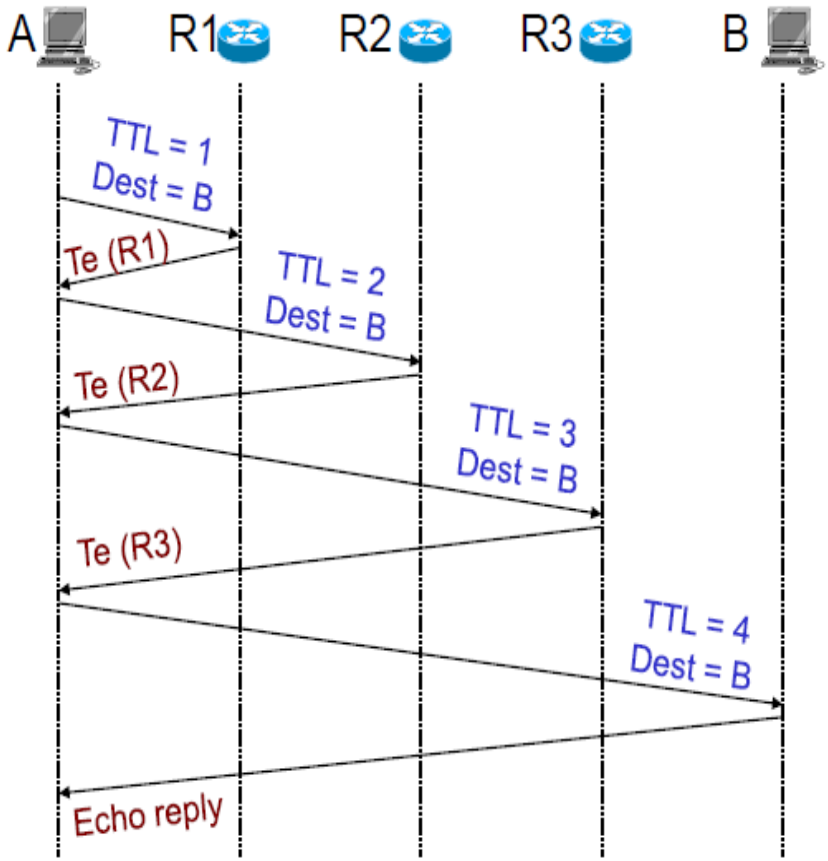
	Implementation
Host	Optional.
Router	Optional.



APPLICAZIONI: TRACEROOT (O TRACERT)

Per determinare la rotta

- l'host invia un pacchetto con TTL=1 (hop)
- il primo router scarta il pacchetto ed invia un pacchetto con il messaggio ICMP "Time Exceeded"
- quando l'host riceve il pacchetto ICMP
  - registra l'indirizzo del mittente (il router che ha scartato il messaggio)
  - incrementa TTL
  - ripete il procedimento, fino a che riceve un messaggio dall'host destinazione oppure si verifica un errore.



APPLICAZIONI: TRACEROOT (O TRACERT)

```
C:\Users\ricci>
C:\Users\ricci>
C:\Users\ricci>
C:\Users\ricci>
C:\Users\ricci>
C:\Users\ricci>
C:\Users\ricci>
C:\Users\ricci>
C:\Users\ricci>
C:\Users\ricci>
C:\Users\ricci>tracert

Sintassi: tracert [-d] [-h max_salti] [-j elenco-host] [-w timeout]
                  [-R] [-S indorig] [-4] [-6] nome_destinazione

Opzioni:
    -d                  Non risolve gli indirizzi in nome host.
    -h max_salti        Numero massimo di punti di passaggio per ricercare
                        la destinazione.
    -j elenco-host      Instradamento libero lungo l'elenco host (solo IPv4).
    -w timeout          Timeout in millisecondi per ogni risposta.
    -R                  Traccia percorso andata e ritorno (solo IPv6).
    -S indorig          Indirizzo di origine da utilizzare (solo IPv6).
    -4                  Impone l'uso di IPv4.
    -6                  Impone l'uso di IPv6.
```

APPLICAZIONI: TRACEROOT (O TRACERT)

```
C:\Users\ricci>tracert  google.com

Traccia instradamento verso google.com [172.217.18.46]
su un massimo di 30 punti di passaggio:

  1     1 ms     1 ms     2 ms  modemtim
  2     *        *        *     Richiesta scaduta.
  3    30 ms    25 ms    27 ms  172.18.32.252
  4    26 ms    27 ms    29 ms  172.18.34.64
  5    30 ms    30 ms    29 ms  172.19.241.125
  6    32 ms    32 ms    32 ms  etrunk44.milano50.mil.seabone.net [93.186.128.108]
  7    31 ms    30 ms    33 ms  74.125.146.168
  8    31 ms    31 ms    35 ms  108.170.245.88
  9    37 ms    38 ms    38 ms  209.85.142.221
 10    38 ms    38 ms    38 ms  108.170.252.241
 11    39 ms    37 ms    37 ms  72.14.233.69
 12    38 ms    37 ms    36 ms  ham02s12-in-f14.1e100.net [172.217.18.46]

Traccia completata.
```

APPLICAZIONI: PING

```
C:\Users\ricci>ping

Sintassi: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
          [-r count] [-s count] [[-j host-list] | [-k host-list]]
          [-w timeout] [-R] [-S srcaddr] [-c compartment]
          [-4] [-6] target_name

Opzioni:
    -t             Esegue il ping dell'host specificato finché non viene
                   interrotto. Per visualizzare le statistiche e continuare -
                   digitare Control-Break; Per interrompere - digitare
                   Control-C.
    -a             Risolve gli indirizzi in nomi host.
    -n count       Numero di richieste echo da inviare.
    -l size        Dimensioni del buffer di invio.
    -f             Imposta il contrassegno per la disattivazione della
                   frammentazione nel pacchetto (solo IPv4).
    -i TTL         Durata (TTL, Time To Live).
    -v TOS         Tipo di servizio (TOS, Type Of Service) (solo IPv4.
                   Questa impostazione è deprecata e non ha alcun effetto sul
                   campo del tipo di servizio nell'intestazione IP).
    -r count       Registra la route per il conteggio degli hop (solo IPv4).
    -s count       Timestamp per il conteggio degli hop (solo IPv4).
    -j host-list   Route di origine libera lungo l'elenco host (solo IPv4).
    -k host-list   Route di origine vincolata lungo l'elenco host (solo IPv4).
    -w timeout     Timeout in millisecondi per l'attesa di ogni risposta.
    -R             Usa l'intestazione di routing anche per il test del routing
                   inverso (solo IPv6). In base a RFC 5095 l'utilizzo di questa
                   intestazione di routing è deprecato. Alcuni sistemi
                   potrebbero ignorare le richieste echo se viene utilizzata
                   questa intestazione.
    -S srcaddr     Indirizzo di origine da utilizzare.
    -c compartment Identificatore del raggruppamento di routing.
    -p             Esegue il ping dell'indirizzo di un provider
                   di virtualizzazione di rete di Hyper-V.
    -4             Impone l'utilizzo di IPv4.
    -6             Impone l'utilizzo di IPv6.
```

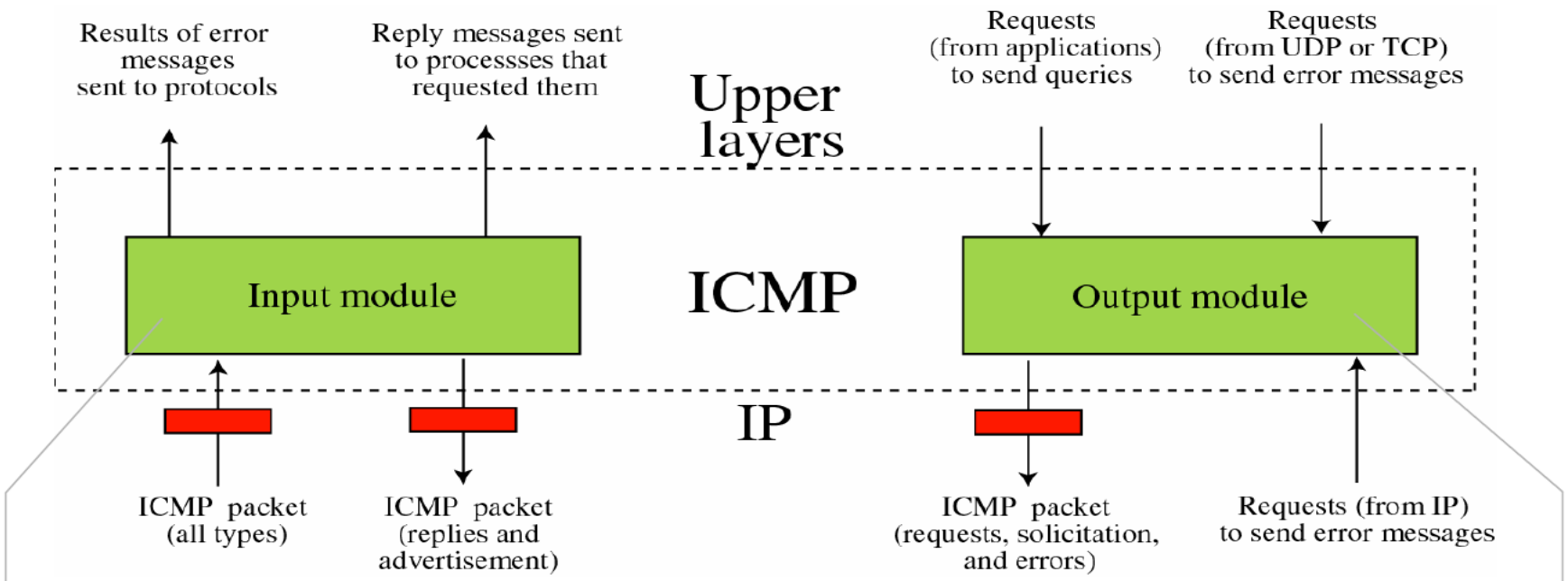
APPLICAZIONI: PING

```
C:\Users\ricci>ping -n 20 unipi.it

Esecuzione di Ping unipi.it [131.114.21.42] con 32 byte di dati:
Risposta da 131.114.21.42: byte=32 durata=97ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=93ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=37ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=46ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=66ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=41ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=94ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=38ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=48ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=58ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=36ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=234ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=37ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=37ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=36ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=37ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=41ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=36ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=38ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=38ms TTL=54

Statistiche Ping per 131.114.21.42:
    Pacchetti: Trasmessi = 20, Ricevuti = 20,
    Persi = 0 (0% persi),
Tempo approssimativo percorsi andata/ritorno in millisecondi:
    Minimo = 36ms, Massimo =  234ms, Medio =  59ms
```

ICMP MESSAGE PROCESSING



Input module handles all received ICMP messages. It is invoked when an ICMP packet gets delivered to it from IP layer.

1) If the received packet is a request or solicitation, the module creates a reply or an advertisement and sends it out.
2) If the received packet is a redirection message the module updates the routing table.
3) If the received packet is an error message, the module informs respective protocol …

Output module is responsible for creating request, solicitation, or error messages requested by higher layers or the IP protocol. The module receives a demand from IP, UDP, TCP, or an application program.

OLTRE DESTINATION BASED FORWARDING

- generalizzare il processo di routing
- non più solo indirizzo mittente/destinatario
- altre considerazioni
  - altre informazioni nel pacchetto IP
  - trattamento diverso di alcuni pacchetti in base al costo pagato per un certo servizio
  - privacy: isolare pacchetti di certi utenti in modo che non transitino in una sottorete
  - scartare traffico proveninete da un insirizzo IP
- passaggio a Software defined networks (SDN)