

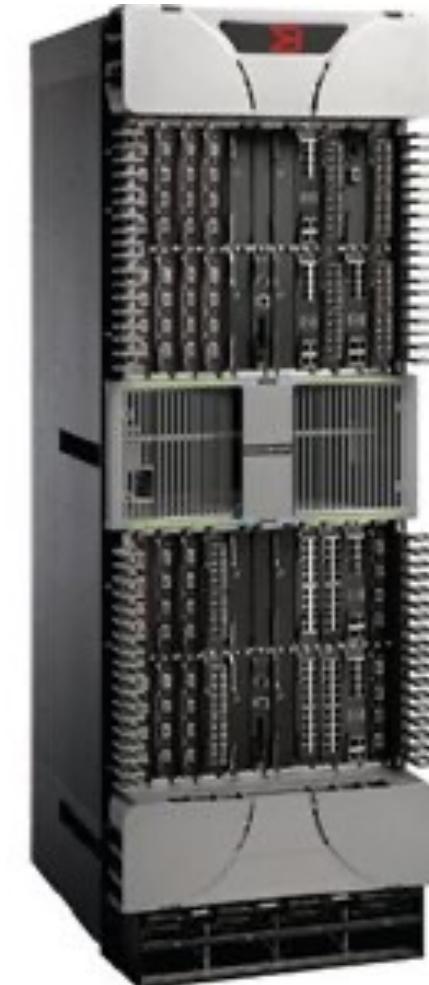
# **Reti di Calcolatori**

Networking - Instradamento



# Router IP

- Un router può interconnettere reti che usano diverse tecnologie, inclusi mezzi fisici diversi, tecniche di accesso, schemi di indirizzamento fisico e formato dei frames.



# Router – Le sue parti

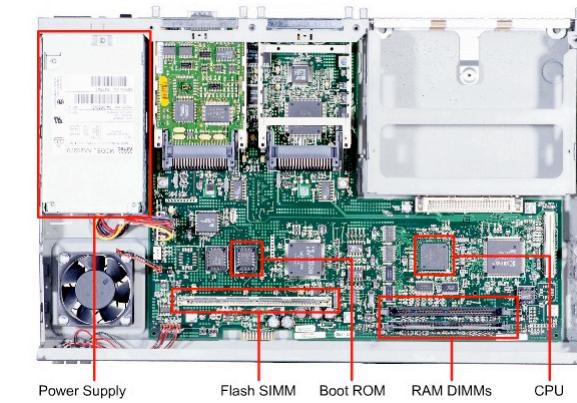
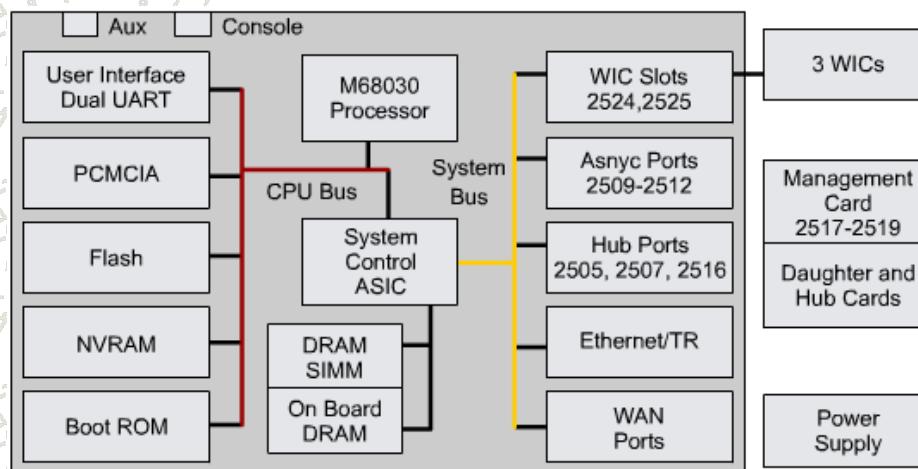
Il router, tipicamente realizzato in architettura special purpose, è configurato o attraverso una porta di console o via telnet ed va inizializzato caricando un sistema operativo (**IOS**) nella sua memoria.

- Porte di Ingresso / Uscita (interfacce Ethernet, SONET, etc..)

- Blocco di commutazione, che collega le porte di ingresso con quelle di uscita

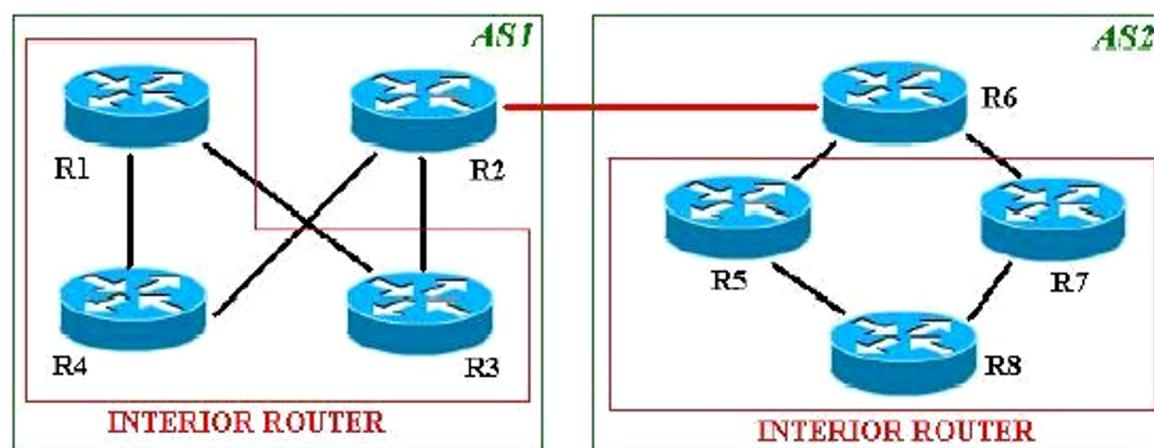
- Processore di instradamento: esegue protocolli di routing, aggiorna tabelle di routing

- Memorie: **ROM** (codice per Bootstrap [avvio] e POST [Power-On Self Test = diagnostica e manutenzione hardware]), **RAM** (tabelle di routing), **NVRAM** (configurazioni di avvio e il registro modalità di avvio), **Flash** (immagine IOS).



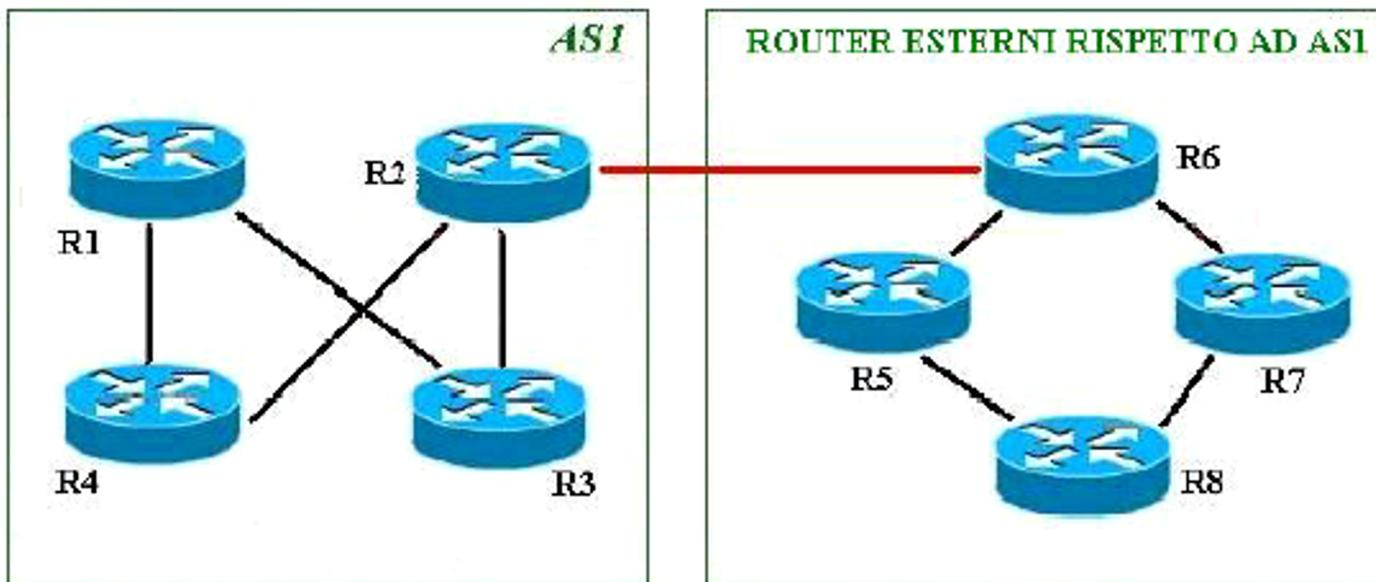
# Collegamenti tra routers (1)

Il collegamento di più reti sotto un unico dominio amministrativo prende il nome di **Autonomous System** (AS). I router che instradano messaggi all'interno dello stesso AS e non hanno diretta connessione con altre reti (network) esterne, sono chiamati **Interior Router** e scambiano informazioni di instradamento tramite un **IGP** (Interior Gateway Protocol).



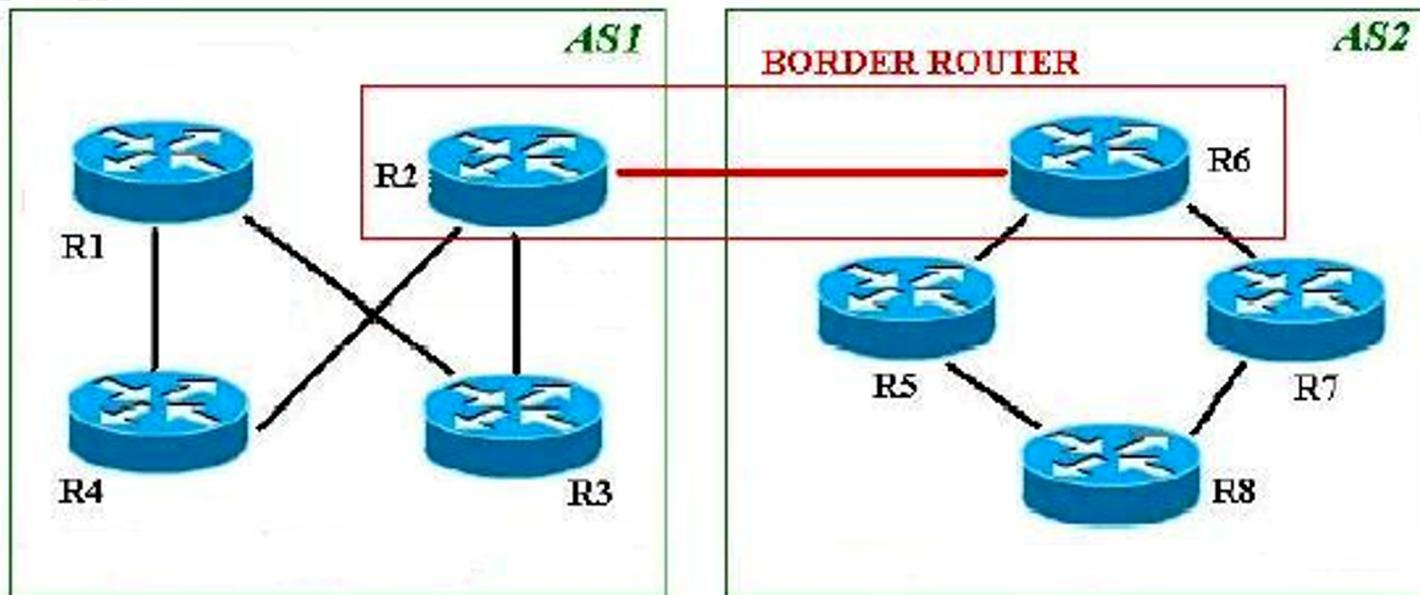
# Collegamenti tra routers (2)

I router, che instradano i messaggi tra AS diversi sono detti **Exterior Router**, scambiano informazioni di instradamento utilizzano un protocollo **EGP** (Exterior Gateway Protocol)



# Collegamenti tra routers (3)

I router che fungono da “ ponte di collegamento ” tra AS diversi cioè sono il punto di ingresso e di uscita verso altri AS vengono detti **Border Router** o router di frontiera.



# Tabelle di routing (1)

Per potere instradare in modo corretto i vari pacchetti che girano nella rete un router ha bisogno di avere alcune informazioni fondamentali:

- Indirizzo IP dell'HOST di destinazione
- L'indirizzo dei router ad esso adiacenti, da cui poter ricavare le informazioni delle varie reti e sottoreti remote raggiungibili
- I possibili percorsi (alternativi) per raggiungere queste reti remote
- Il miglior percorso verso ciascuna delle reti remote non direttamente connesse ad esso

# Tabelle di routing (2)

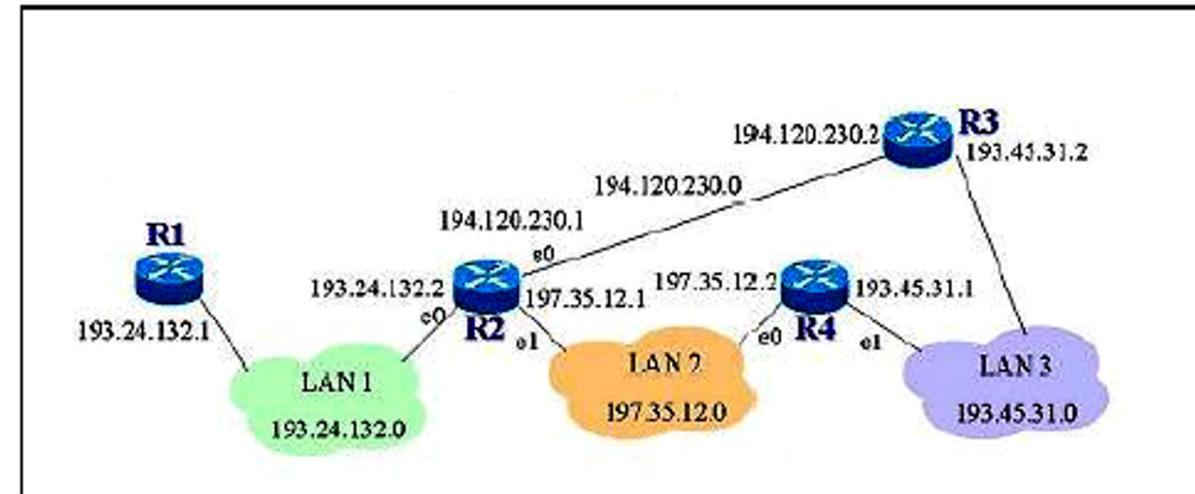
Una tabella di instradamento (**Routing Table**) raccoglie le informazioni necessarie per individuare il percorso ottimale verso tutte le possibili reti.

**TABELLA DI ROUTING**

<b>INDIRIZZO IP DI DESTINAZIONE</b>	E' il campo più importante contenuto nella Routing Table, quando un router riceve un pacchetto dati attraverso la sua porta di IN, controlla nella propria tabella di routing se esiste una entry per tale destinazione, ed in caso affermativo inoltra il flusso dati nella corrispondente porta di OUT.
<b>METRICA</b>	Definisce l'algoritmo di instradamento (Hop Count, Load, Delay, Bandwidth, ecc. )
<b>INDIRIZZO DEL ROUTER DI NEXT HOP</b>	E' l'indirizzo del router successivo per raggiungere la rete di destinazione
<b>INTERFACE</b>	Interfaccia del router attraverso cui deve essere instradato il pacchetto verso il next hop
<b>TIMER</b>	Scandisce temporalmente ogni quanto tempo inviare gli updates ad i router vicini

# Tabelle di routing - Esempio(3)

Per la rete illustrata le tabelle per R2 ed R4 sono:



**TABELLA DI ROUTING DEL ROUTER R2**

NETWORK	INTERFACE	NEXT HOP	METRIC
192.24.132.0	Ethernet 0		0
197.35.12.0	Ethernet 1		0
195.45.31.0	Ethernet 1	197.35.12.2	1
195.45.31.0	Serial 0	194.120.230.2	1

**TABELLA DI ROUTING DEL ROUTER R4**

NETWORK	INTERFACE	NEXT HOP	METRICA
197.35.12.0	Ethernet 0		0
195.45.31.0	Ethernet 1		0
193.24.132.0	Ethernet 0	197.35.12.1	1

# Router di default (default gateway)

- Router verso cui è inviato il traffico diretto ad una destinazione non presente nella tabella di routing
- Non obbligatorio ma molto utilizzato:
  - negli host, che possono anche non avere una tabella di routing propria e che inviano al router di default tutti i datagrammi non diretti alla rete cui sono collegati
  - nei router, che pur avendo tabella di discrete dimensioni non coprono tutte le possibili destinazioni
- Il default gateway è presente all'ultima riga della tabella di instradamento ed è rappresentato con tutti zero sia nel campo Prefisso di Destinazione che nel campo Subnet Mask

Prefisso di Destinazione	Subnet Mask	Next hop	Interfaccia
20.0.0.0	255.0.0.0	d.c. (20.0.0.6)	eth0
130.11.0.0	255.255.0.0	d.c. (130.11.0.6)	eth1
0.0.0.0	0.0.0.0	20.0.0.5	eth0

# Longest Prefix Matching

- Per valutare se un host con indirizzo X appartiene ad una sottorete con indirizzo Y/M si effettua l'operazione di matching, cioè si verifica che:

$$X \text{ and } M = Y \text{ and } M$$

- Quando all'interno di un router si effettua l'operazione di instradamento il matching va effettuato per tutte le righe della tabella di routing: se il matching dà esito positivo per più righe si attua la regola del **Longest Prefix Matching**, si utilizza la riga che ha il maggior numero di bit in comune con X and M

## ➤ Instradamento

- indirizzo 198.15.7.3
- indirizzo 198.15.4.4

➤ **198.15.7.3 -> porta 7**

➤ **198.15.4.4 -> porta 1**

## *Tabella di instradamento*

Prefix	Porta d'uscita
198.15.0.0/16	1
198.15.7.0/24	7

# Operazioni di instradamento

Instradamento effettuato dal router X:

- 1) estraе l'indirizzo IP di destinazione dal datagramma (Y)
- 2) se l'indirizzo di destinazione Y coincide con X (la destinazione è il router X), estraе il contenuto informativo e lo consegna al protocollo indicato
- 3) decrementa il Time To Live del datagramma; se il Time To Live è arrivato a zero scarta il datagramma e ne da comunicazione all'host mittente
- 4) confronta la componente  $X \text{ and } M$  con  $Y \text{ and } M$  ( $M$  maschera di rete del router X), se sono uguali inoltra il datagramma direttamente ( $\rightarrow$  ARP)
- 5) Per tutte le righe della tabella di instradamento  $[N, M, NH, I]$  confronta se  $Y \text{ and } M = N \text{ and } M$  ( $N$  indirizzo della sotto-rete,  $M$  maschera di sotto-rete,  $NH$  ind. prossimo router ed  $I$  interfaccia d'uscita). Inoltra il datagramma verso il router  $NH$  relativo alla corrispondenza con la maschera più lunga (Longest Prefix Match)
- 6) altrimenti (non c'è nessuna corrispondenza) inoltra verso il default router se ne è stato specificato uno oppure scarta il datagramma e dichiara errore di instradamento

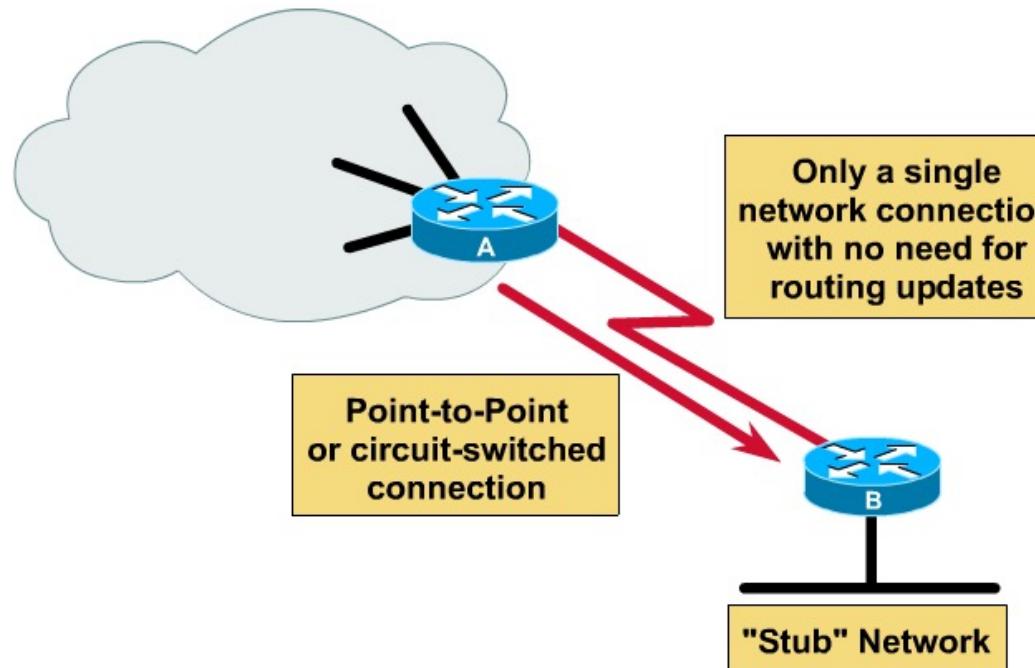
# Strategie di Instradamento

Nel configurare le politiche di instradamento su una rete abbiamo essenzialmente due opzioni:

- **Routing Statico**: Prevede il calcolo dei percorsi offline quando la rete non è ancora attiva e la loro configurazione manuale, a carico dell'operatore.
- **Routing Dinamico**: I percorsi cambiano dinamicamente in base alle situazioni di traffico ed ad altre informazioni locali come congestione, guasti, ecc.

# Quando usare route statiche

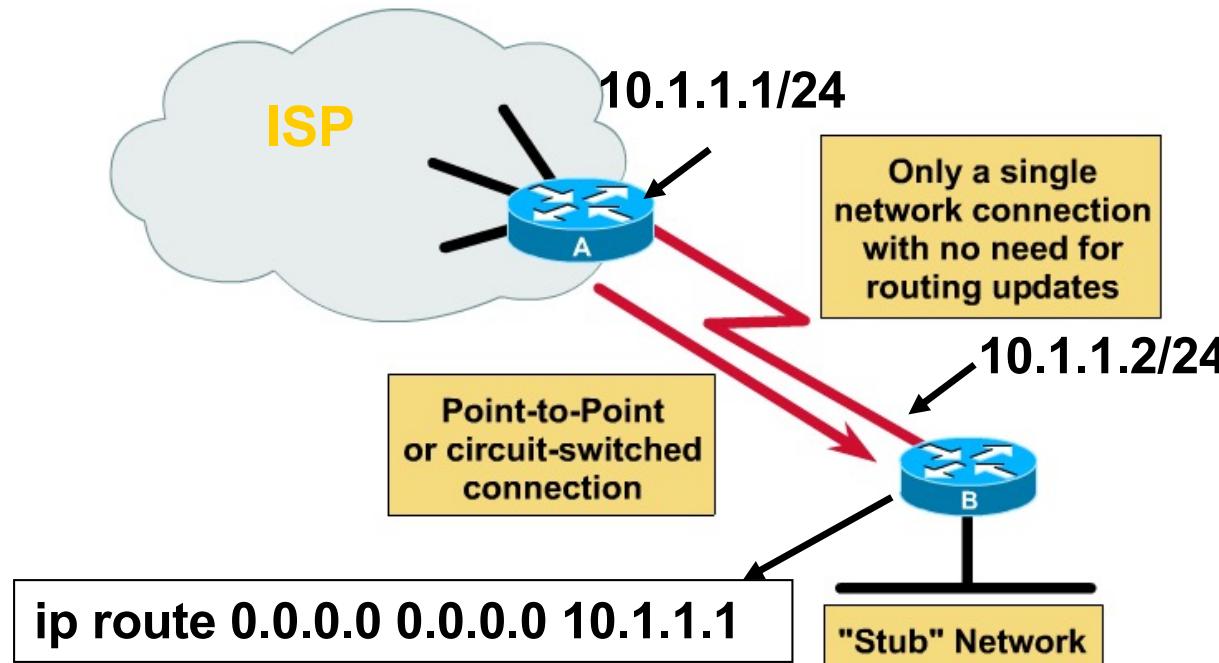
## Static Routing Example



- Il routing statico e' virtualmente ingestibile in condizioni di reti anche banalmente complesse, ma viene utilizzato in circostanze particolari:
  - un router ha una unica connessione
  - un router ha piu' di una connessione, ma per tutte le connessioni tranne una c'e' una destinazione ben precisa che non puo' cambiare per motivi di topologia, e l'ultima connessione deve essere utilizzata per tutte le altre destinazioni

# La route di default

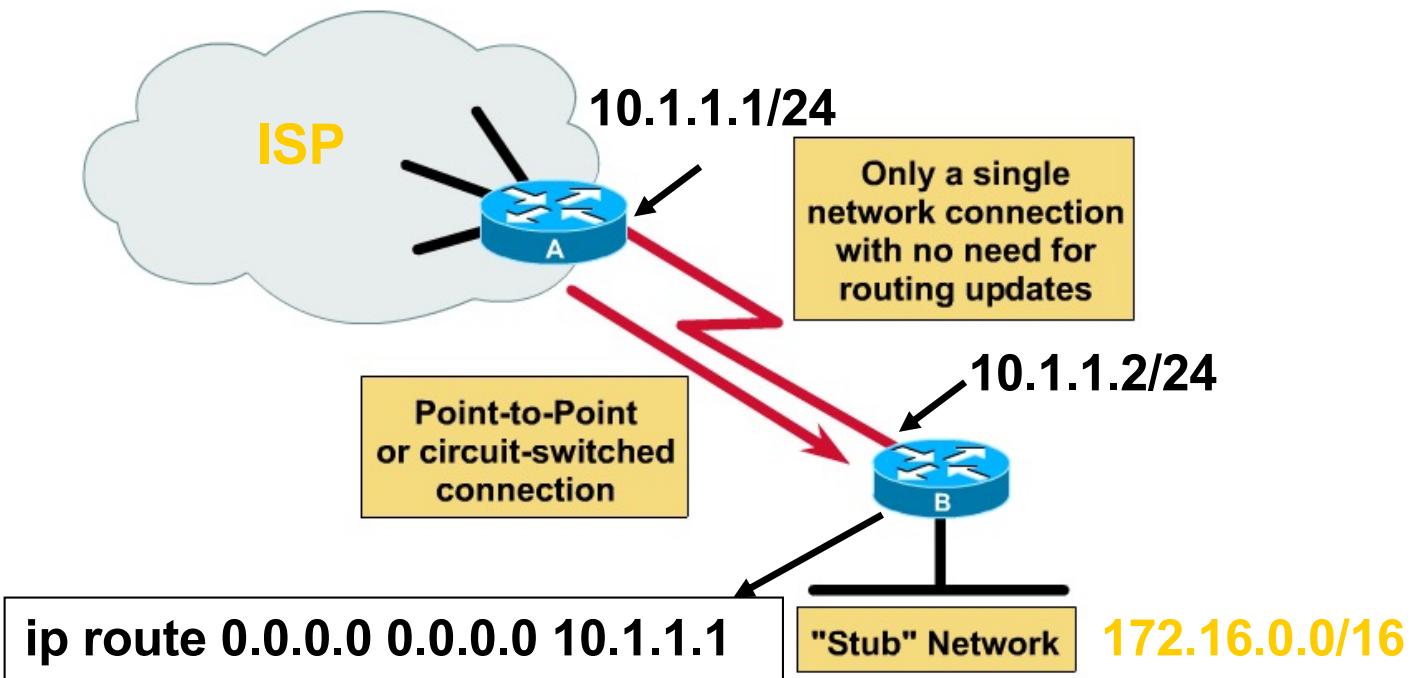
## Default Static Routing Example



- La route di default specifica l' instradamento di tutti i pacchetti per i quali la tabella di instradamento non fornisce una route esplicita.
  - Si specificano rete e mask con “0.0.0.0 0.0.0.0” oppure “0.0.0.0/0”.
  - La rete 0.0.0.0/0 risulta sempre ultima nel matching (longest prefix match) ma corrisponde sempre

# La route di default (2)

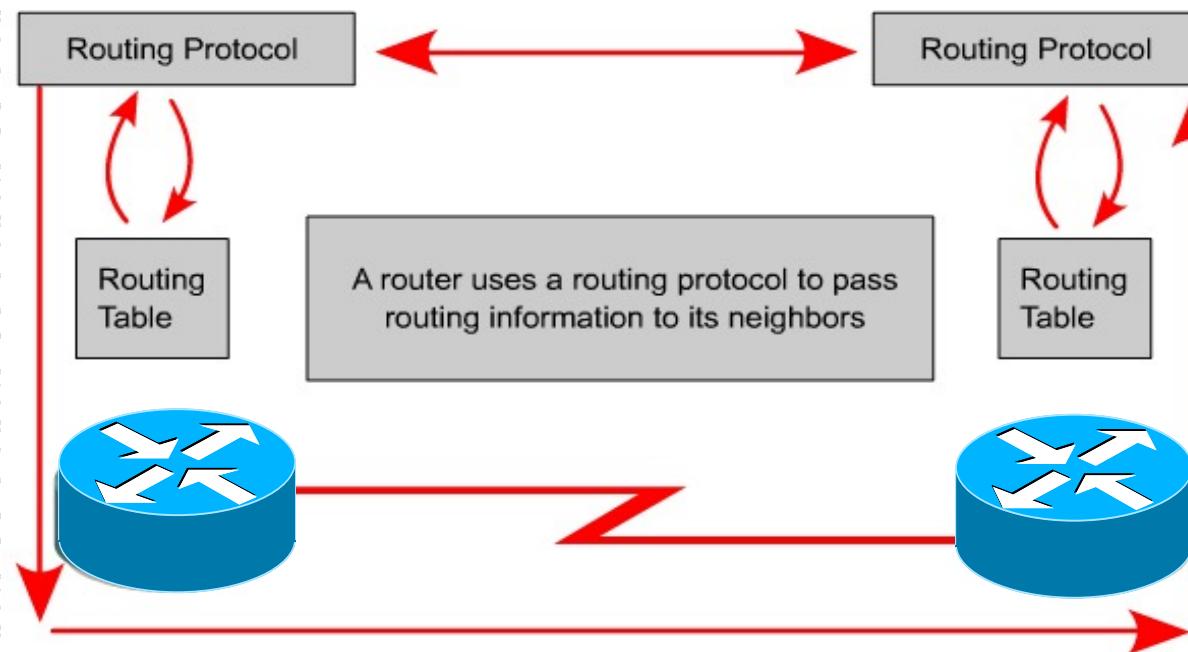
## Default Static Routing Example



```
RTB#show ip route
Gateway of last resort is 10.1.1.1 to network 0.0.0.0
C    172.16.0.0/16 is directly connected, Ethernet0
      10.0.0.0/24 is subnetted, 1 subnets
C      10.1.1.0 is directly connected, Serial1
S*   0.0.0.0/0 [1/0] via 10.1.1.1
```

# Routing dinamico - Protocolli di routing

- Sono protocolli utilizzati dai routers per costruire le tabelle che contengono le informazioni di instradamento dei pacchetti.
- Per costruire la tabella, ciascun router dovrà scambiare pacchetti informativi con i routers ad esso collegati.

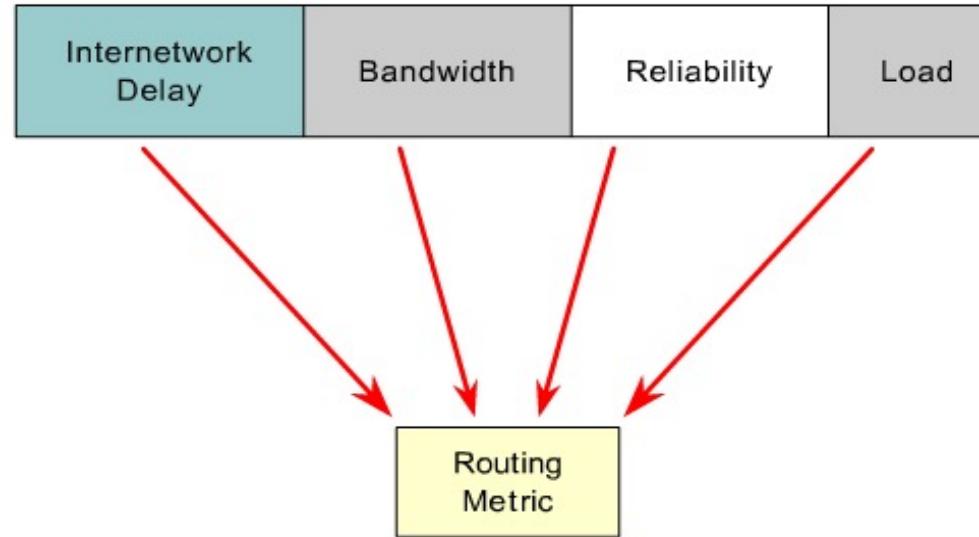




# Metriche

- Interviene nella caratterizzazione di un percorso per l' instradamento di pacchetti tra due nodi.
- La Metrica serve per selezionare il percorso “migliore”
  - Percorso più corto
  - Percorso meno congestionato
  - Percorso più ampio
  - Percorso meno costoso
  - ...
- Metriche dinamiche: Come aggiornare le decisioni in presenza di variazioni
  - Guasti
  - Modifiche ai parametri
  - Variazioni nel carico dei link

# Metriche



- RIP – Hop Count
- IGRP and EIGRP – Bandwidth, Delay, Reliability, Load
- OSPF – Bandwidth
- IS-IS – Cost
- BGP – Number of AS or policy

# Distanza amministrativa

- La distanza amministrativa quantifica l'attendibilità dell'informazione di instradamento
  - Più il valore è basso, più l'informazione è “sicura”
  - Se il router riceve da fonti diverse route alternative verso la stessa rete, userà la distanza amministrativa per decidere quale rendere attiva

Route Source	Administrative Distance	Default Metric(s)
Connected	0	0
Static	1	0
EIGRP Summary Route	5	0
External BGP	20	Bandwidth,Delay
Internal EIGRP	90	Link cost (bandwidth)
IGRP	100	Link cost (bandwidth)
OSPF	110	Hop count
IS-IS	115	Value assigned by admin
RIP	120	
External EIGRP	170	
Internal BGP	200	

# Classificazione Algoritmi di routing

- **Non adattivi**: Utilizzano criteri fissi di instradamento, sono statici e deterministici. Appartengono al gruppo il Fixed Directory Routing ed il Flooding.
- **Adattivi**: Calcolano ed aggiornano le tabelle di instradamento in funzione della topologia della rete e dello stato dei link, sono dinamici e non deterministici. Il Routing Centralizzato, il Routing Isolato ed il Routing Distribuito appartengono a questo gruppo.



# Algoritmi di routing non adattivi

- **Fixed Directory Routing.** L'algoritmo prevede che ogni nodo abbia una tabella di instradamento che metta in corrispondenza il nodo da raggiungere con la linea da usare. Queste entry sono puramente statiche, poiché è il gestore che si occupa di determinarle e di configurare il router. Il gestore ha così il completo controllo sul traffico ed è necessario un suo intervento in caso di guasto.
- **Basati su Flooding.** Ciascun pacchetto che arriva ad un router viene instradato su tutte le porte, eccetto quella da cui è arrivata. Questo metodo, concepito per reti militari, massimizza la probabilità che i dati arrivino a destinazione, ma produce un altro volume di traffico sulla rete.



# Algoritmi di routing adattivi

- **Routing Centralizzato.** Un RCC, Routing Control Center, conosce la topologia di tutta la rete, calcola e distribuisce le tabelle di instradamento di ogni router. La gestione permette tabelle calcolate con algoritmi sofisticati, ma necessita un gestore a livello mondiale.
- **Routing Isolato.** Ogni router calcola le proprie tabelle di instradamento in modo indipendente.
- **Routing Distribuito.** E' fusione dei due metodi precedenti, realizza le funzionalità RCC in ogni singolo nodo della rete. Le tabelle vengono aggiornate dai routers scambiandosi informazioni di servizio mediante apposito protocollo.
- Gli algoritmi sono il Distance Vector ed il Link State

# Routing e Grafo di rete

Il concetto è quello di costruire un grafo della rete, dove ogni nodo del grafo rappresenta un router ed ogni arco del grafo rappresenta una linea di comunicazione (chiamata anche **canale**). Per scegliere un percorso tra due router, l'algoritmo cerca nel grafo il cammino più breve tra di essi.

## Metriche possibili

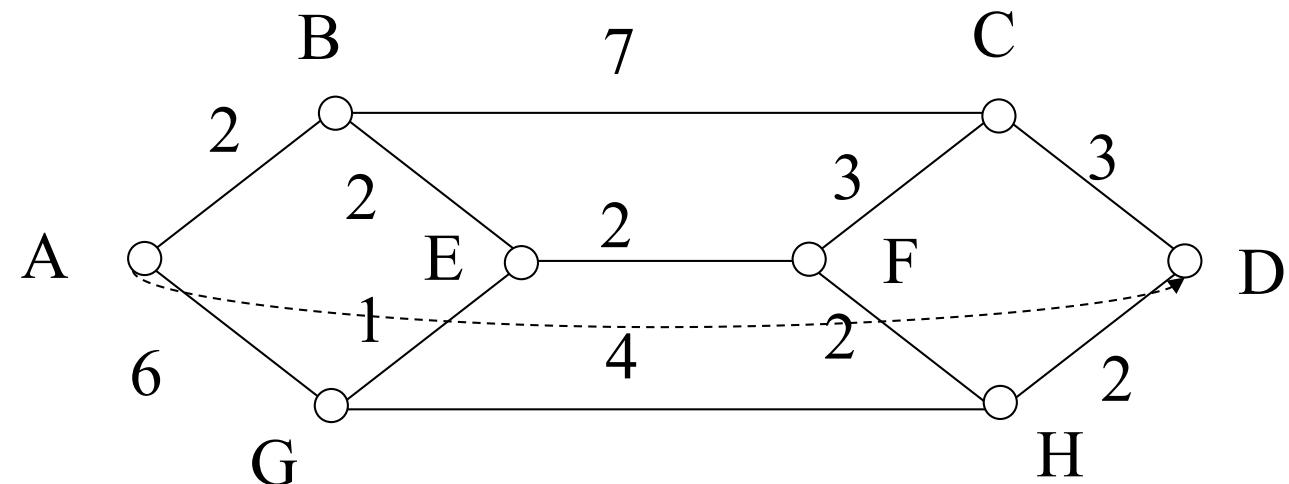
- distanza geografica
- costi
- capacità

# Algoritmi basati sul percorso piu' breve

- Per scegliere il percorso ottimale la soluzione e' trovare il percorso **piu' breve**, in base ad una certa metrica
- Possono essere scelte **differenti metriche**:
  - distanza geografica
  - numero di salti
  - costo delle linee
  - ritardo di accodamento
  - distanza chilometrica
  - larghezza di banda
- o una **funzione** di alcune di queste
- Definita una metrica, le linee potranno essere **etichettate** con un numero (peso o metrica): piu' basso e' il numero, "**piu' breve**" e' la linea (quindi preferibile)
- La distanza di un percorso e' dato dalla somma delle distanze dei singoli salti
- Dijkstra ha ideato un algoritmo (**Shorted Path First**) che determina, in base alla topologia ed ai pesi, il cammino piu' breve tra due nodi del grafo

# Algoritmo di Dijkstra

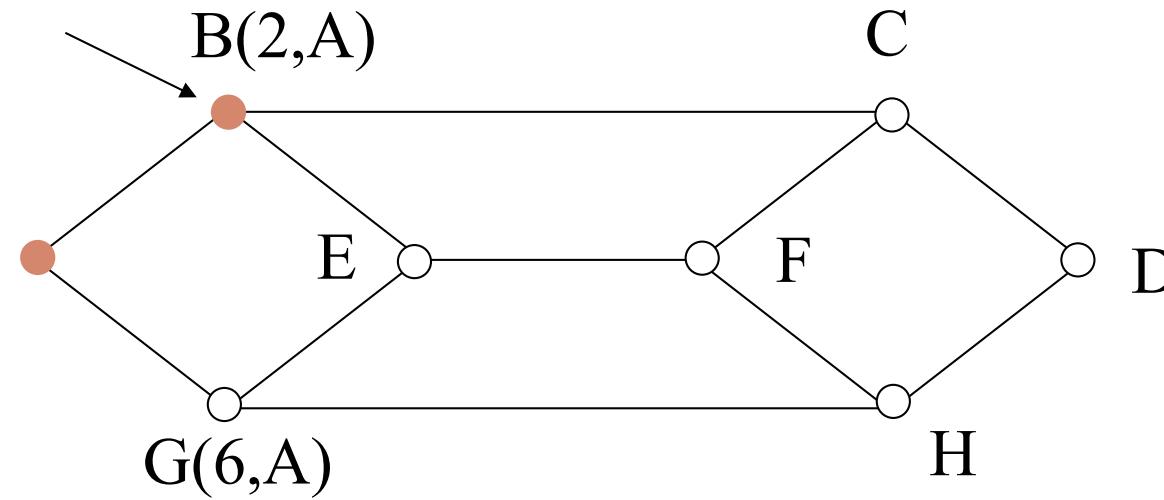
- L'algoritmo di Dijkstra (1959) lavora su grafi orientati, che hanno pesi non negativi sui collegamenti. Questo algoritmo trova i percorsi più brevi tra un nodo di partenza e tutti gli altri.
  - E' utilizzato per calcolare il percorso a costo minimo
  - Obiettivo: trovare il percorso piu' breve tra le coppie di nodi
- Esempio



# Procedimento

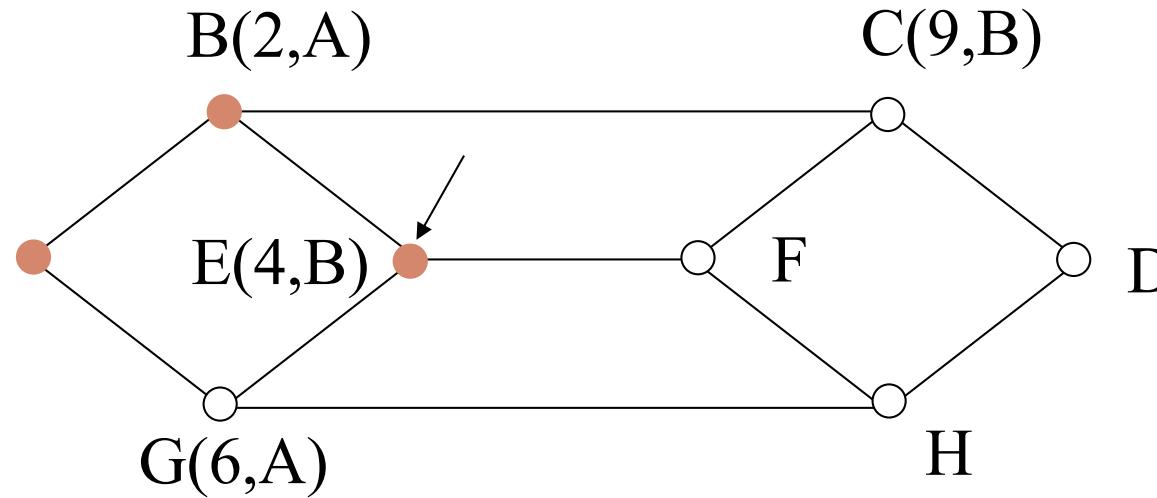
- I nodi vengono messi in due insiemi: **esaminati** e **non esaminati**
- Ad ogni nodo  $i$ , devono essere associate due etichette, COSTO[i] che indica il peso totale del cammino (la somma dei pesi sugli archi percorsi per arrivare al nodo  $i$ -esimo) e PRED[i] che indica il nodo che precede  $i$  nel cammino minimo.
  - Le etichette possono essere provvisorie o permanenti
- Il nodo sorgente è il primo nodo attivo
- Si esaminano tutti i nodi  $i$  connessi al nodo attivo  $j$  e per ciascuno di essi si pone:
  - $PRED[i] = j$
  - $COSTO[i] = COSTO[j] + LINK[j,i]$
- Il nuovo nodo attivo è quello con il costo più basso tra quelli esaminati e viene marcato come permanente
  - L'etichetta di un nodo permanente non viene più cambiata

# Ricerca del percorso A-D (1)



- Esamino ogni nodo adiacente ad A etichettandolo con la sua distanza da A
- Da A trovo il nodo adiacente più vicino e lo marco con costo e provenienza
- La provenienza permetterà di ricostruire il cammino finale

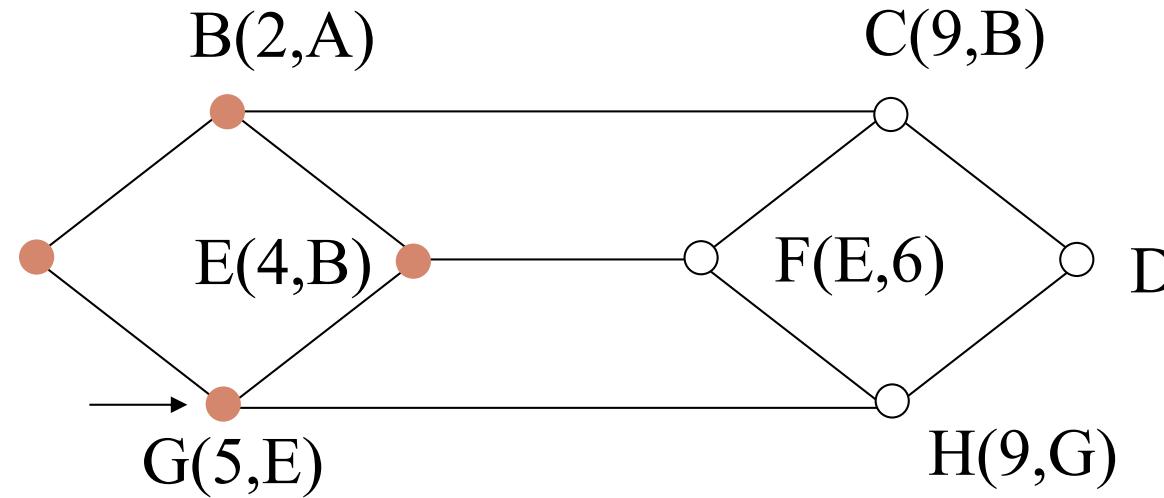
# Ricerca del percorso A-D (2)



Da B si procede nello stesso modo esaminando i vicini E e C

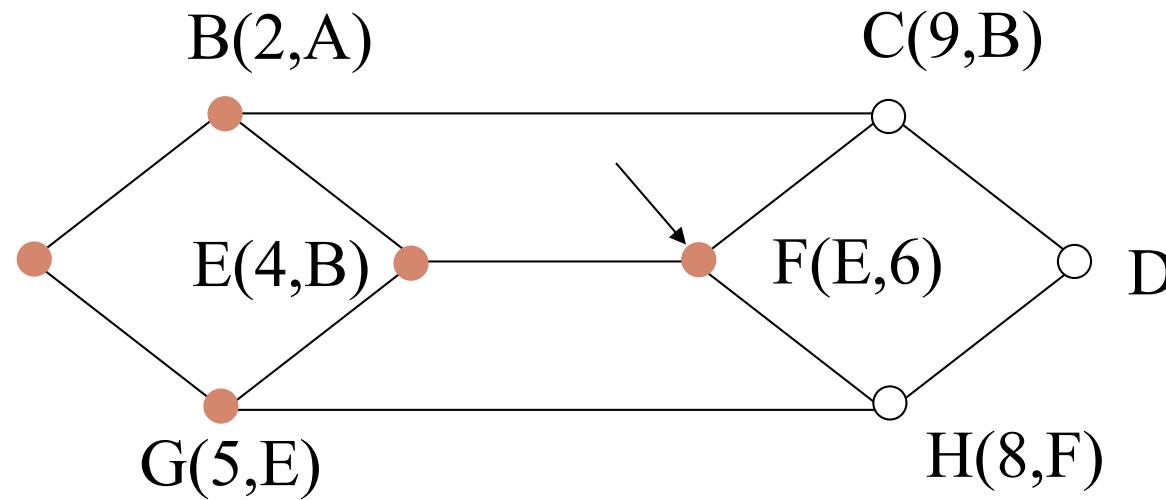
- se un nuovo nodo raggiungibile da B ha costo da A inferiore al costo del percorso che passa per B diventa il nuovo nodo attivo  
(appartiene al percorso minimo corrente)

# Ricerca del percorso A-D (3)



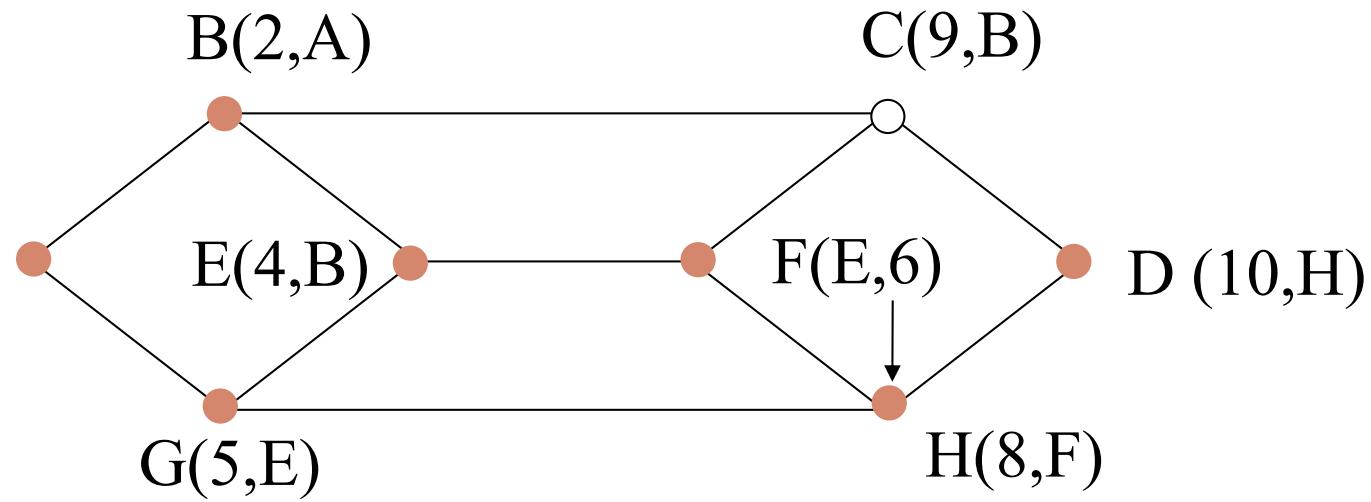
- Da E: si trova un percorso a costo inferiore che passa per G (rispetto a quello che passa per F) e quindi G diventa il nuovo nodo attivo
- Partendo da G il prossimo nodo sarebbe solo H a distanza 9

# Ricerca del percorso A-D (4)



Viceversa, ripartendo da F, H risulta la scelta a costo inferiore (distanza 8)

# Ricerca del percorso A-D (5)



Il percorso si costruisce individuando per ciascun nodo del percorso minimo il predecessore.

Risulta : A-B-E-F-H-D



# Perche' funziona?

- Supponiamo di aver trovato ABE come percorso minimo
- Se esistesse un altro percorso AXE a costo piu' basso ci sarebbero 2 possibilita'
  - Z e' un nodo permanente
    - E e' gia' stato esaminato e AXE e' gia' stato individuato
  - Z e' un nodo provvisorio
    - O la sua label e' > di quella di E e quindi non rappresnta il percorso minimo
    - O la sua label e' < di quella di E e quindi Z sarebbe il nodo attivo
- Le precedenti considerazioni portano a concludere che la procedura individua il percorso a costo minimo
- La complessità è  $O(|V|^2)$

# Distance vector (Bellman-Ford)

- Algoritmo **adattivo** sviluppato attorno al 1960, utilizzato per molto tempo
  - Arpanet lo ha utilizzato fino al 1979 nella implementazione chiamata **RIP** (ancora in uso adesso in realtà di **piccole dimensioni**)
- L'idea è quella di partire dal nodo sorgente e cominciare a guardare i nodi adiacenti assegnando loro il valore del costo per raggiungerli (determinato dal costo dell'arco + il valore del nodo da cui si è partiti. Si itera il ragionamento per ciascuno dei nodi raggiunti.
- Se il grafo ha  $|V|$  nodi dopo  $|V|-1$  iterazioni tutti i nodi avranno assegnato il costo minimo per essere raggiunti dal nodo sorgente.
- Nella sua struttura base è molto simile a quello di Dijkstra, ma invece di selezionare il nodo di peso minimo, tra quelli non ancora processati, con tecnica greedy, semplicemente processa tutti gli archi e lo fa  $|V|-1$  volte
- Ha una complessità temporale  $O(|V| |E|)$ ,  $|E|$  numero di archi del grafo

# Distance vector (Bellman-Ford)

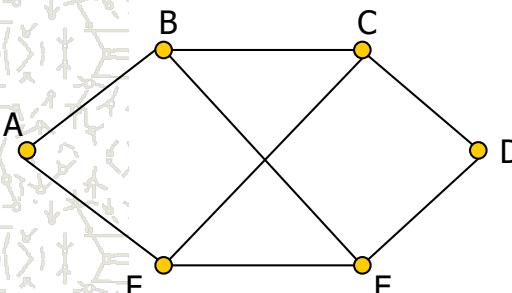
- Ad ogni linea e' assegnata una **distanza**, valutata in base ad una metrica, indicata anche col nome di costo
- Le informazioni di routing vengono **scambiate** con tutti i router **adiacenti**
- Ogni nodo acquisisce una visibilità della rete **indiretta**, mutuata dai **vicini**
- In base alle informazioni ricevute dai vicini viene **ricostruita** la tabella di routing nuova
- Il valore della distanza avra' un certo **limite massimo**, indicante che la destinazione **non e' raggiungibile**
  - nel caso di metrica ad hop, la distanza di infinito sara' pari al **numero massimo** di hop possibili nella rete **piu' uno**
  - in caso di metrica secondo i **ritardi** puo' essere piu' complesso determinare l'irraggiungibilita' della destinazione: si deve valutare a priori un valore **ragionevolmente elevato ma non troppo**

# Distance vector (Bellman-Ford)

- Valutazione delle distanze:
  - il router deve **conoscere** i router adiacenti, ed il **costo** delle linee che li connettono direttamente
    - per fare questo un router **scambia** a tempi definiti con i propri vicini dei pacchetti per essere **aggiornati** sulla loro presenza
    - se la metrica scelta dipende dal ritardo della linea, questi pacchetti sono utilizzati per aggiornare la **valutazione del costo** della linea diretta verso il router adiacente
  - Quando viene **ricevuta** la tabella di routing dai vicini, per ogni destinazione si valuta la distanza **aggiungendo** alla distanza riportata dal router adiacente quella relativa alla linea che li connette
  - Tra tutte le distanze relative alla stessa destinazione riportate dai router adiacenti, e rivalutate in base alla distanza verso ciascun router adiacente, si sceglie quella con **distanza inferiore**

# Algoritmi di Routing Distance Vector

In dettaglio, questo tipo di algoritmi funziona mantenendo una tabella contenente la più piccola distanza conosciuta per ogni destinazione e quale canale utilizzare per raggiungerla. Queste tabelle sono aggiornate scambiando informazioni con i propri vicini.



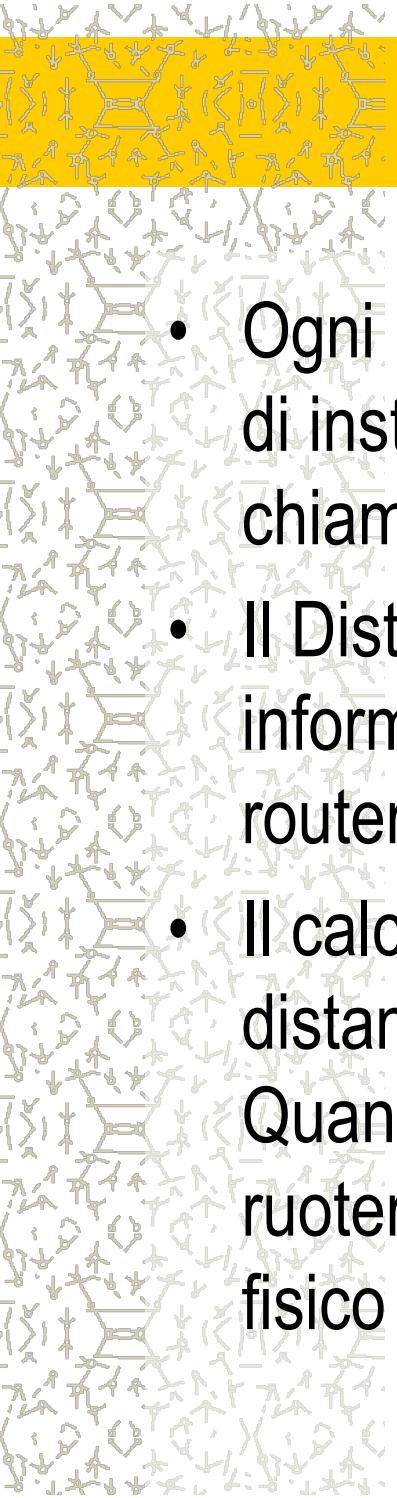
Valori misurati: B => 1 F => 2

B	A	2
F	A	3
	B	0
	C	4
	D	8
	E	3
	F	9

F	A	3
	B	9
	C	4
	D	6
	E	2
	F	0

=>

A	A	0	-
	B	1	B
	C	5	B
	D	8	F
	E	4	B
	F	2	F



# Algoritmi Distance Vector (1)

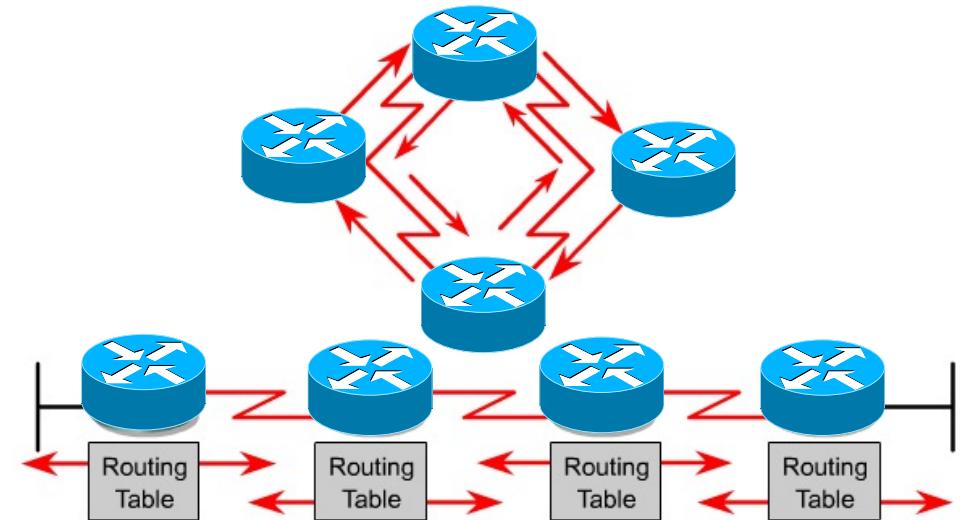
- Ogni router mantiene in memoria, oltre alla propria tabella di instradamento, una struttura dati per ogni linea chiamata **Distance Vector**.
- Il Distance Vector associato a ciascuna linea, contiene informazioni ricavate dalla tabella di instradamento del router collegato all'altro estremo della linea.
- Il calcolo delle tabelle di instradamento dipende da tutti i distance vector associati alle linee attive del router.

Quando un router calcola una nuova tabella, la invia ai routers adiacenti (cioè quelli collegati da un cammino fisico diretto) sotto forma di distance vector.

# Algoritmi Distance Vector (2)

- Ciascun router riceve la tabella di instradamento dai suoi vicini direttamente connessi
- Ogni entry è composta da quattro parametri, indirizzo, hops, costo e linea, e la tabella contiene entry relative ad ogni nodo presente in rete.
- Il Distance Vector da inviare al router adiacente è composto dalle prime tre colonne.

Indirizzo	Hops	Costo	Linea
1	3	25	3
2	5	35	2
3	9	50	6
4	1	5	7
5	0	0	0



Pass periodic copies of a routing table to neighbor routers and accumulate distance vectors.

# Algoritmi Distance Vector (3)

- Il router che lo riceve verifica eventuali modifiche dal precedente e affermativamente aggiorna campi, hops e costo della propria tabella, fondendo (merge) tutti i Distance Vector pervenuti da ogni linea attiva. La fusione avviene selezionando tra le entry di uguale indirizzo quella a minor costo. A parità di costo è scelta quella con minor numero di hops.
- Il Distance Vector è un algoritmo di facile implementazione, ma è sconsigliato per reti di vaste dimensioni (mille nodi) perchè lento a convergere (dipende dalla velocità del router più lento della rete).
- Reagisce bene ai miglioramenti del costo ma molto lentamente ai peggioramenti
- Converge alla velocità degli elementi più lenti

# Flooding

Ogni pacchetto in arrivo viene inoltrato su ogni linea in uscita eccetto quella da cui è arrivato. Per prevenire la duplicazione eccessiva dei pacchetti:

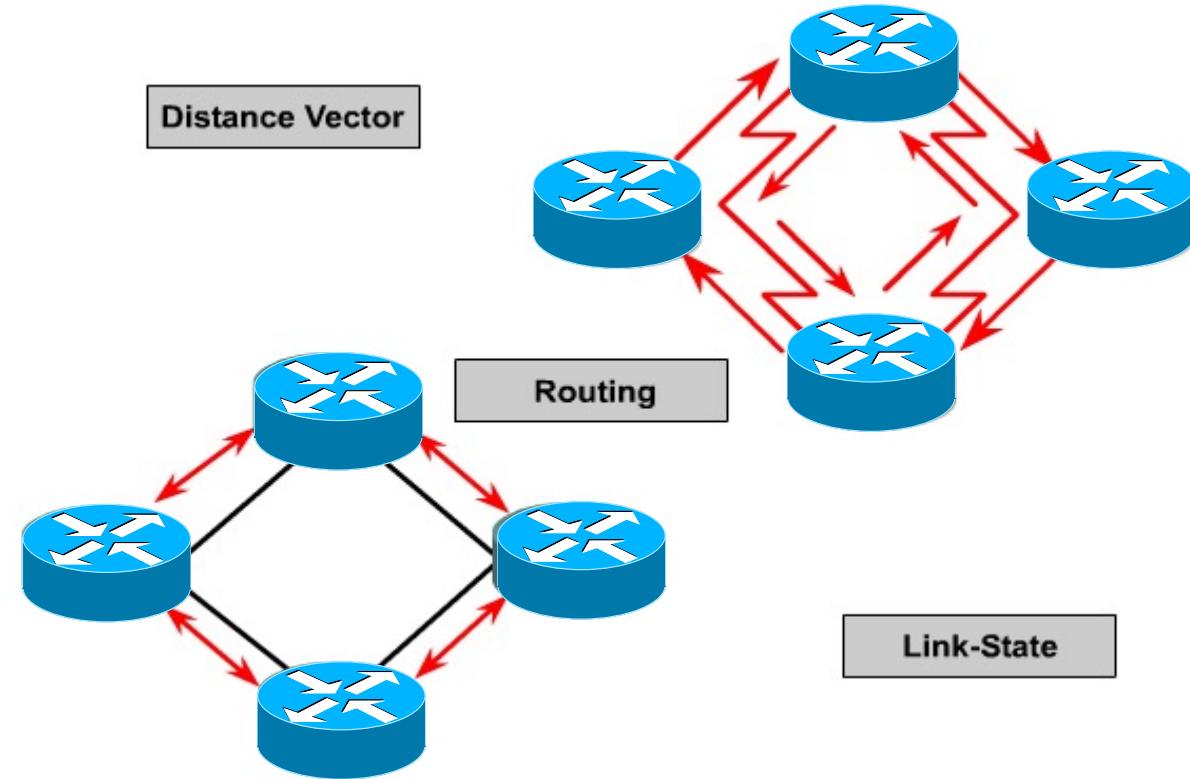
- Questi vengono dotati di un contatore. Quando questo contatore raggiunge lo 0, il pacchetto viene eliminato.
- I router tengono traccia dei messaggi ricevuti e ritrasmessi, e non duplicano messaggi già replicati.
- Nel flooding selettivo invece, i pacchetti in arrivo vengono replicati ma solo sulle linee che approssimativamente vanno nella direzione richiesta dalla sorgente.

Gli algoritmi di flooding vengono utilizzati come benchmark, perché scelgono sempre il cammino più breve, in quanto lo ricercano in Parallello.

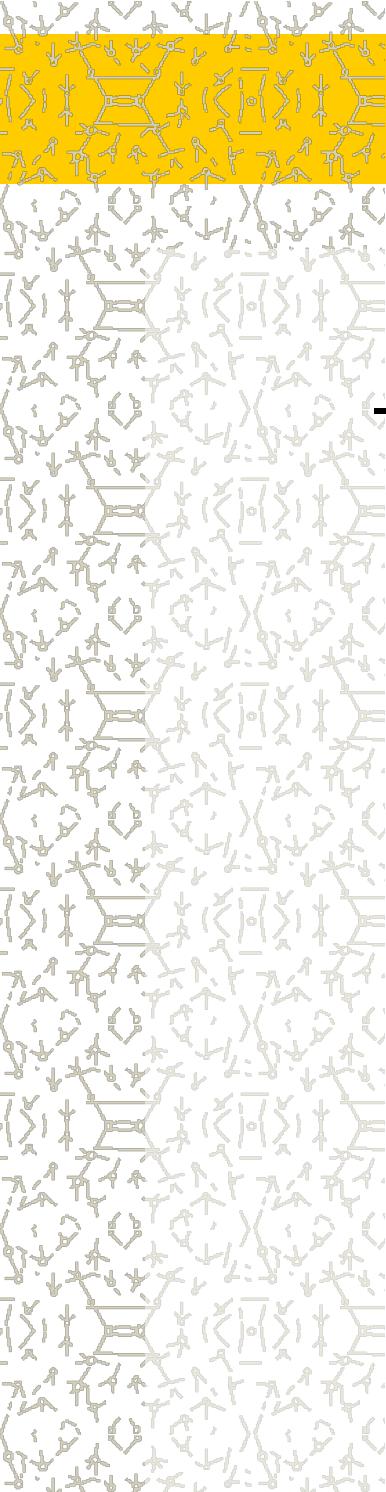
# Algoritmo flooding (cont.)

- Gli aspetti negativi di questo algoritmo sono essenzialmente legati alla **inefficienza**
  - ogni pacchetto va a finire su **tutte le linee della rete**, provocando un utilizzo inefficiente della rete stessa
  - ogni pacchetto va a finire su **tutti i router**, aumentando il carico di lavoro dei router stessi (CPU, occupazione di buffer)
- Aspetti positivi sono:
  - qualsiasi pacchetto arriverà nel tempo **piu' breve** possibile (segue tutte le strade, anche la piu' veloce)
  - estremamente resistente a **modifiche** della topologia
    - anche il malfunzionamento di grandi porzioni della rete permette il **recapito** del pacchetto se **almeno un cammino** rimane operativo
  - non richiede una **conoscenza** a priori della topologia della rete
- Scarsamente utilizzato per via della inefficienza, le caratteristiche di estrema affidabilità di questo protocollo sono sfruttate in diverse circostanze particolari, ad esempio in campo militare

# Classificazione dei protocolli di routing

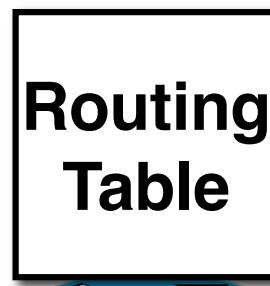


- **Distance Vector:** RIP, IGRP
- **Link State:** OSPF, IS-IS

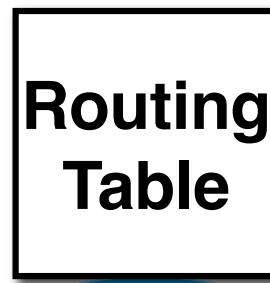
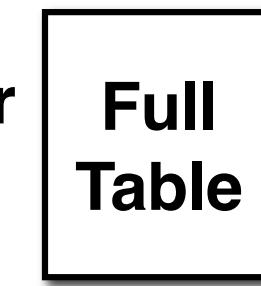


# Aggiornamenti

- Modi differenti di propagare gli aggiornamenti



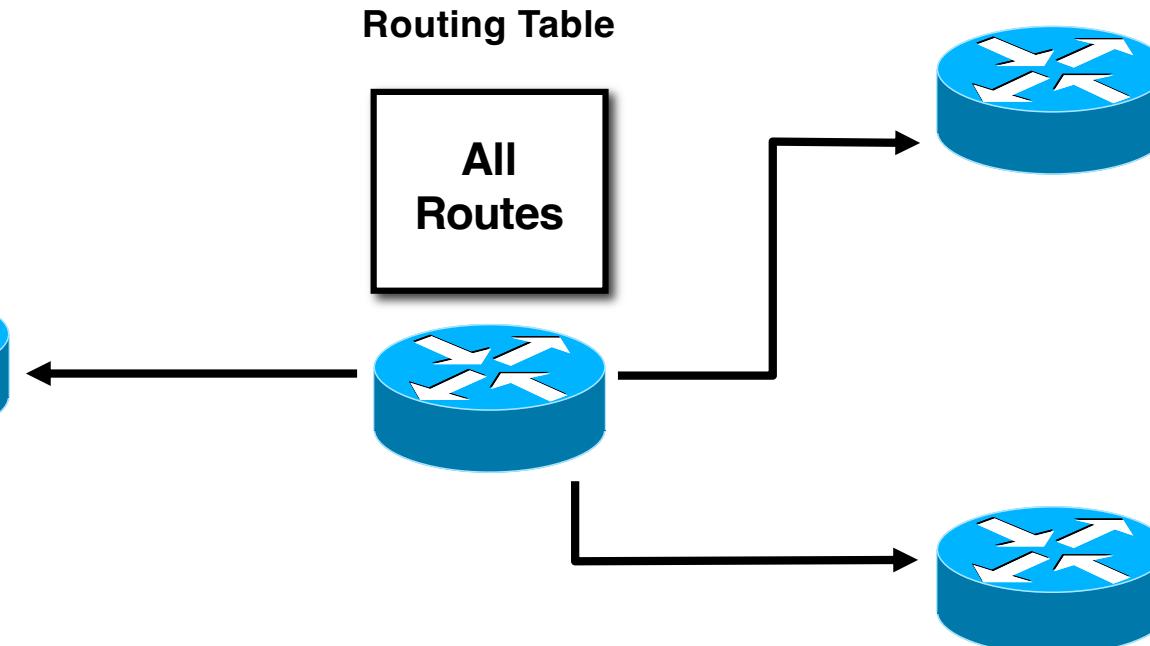
**Distance Vector  
Approach**



**Link-State  
Approach**

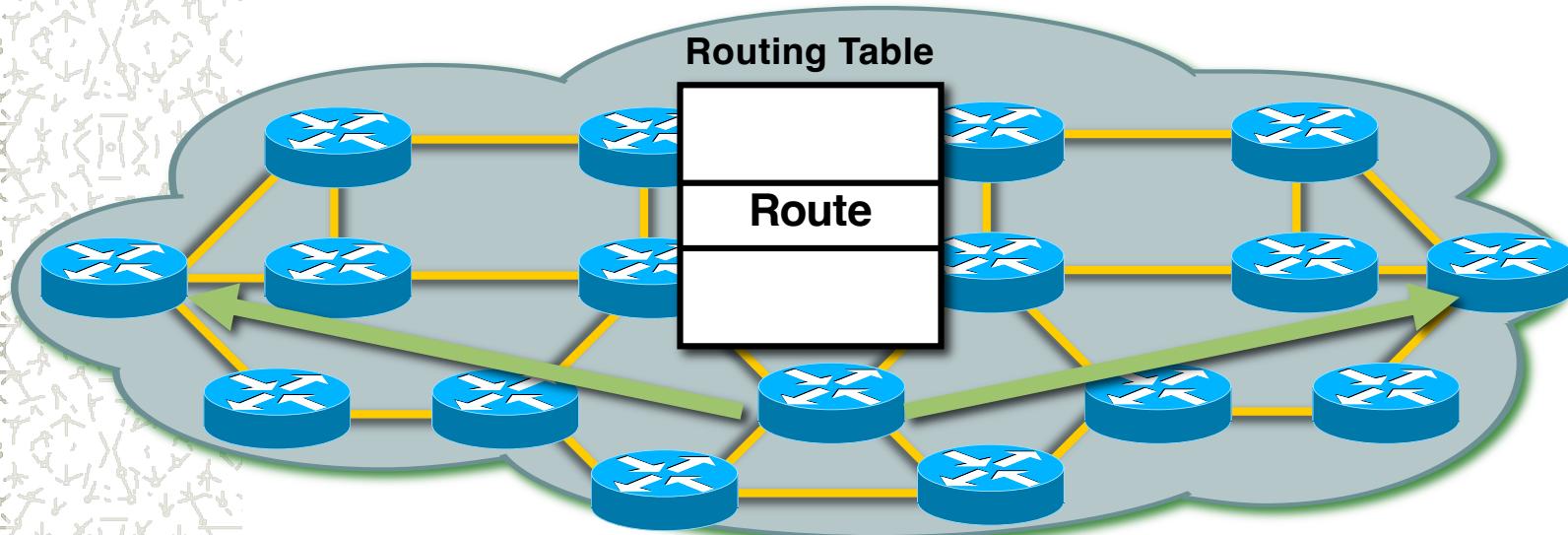


# Aggiornamento in Distance Vector



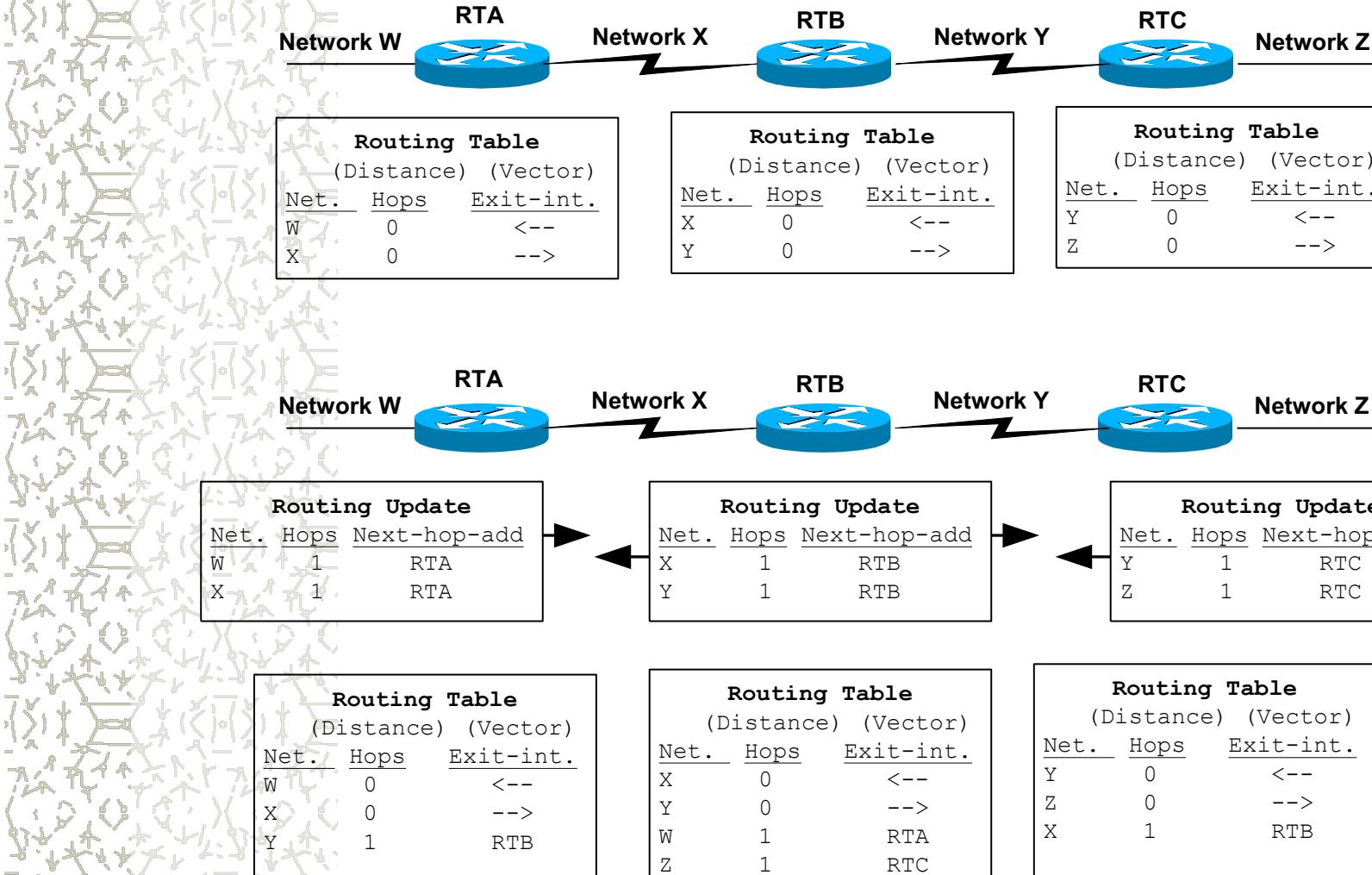
- In una logica distance vector gli aggiornamenti presenti nell'intera routing table sono inviati solo ai vicini direttamente connessi
- Gli aggiornamenti avvengono
  - Al cambiare della topologia
  - Periodicamente

# Aggiornamenti Link-State



- In una logica link state gli aggiornamenti sono propagati uno ad uno a tutti i nodi nel dominio di routing attraverso il flooding
  - Tecniche di design gerarchico possono limitare la dimensione del bacino di diffusione degli annunci

# Distance Vector Network Discovery



# Distance Vector Network Discovery

Existing  
Routing  
Tables

New  
Routing  
Tables



Routing Table (Distance) (Vector)		
Net.	Hops	Exit-int.
W	0	<--
X	0	-->
Y	1	RTB

Routing Table (Distance) (Vector)		
Net.	Hops	Exit-int.
X	0	<--
Y	0	-->
W	1	RTA
Z	1	RTC

Routing Table (Distance) (Vector)		
Net.	Hops	Exit-int.
Y	0	<--
Z	0	-->
X	1	RTB

Routing Update		
Net.	Hops	Next-hop-add
W	1	RTA
X	1	RTA
Y	2	RTA

Routing Update		
Net.	Hops	Next-hop-add
X	1	RTB
Y	1	RTB
W	2	RTB
Z	2	RTB

Routing Update		
Net.	Hops	Next-hop-add
Y	1	RTC
Z	1	RTC
X	2	RTC

Routing Table (Distance) (Vector)		
Net.	Hops	Exit-int.
W	0	<--
X	0	-->
Y	1	RTB
Z	2	RTB

Routing Table (Distance) (Vector)		
Net.	Hops	Exit-int.
X	0	<--
Y	0	-->
W	1	RTA
Z	1	RTC

Routing Table (Distance) (Vector)		
Net.	Hops	Exit-int.
Y	0	<--
Z	0	-->
X	1	RTB
W	2	RTB

Convergenza!

# Inserimento di un nodo

- Supponiamo che la metrica sia il numero di tratte
- Ogni riga indica la distanza dopo l'i-esimo scambio di informazioni tra i nodi adiacenti a partire dalla attivazione del nodo A

A.....	B.....	C.....	D.....	E
1	inf	inf	inf	
1	2	inf	inf	
1	2	3	inf	
1	2	3	4	

- Dopo 4 scambi tutti i nodi sono già aggiornati

# Problemi con routing distance vector

- Il problema del conteggio all'infinito:

A	B	C	D	E
1	2	3	4	
3	2	3	4	
3	4	3	4	
5	4	5	4	
5	6	5	6	
7	6	7	6	
7	8	7	8	
:	:	:	:	
:	:	:	:	
$\infty$	$\infty$	$\infty$	$\infty$	

- Non tiene conto delle capacità delle linee
- Converge con tempi molto lunghi

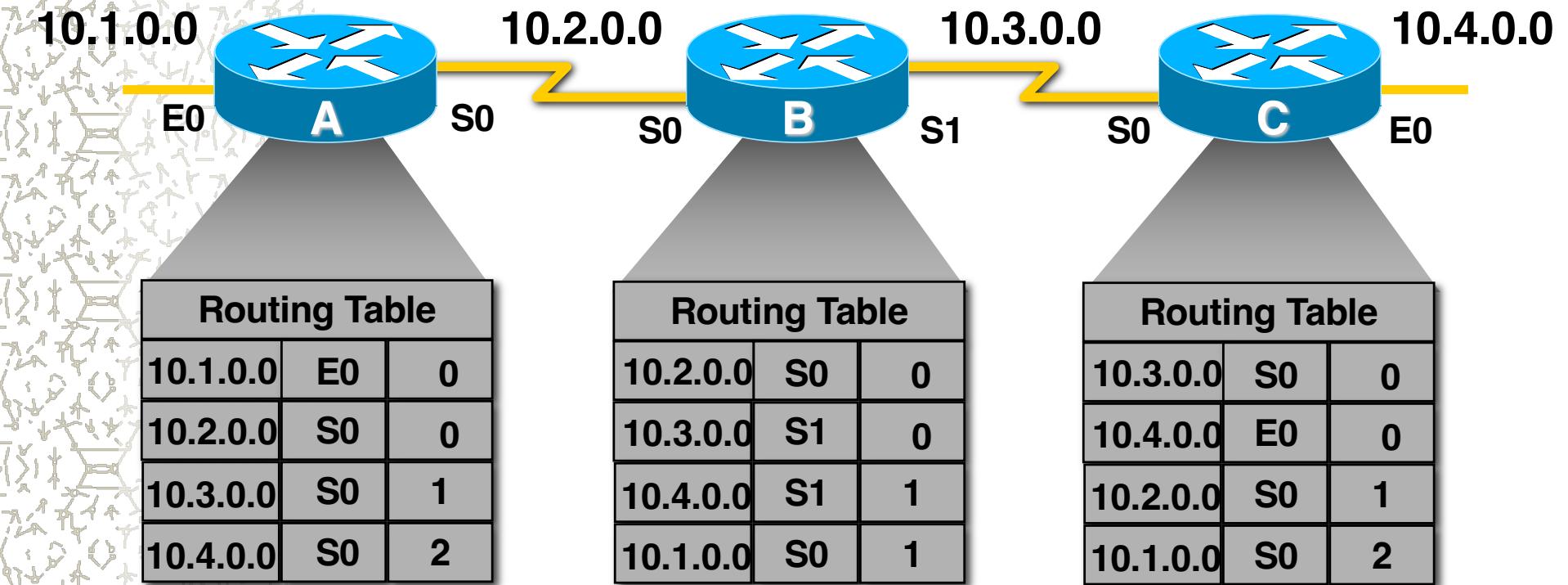
# Routing loops e conto all'infinito

- Supponiamo che A si disconnetta dalla rete

A.....	B.....	C.....	D.....	E
1	2	3	4	
3	2	3	4	
3	4	3	4	
5	4	5	4	
5	6	5	6	
7	6	7	6	
.	.	.	.	
inf	inf	inf	inf	

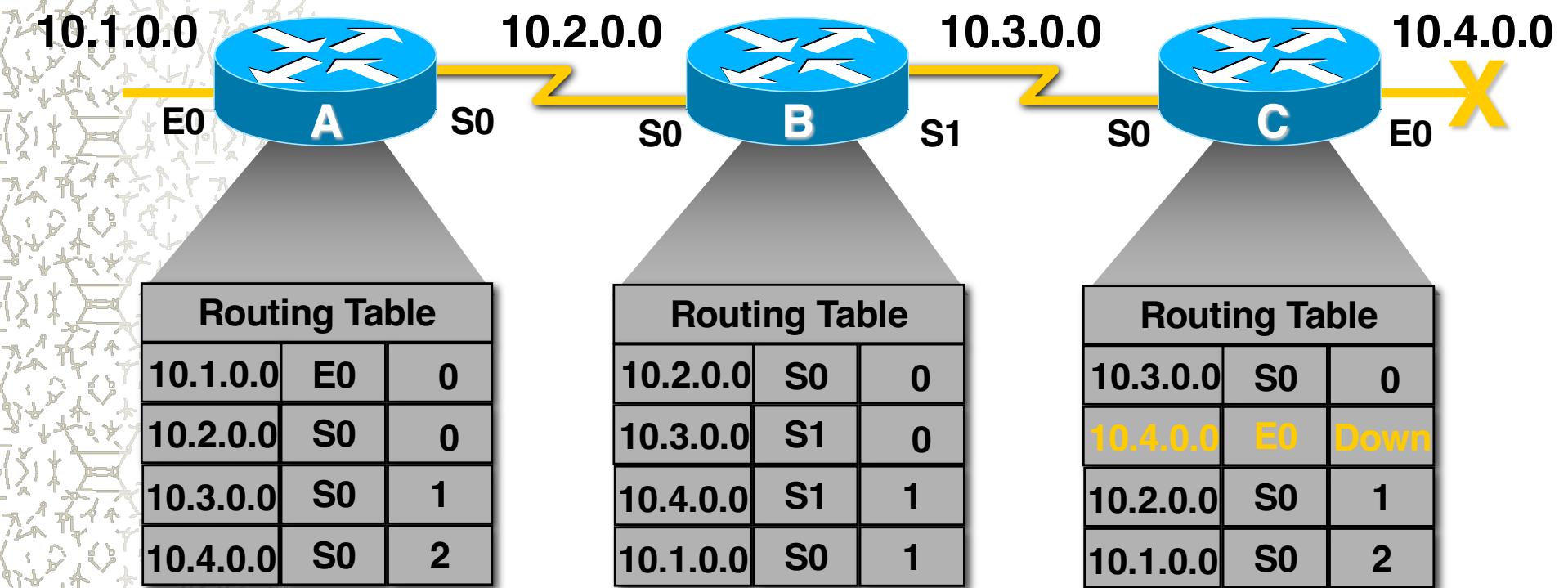
- B riconosce che A non risponde e che il suo vicino è a distanza 2 da A: pone erroneamente a 3 la sua distanza da A
- L'informazione errata si propaga agli altri nodi
- I costi vanno all'infinito

# Routing loops



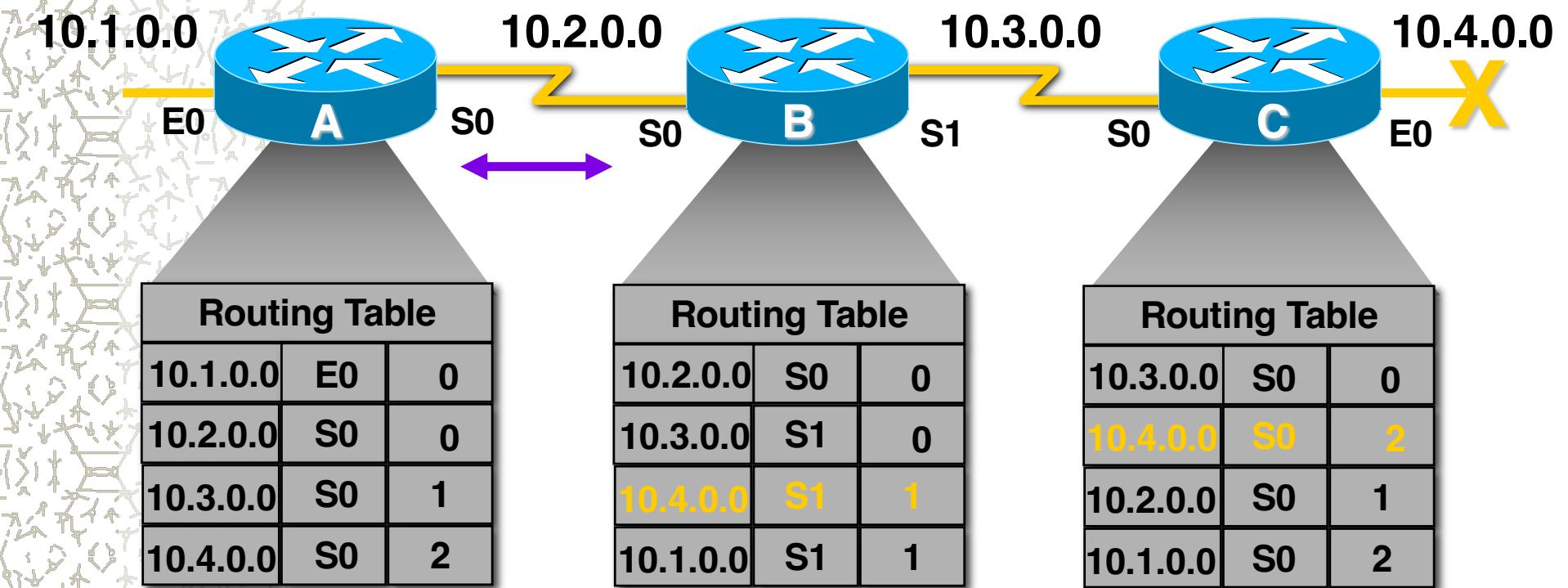
- Ogni nodo conserva la distanza fra se stesso e ogni possibile rete di destinazione

# Routing Loops



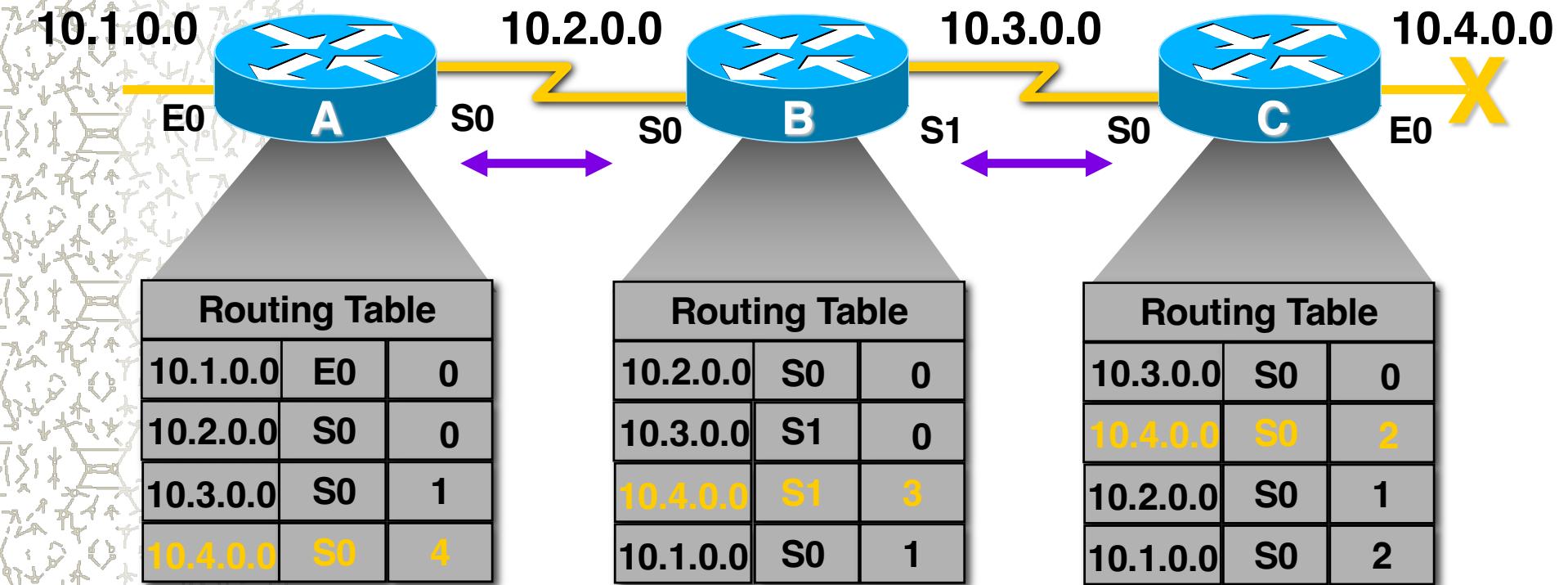
- Una convergenza troppo lenta può portare a informazioni inconsistenti

# Routing Loops



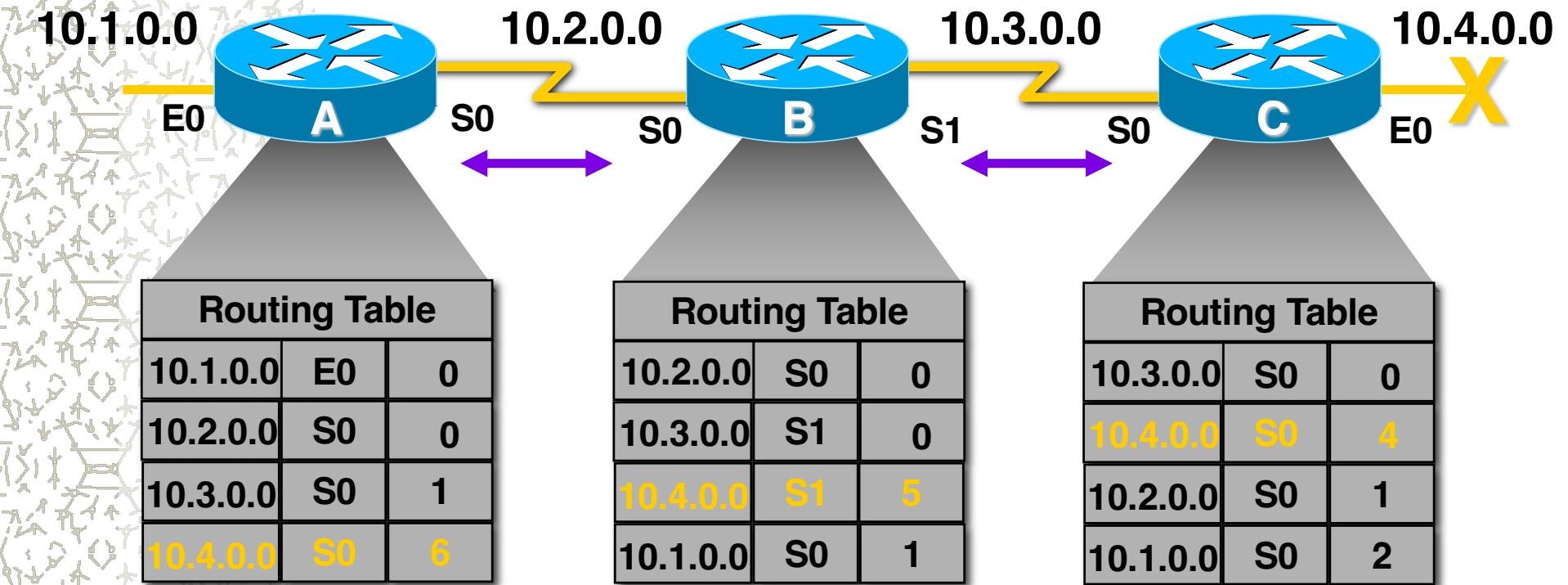
Il Router C conclude che il miglior percorso verso la network 10.4.0.0 è attraverso il Router B

# Routing Loops



Il Router A aggiorna erroneamente la sua routing table riflettendo il nuovo hop count che è errato

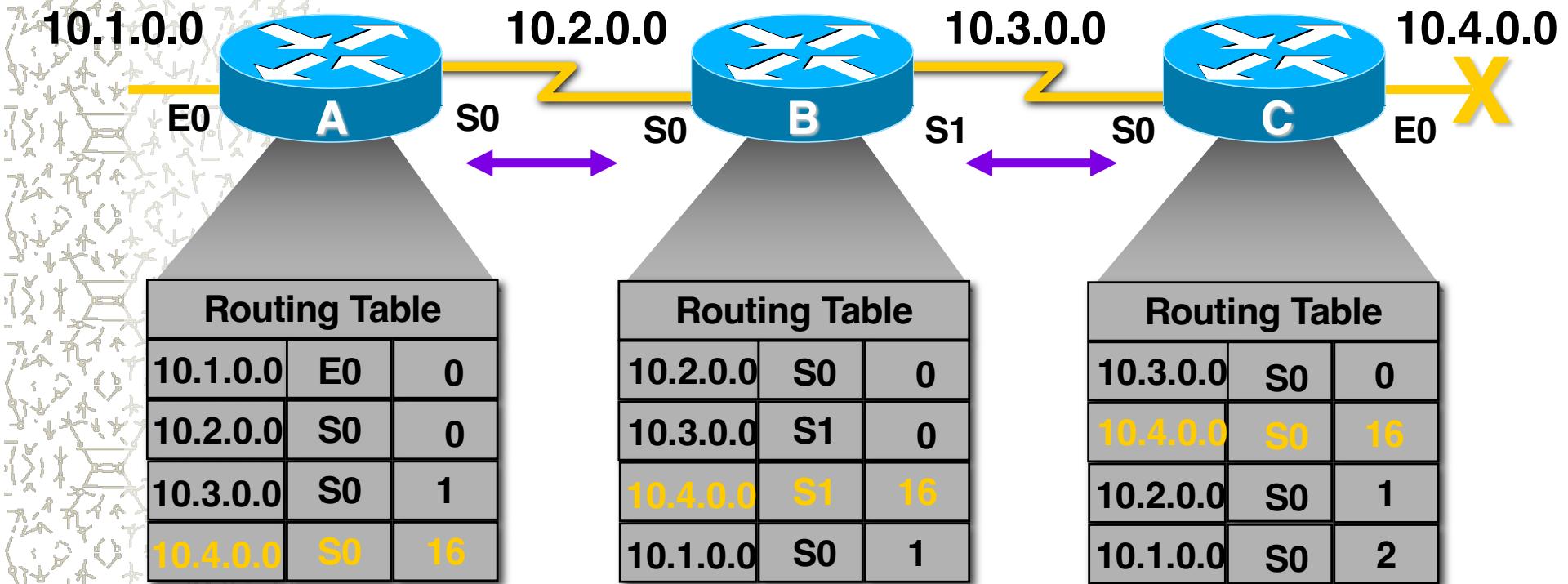
# Sintomi: Counting a infinito



I pacchetti per la network 10.4.0.0 rimbalzano fra i routers A, B, and C

L'hop count per la network 10.4.0.0 tende a infinito

# Soluzione: Limitare l'hop count



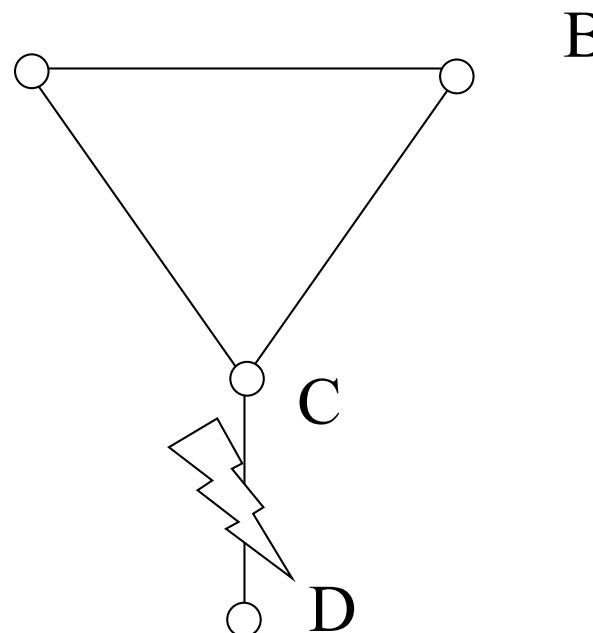
- Va definito un limite sul numero di hops per evitare loops

# Una soluzione: split horizon

- Non si inviano le informazioni di costo verso la destinazione X sul link al quale vengono inviati i pacchetti per la destinazione X
  - Nell'esempio precedente: C dice a D quanto dista da A ma dice a B che dista infinito da A
- In certi casi non serve



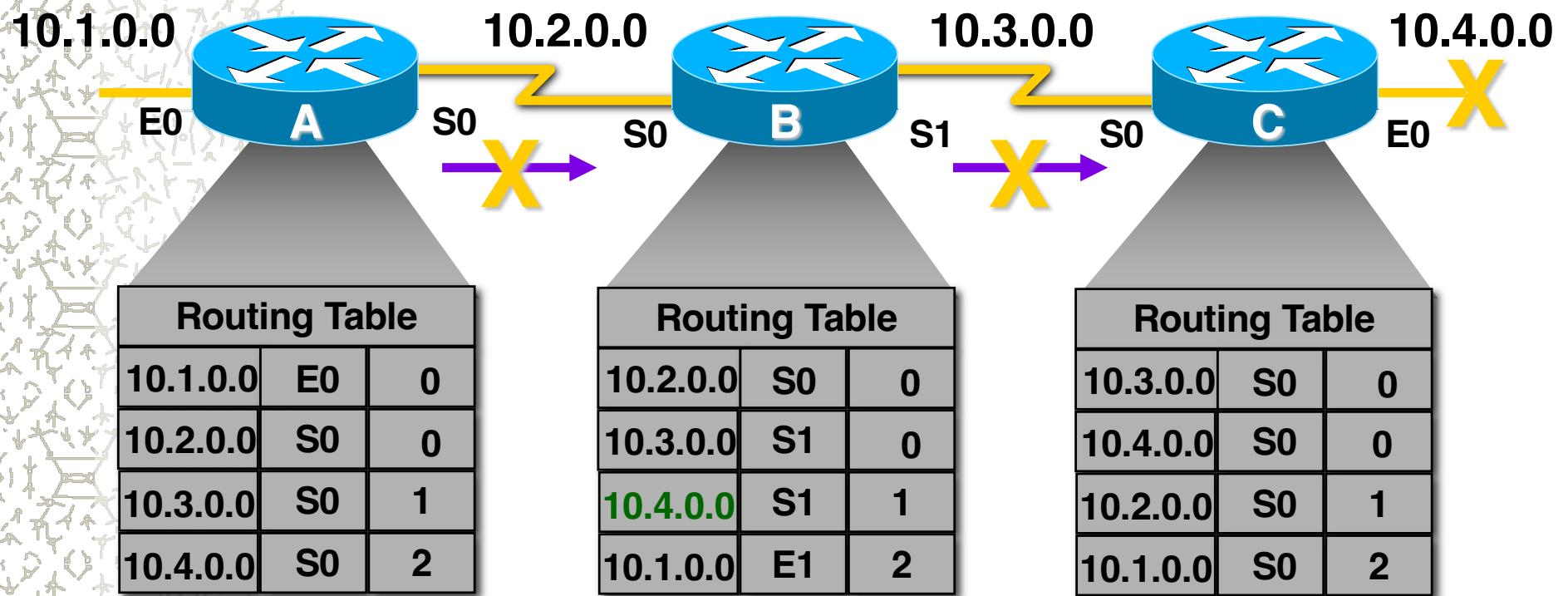
# Split horizon: esempio di fallimento



B

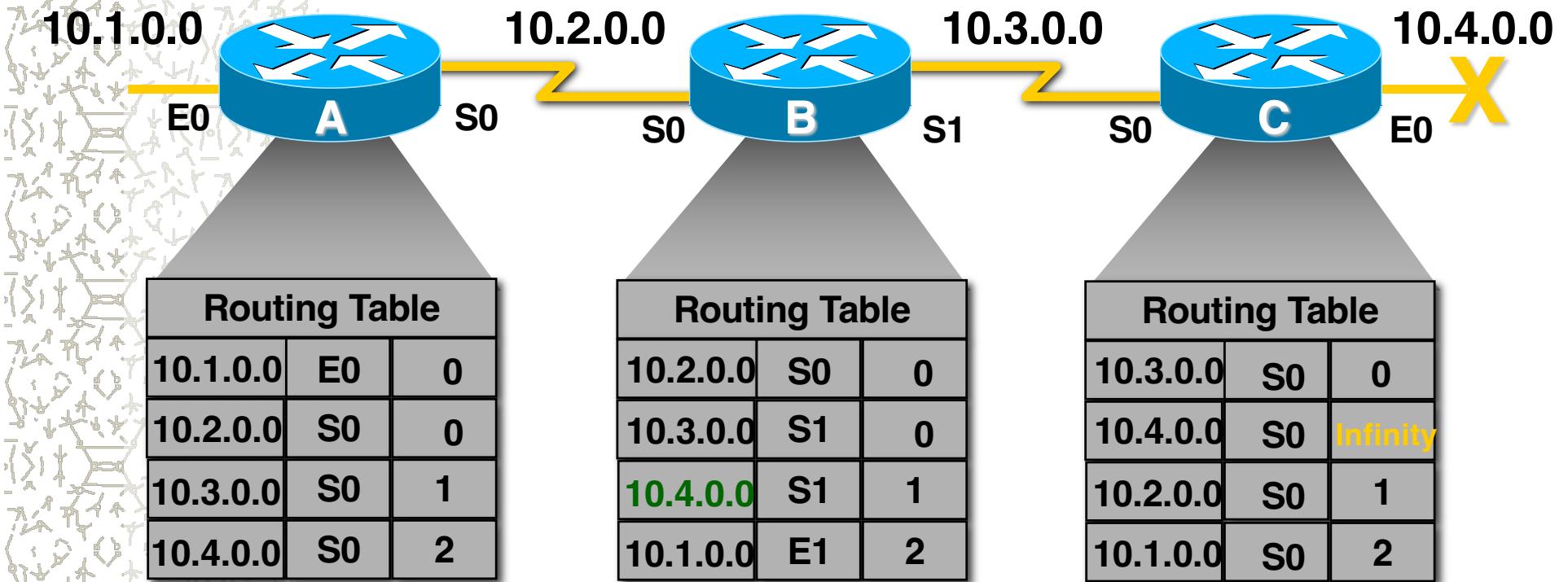
- Se C-D va fuori servizio si innesca nuovamente il conto all'infinito
- A infatti sa di avere un percorso alternativo verso B e la stessa cosa per B

# Split Horizon



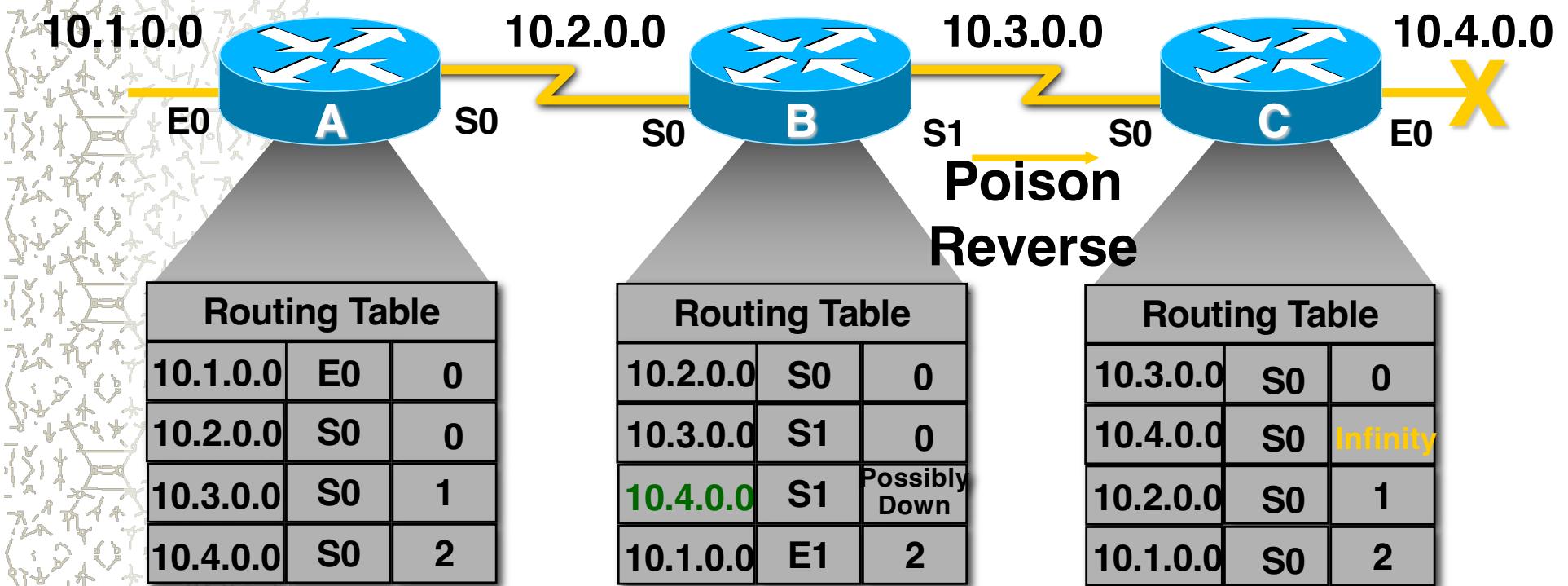
- Non è mai utile inviare informazione circa lo stato di una route sulla stessa interfaccia da cui le stesse sono state acquisite

# Route Poisoning



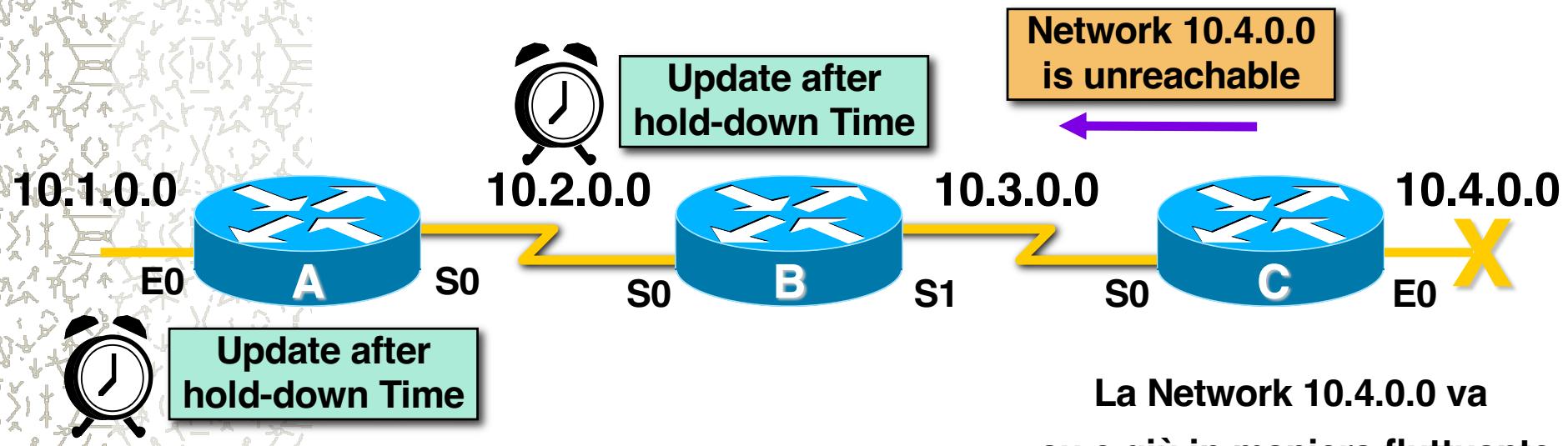
- Tutti i Routers settano la distanza di ogni route che passa in stato down a infinito

# Poison Reverse



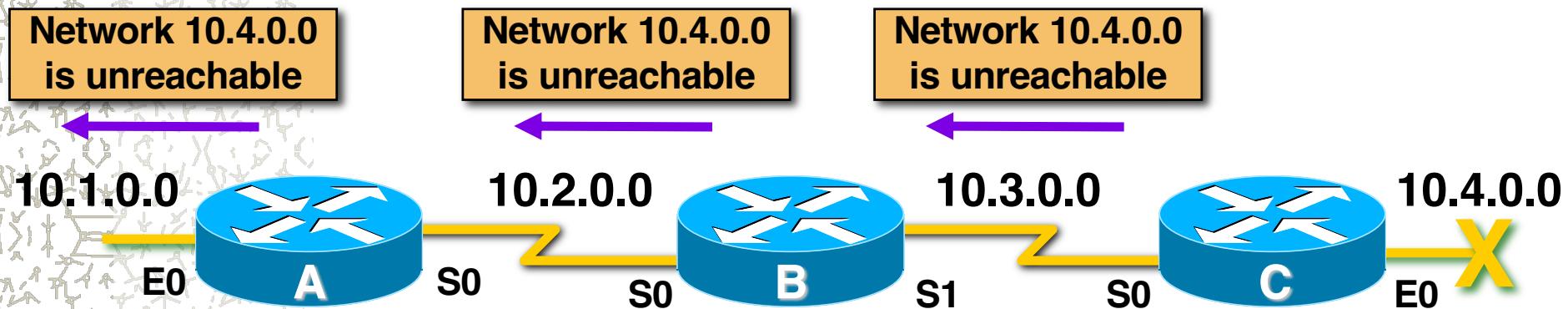
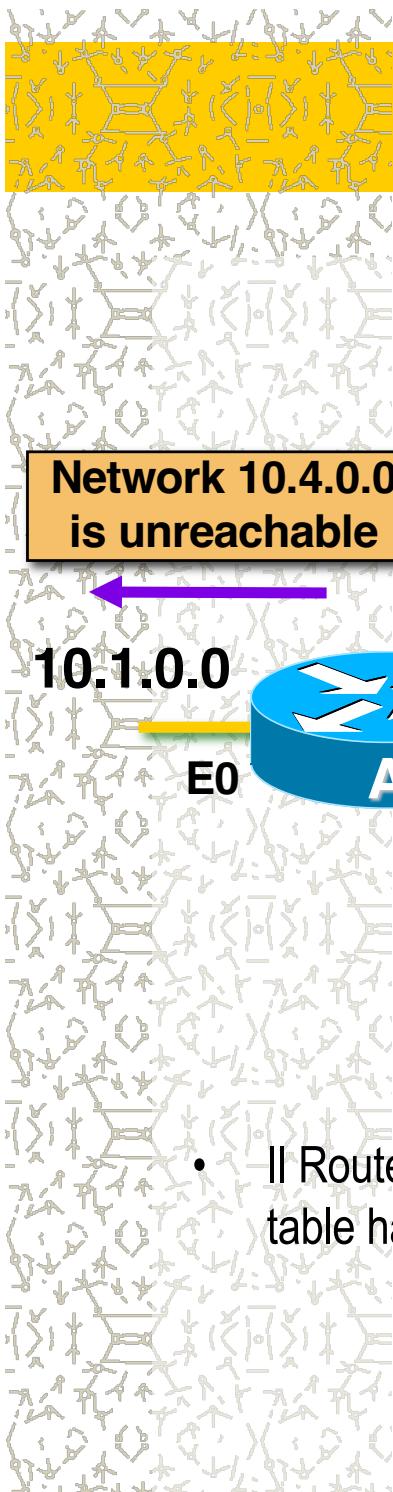
- Poison Reverse viola la regola dello split horizon

# Hold-Down Timers



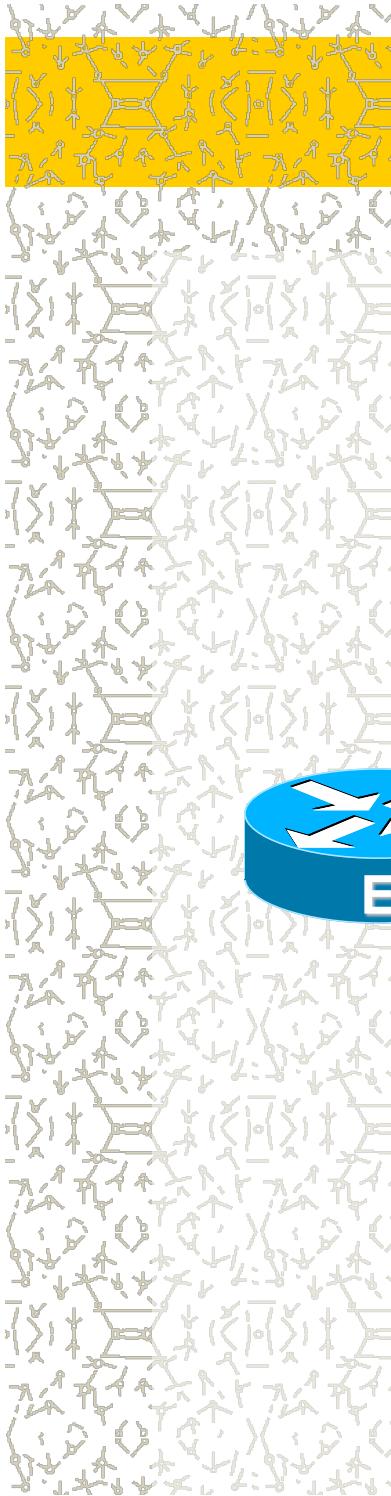
- Per evitare fluttuazioni alla ricezione di un annuncio il router setta un timer di hold-down e accetta la modifica solo alla spirazione dello stesso

# Triggered Updates

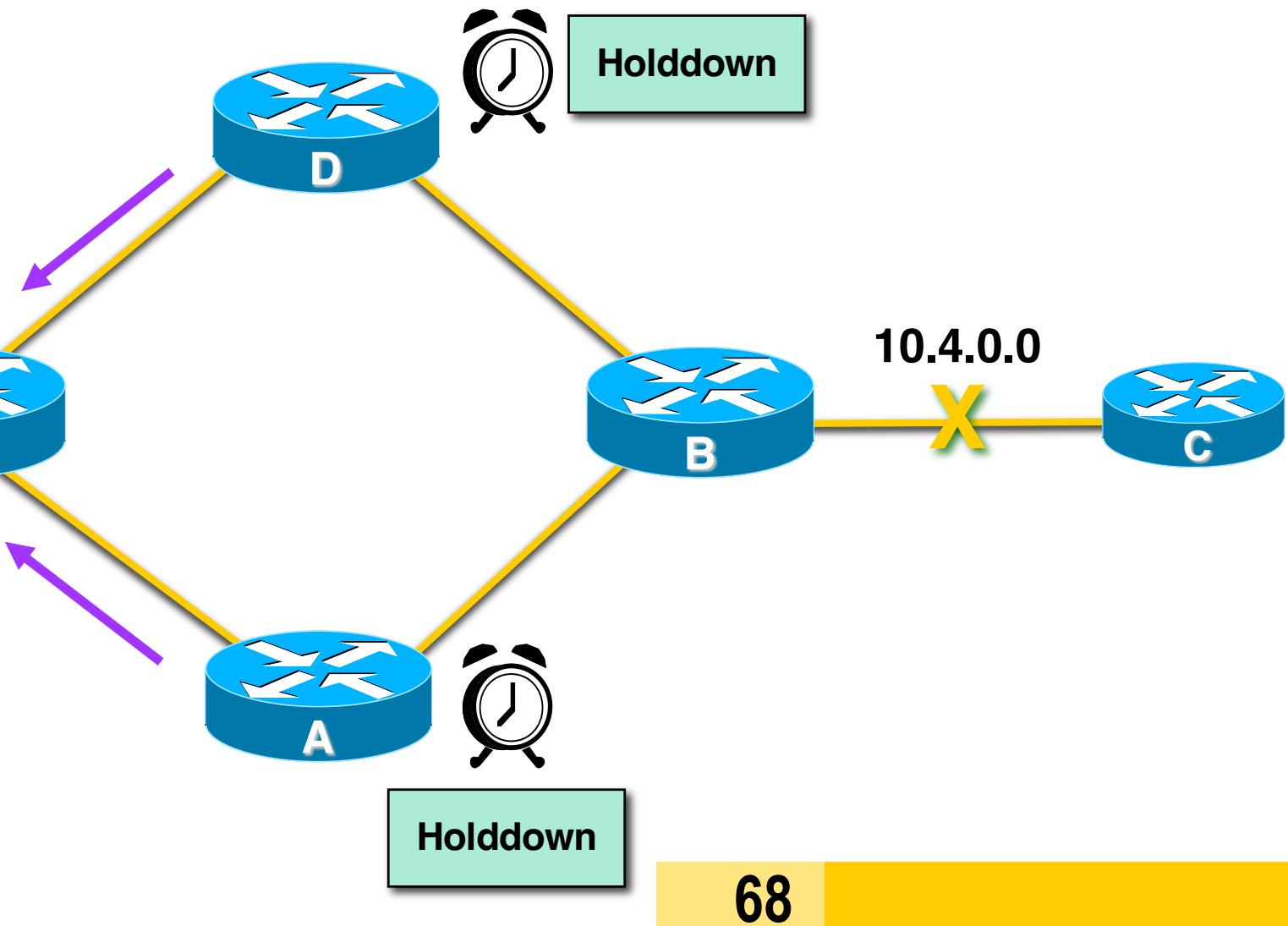


- Il Router invia un aggiornamento immediato appena una modifica nella sua routing table ha luogo

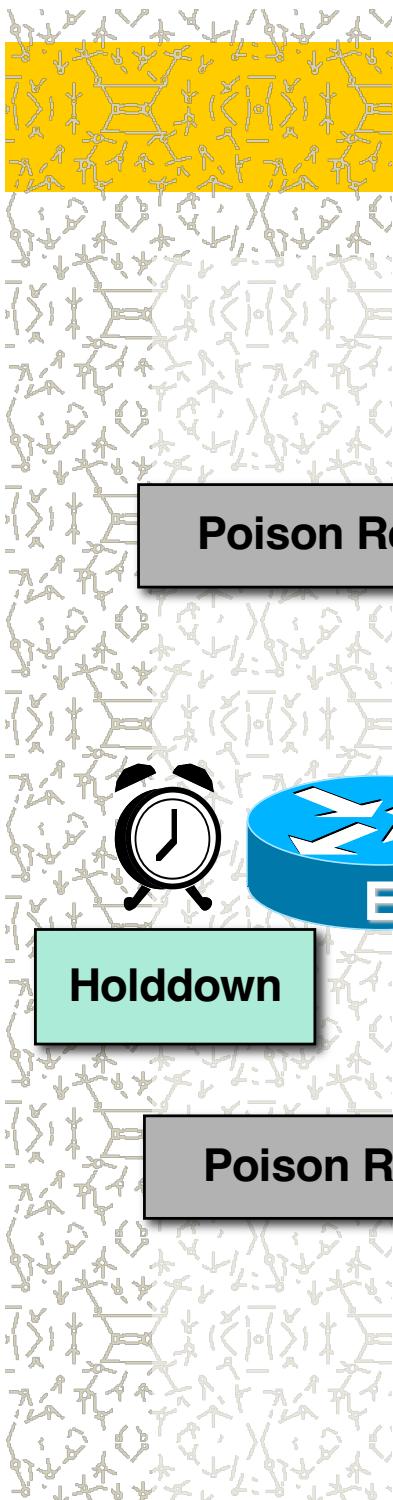
# Esempio



# Esempio



# Esempio



Poison Reverse



Holddown

Holddown

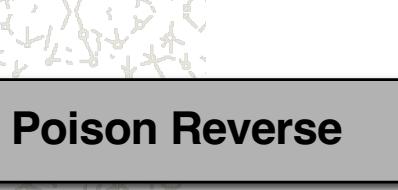
Poison Reverse

Holddown

69



10.4.0.0

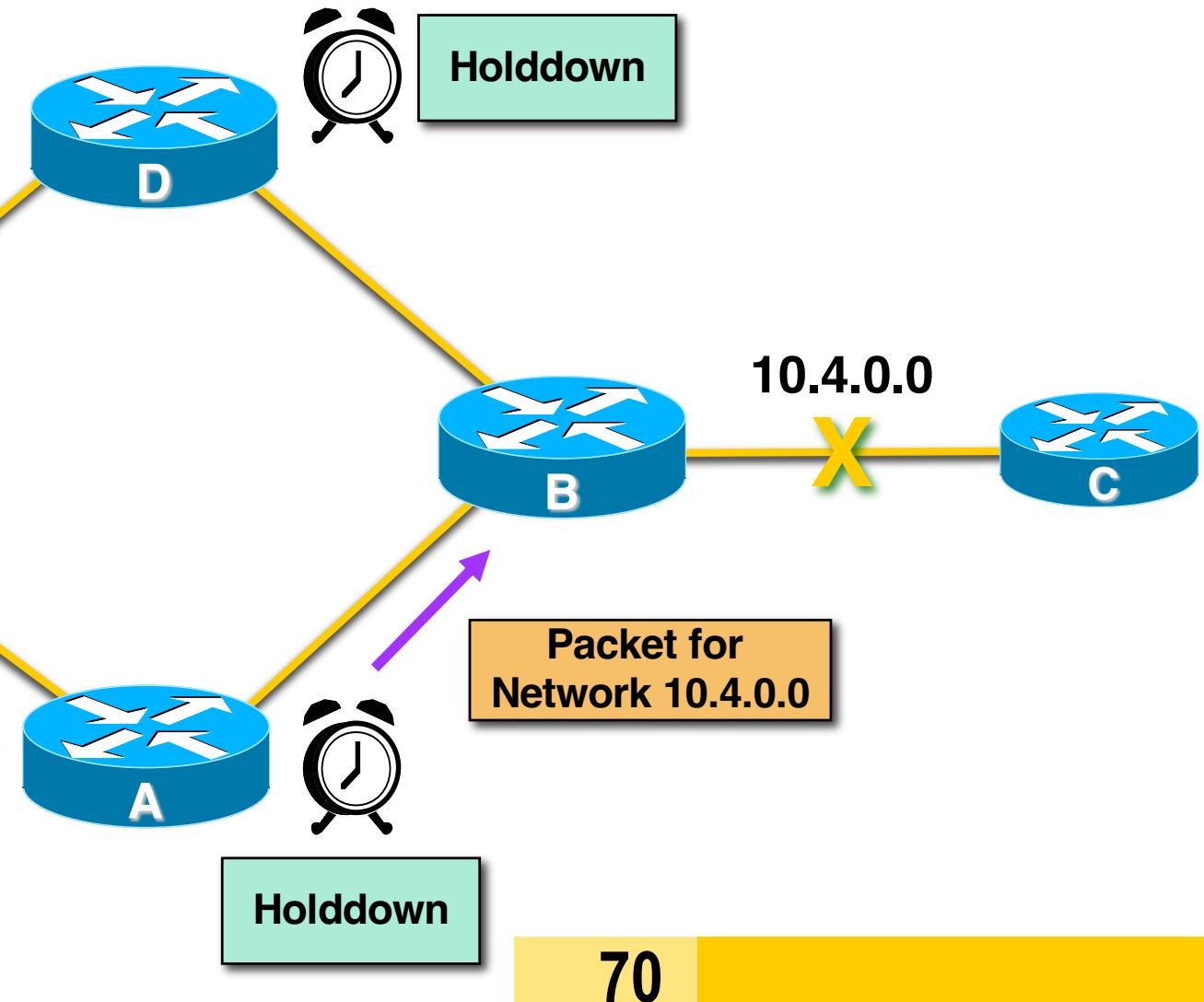


Poison Reverse

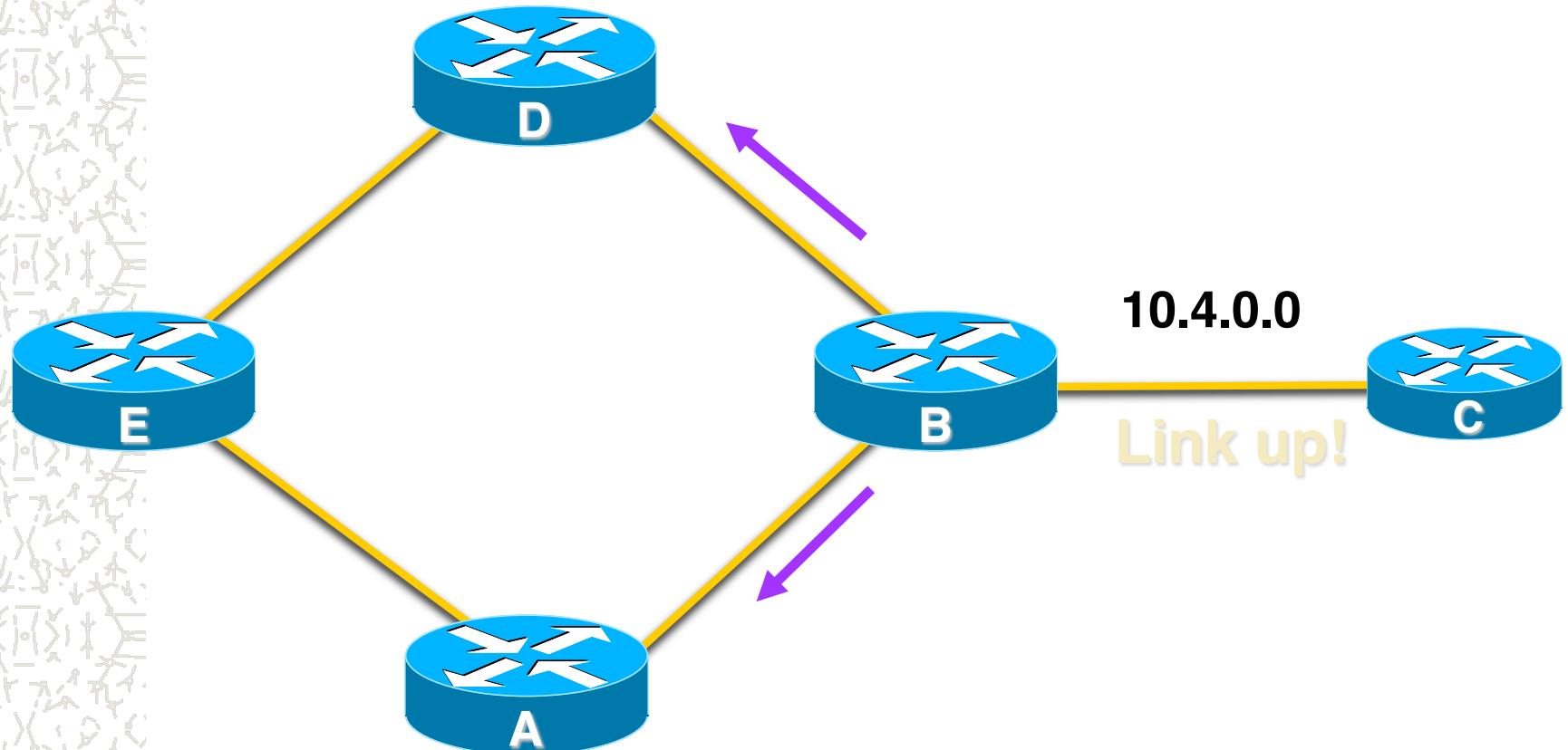
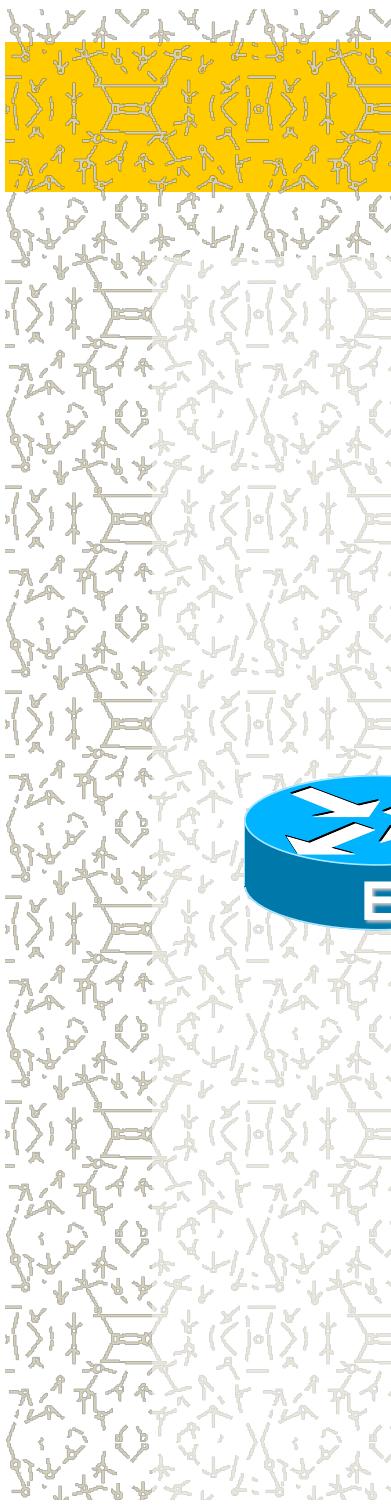
Poison Reverse

Poison Reverse

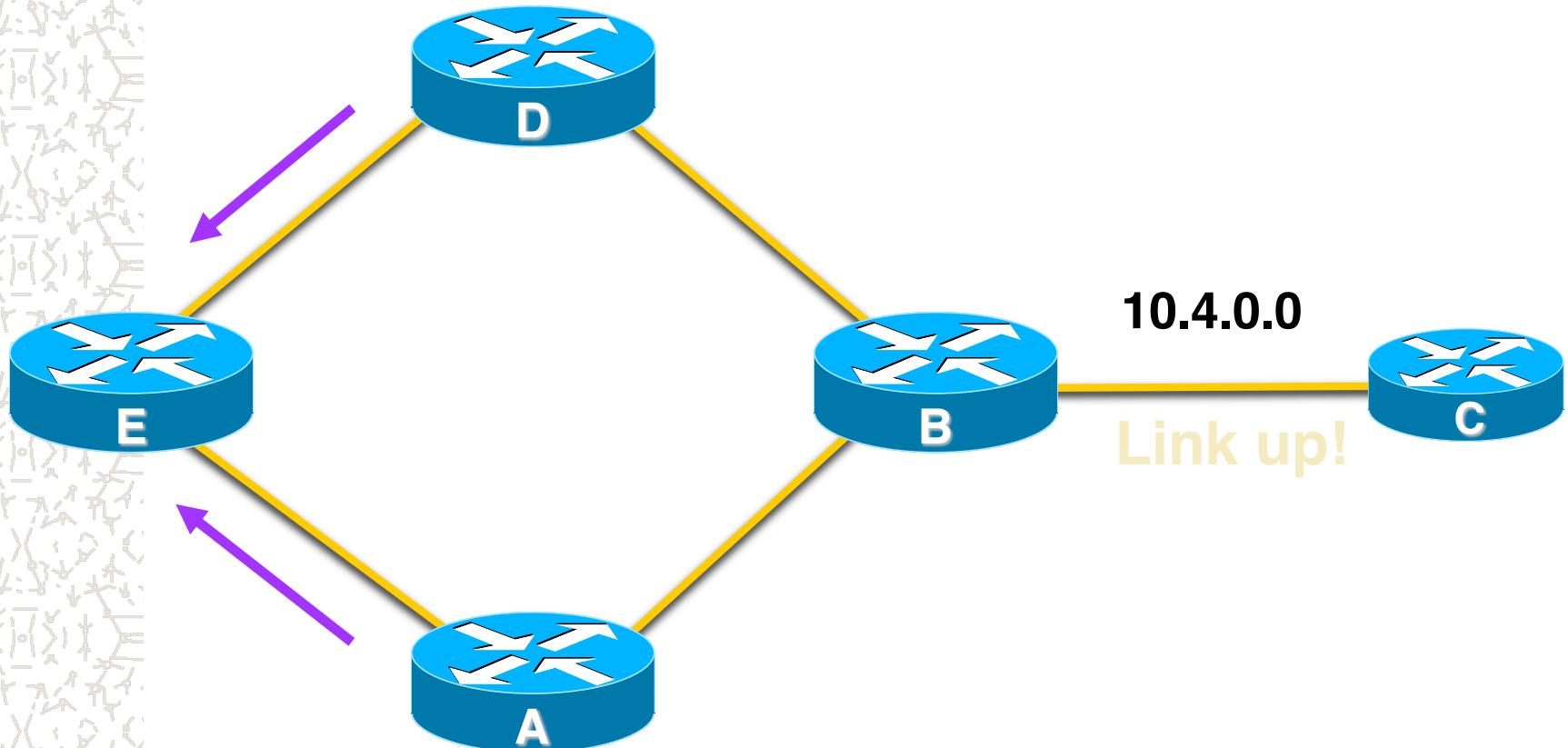
# Esempio



# Esempio



# Esempio



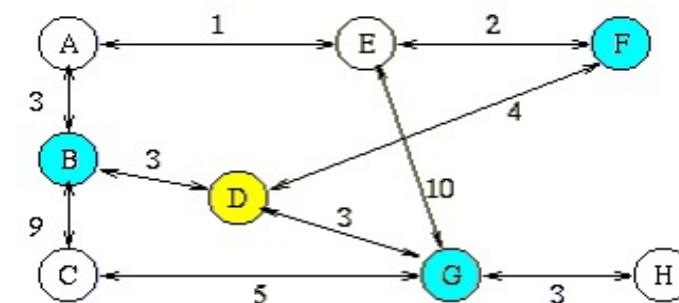


# Algoritmi di Routing link state

- Sono stati introdotti dal 1979 al posto degli algoritmi distance vector
- Si basa sull'invio di pacchetti contenenti le informazioni di costo e di ritardo relativi a ciascun link ai nodi adiacenti
- Ogni nodo utilizza queste informazioni per calcolare il costo minimo verso tutti i nodi a lui noti

# Link State Packet (1)

- Ogni router deve avere in memoria la mappa di tutta la rete e coopera per crearne una di tutta la rete, mantenendola aggiornata; poi calcola indipendentemente la propria tabella.
- I routers apprendono i primi n nodi vicini, associando ad ognuno di essi il costo della linea.
- L'informazione è propagata ai router della rete con un messaggio definito **Link State Packet** (LSP). La propagazione avviene con algoritmo di tipo Flooding.



# Link State Packet (2)

- La mappa della rete si costruisce fondendo ogni LSP in un database come mostrato in tabella.
- Il LSP database deve essere uguale per ogni router della rete, ne rappresenta la mappa e i costi associati, permette ai router di calcolare le tabelle di routing.

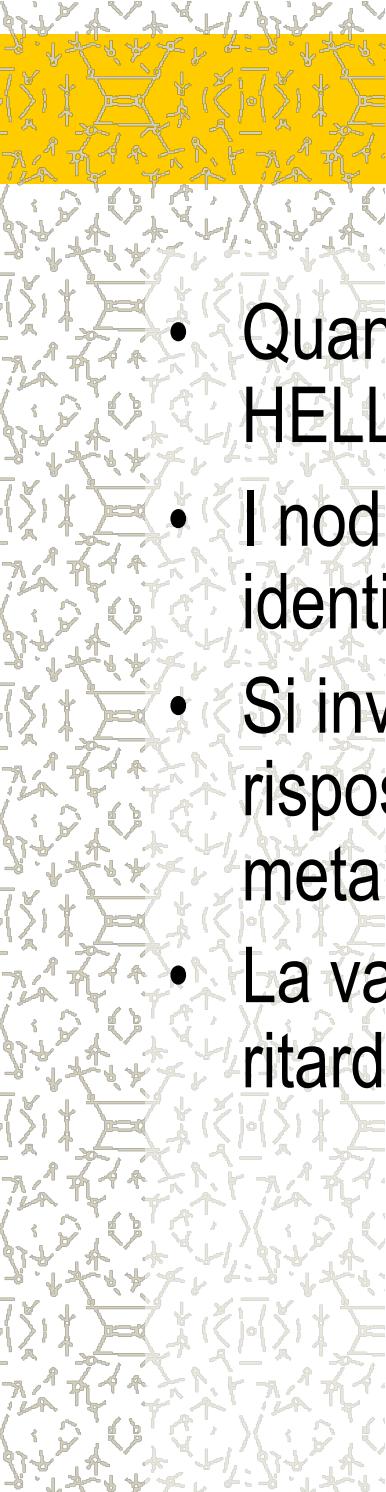
Link State Paket nodo D	
Adiacente	Costo
B	3
F	4
G	3

A	B/3	E/1	
B	A/3	C/9	D/3
C	B/9	G/5	
D	B/3	F/4	G/3
E	A/1	F/2	G/10
F	D/4	F/2	
G	C/5	D/3	E/10
H	G/3		



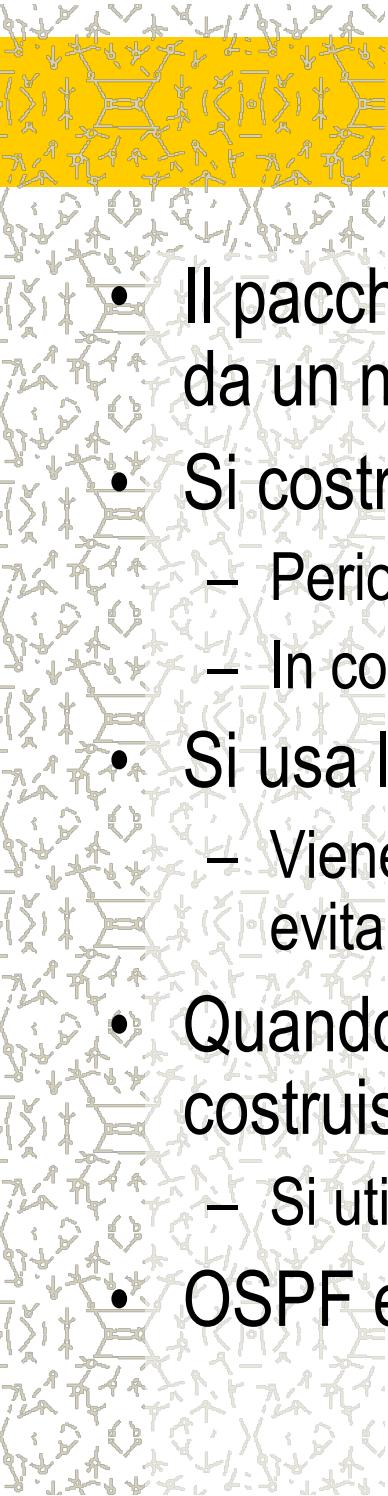
# Passi fondamentali

- Individuazione dei nodi vicini e dei relativi indirizzi
- Misura del ritardo e del costo verso ciascun vicino
- Costruzione dei pacchetti con le informazioni di routing
- Invio dei pacchetti a tutti gli altri router
- Calcolo del cammino più breve verso ciascun router



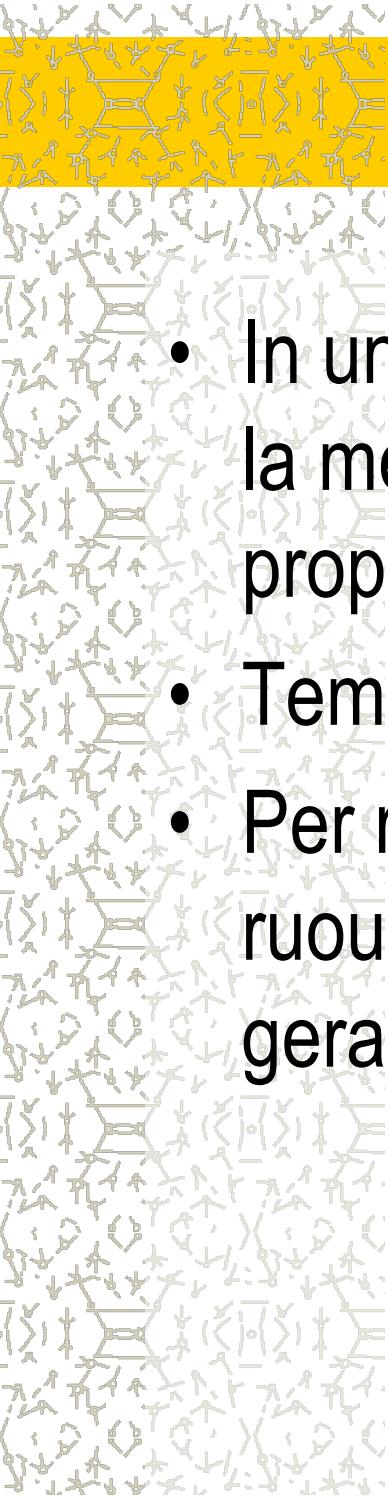
# Individuazione dei nodi vicini

- Quando un nodo si inserisce in rete invia un messaggio HELLO su ciscuna linea
- I nodi che ricevono il messaggio rispondono con il loro identificativo
- Si invia un messaggio ECHO a cui e' richiesta una risposta immediata per misurare il round trip time la cui metà e' una stima del ritardo sulla linea
- La valutazione del ritardo puo' comprendere o meno il ritardo di coda



# Pacchetti link state

- Il pacchetto inizia con l'identificativo di chi lo manda seguito da un numero di sequenza (32 bit) e dall'eta'
- Si costruiscono
  - Periodicamente
  - In conseguenza di variazioni significative
- Si usa la tecnica *flooding* per la distribuzione
  - Viene controllata la coppia (sorgente, numero di sequenza) per evitare di considerare piu' volte lo stesso pacchetto
- Quando un nodo ha ricevuto le informazioni da tutti gli altri costruisce il grafo di rete
  - Si utilizza Dijkstra per trovare i cammini minimi
- OSPF e' un algoritmo link state



# Requisiti degli algoritmi link state

- In una rete con  $n$  router e  $k$  nodi vicini per router la memoria richiesta in ciascun nodo è proporzionale a  $kn$
- Tempi di calcolo lunghi
- Per reti estese si realizzano sottosistemi di routing indipendenti con organizzazione gerarchica

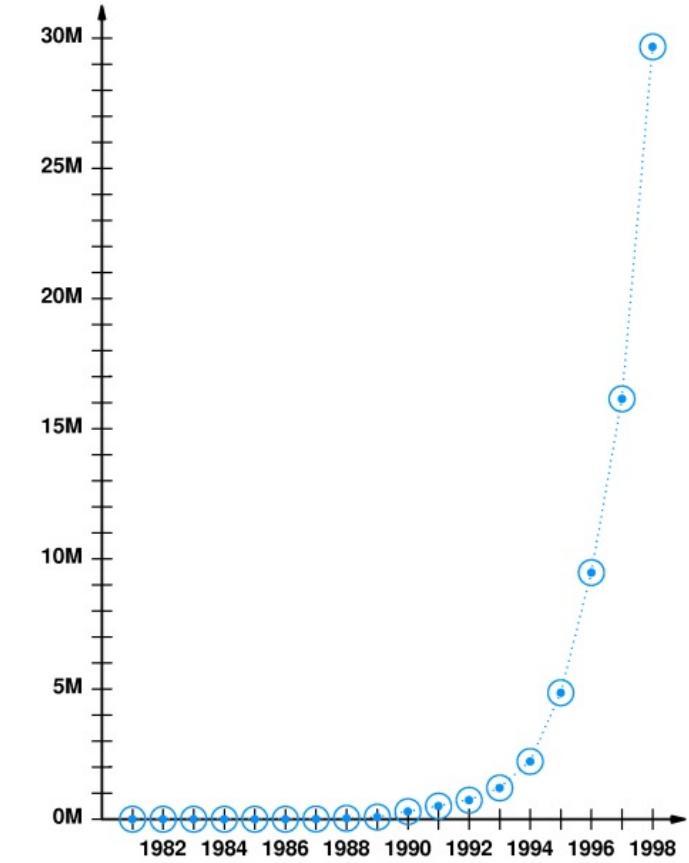
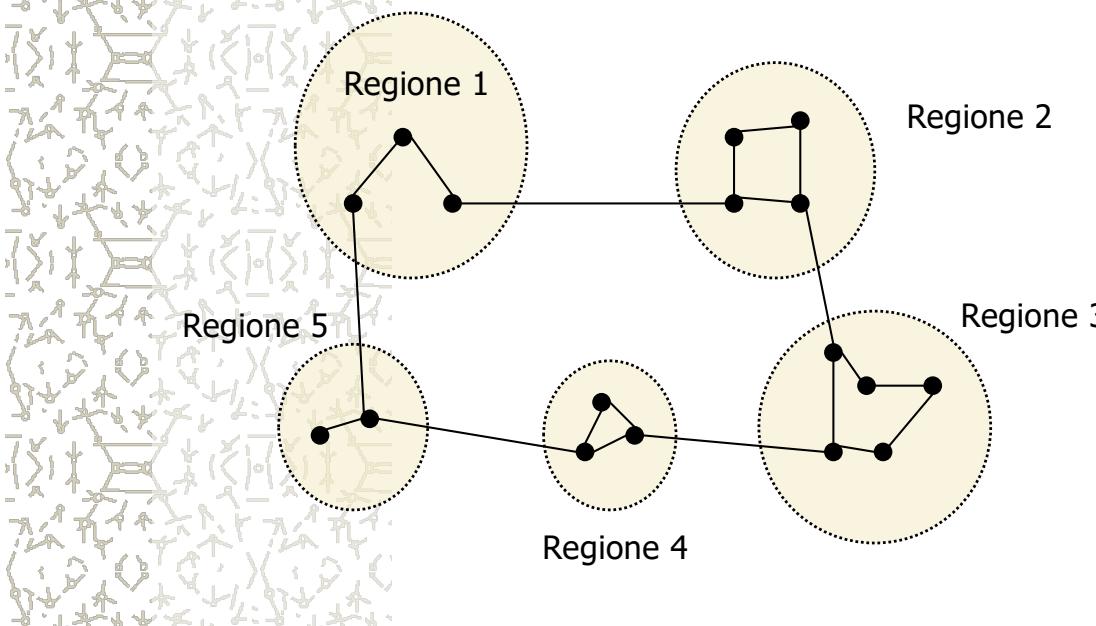


# Protocolli di Routing Gerarchico

- Non potendo gli algoritmi LSP gestire qualsiasi rete di qualsiasi dimensione, occorre organizzare il routing in modo gerarchico, cioè suddividere la rete in aree, dove il routing segue esattamente le regole descritte in precedenza.
- Per comunicare tra due nodi appartenenti ad aree diverse è necessario conoscere:
  - l'instradamento tra il nodo mittente e la periferia dell'area cui il nodo mittente appartiene
  - instradamento tra l'area mittente e l'area destinazione
  - instradamento all'interno dell'area destinazione.

# Routing gerarchico

- A causa della crescita esponenziale di Internet, le tabelle di routing diventano sempre più grandi. Quindi si divide il gruppo di router in regioni. Ogni router conosce i dettagli della propria regione e come comunicare con le altre, ma non conosce la loro struttura interna.
- Puo' occasionalmente generare cammini non ottimali, ma il vantaggio in termini di riduzione delle tabelle di routing vale la spesa



# Protocolli IGP: RIP

- Il primo protocollo di routing interno utilizzato in Internet e' il RIP (**Routing Information Protocol**), ereditato da Arpanet
- RIP e' un protocollo basato sull'algoritmo **distance vector**
- Deve la sua diffusione al fatto che una sua implementazione (**routed**) era compresa nella Berkeley Software Distribution di Unix che supportava il TCP/IP
- Adatto a reti di dimensioni **limitate**, ha iniziato a mostrare i suoi limiti già alla fine degli anni '70
- Attualmente ancora utilizzato come protocollo di routing in qualche **piccola rete privata**

# RIP (cont.)

- Caratteristiche del modello distance vector utilizzate da RIP:
  - Usa **numero degli hop** come metrica per il costo dei link: tutte le linee hanno **costo 1**
  - Il costo massimo e' fissato a **15**, quindi impone il suo uso su reti di estensione limitata (diametro inferiore a 15 hop)
- Le tabelle di routing vengono scambiate tra router adiacenti ogni **30s** via un messaggio di replica del RIP (*RIP response message*) o avviso di RIP (*RIP advertisement*)
  - Questo messaggio contiene voci della tabella del mittente per un **massimo di 25 reti** di destinazione dell'AS

Header	0	7	15	31
	Command	Version	Must be zero	
	AFI		Must be zero	
Route Entries		IP address		
		Must be zero		
		Must be zero		
		Metric		



# Protocolli IGP: OSPF

- Nel 1979 Internet e' passata ad un protocollo di tipo **link state**
  - ciascun router utilizza il flooding per propagare lo stato delle sue connessioni agli altri router della rete
  - ciascun router **conosce la topologia completa** della rete
- Alla fine del 1988 e' stato sviluppato un successore chiamato OSPF (**Open Short Path First**), definito nell'RFC 2328
- OSPF e' oggi il piu' diffuso protocollo IGP utilizzato in Internet

# Caratteristiche di OSPF

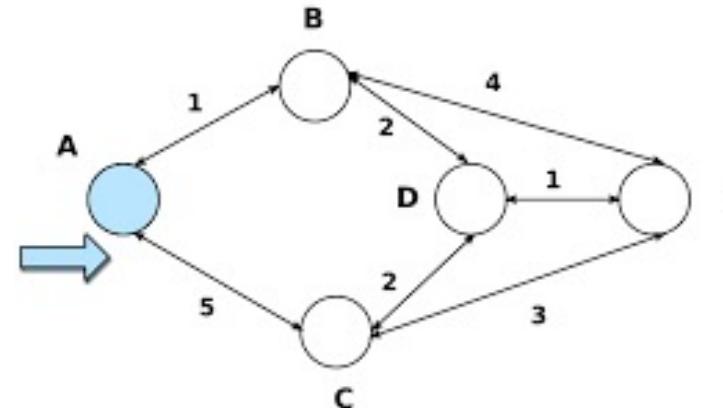
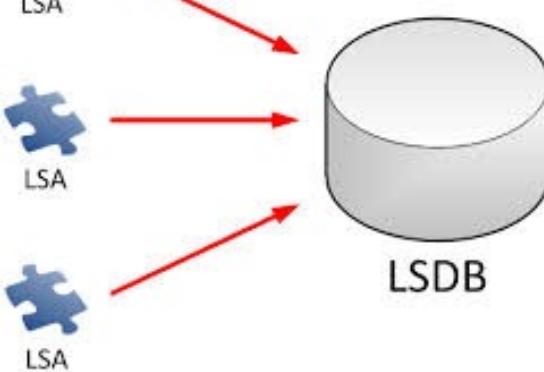
- OSPF e' stato progettato cercando di soddisfare diversi requisiti:
  - nessun vincolo di brevetto (O = Open)
  - supporto di diverse metriche per la distanza (distanza fisica, hop, ritardo, costo della linea, ...)
  - algoritmo capace di reagire dinamicamente e rapidamente ad eventi che modificano la topologia (link state)
  - instradamento basato sul tipo di servizio (sfruttando i campi esistenti nell'header di IP, ad esempio)
    - questo e' stato incluso in OSPF, ma tutti (implementazioni ed applicativi) hanno continuato ad ignorare questa possibilita'
  - capacita' di bilanciare il carico su diversi cammini
  - supporto per sistemi gerarchici (quindi routing gerarchico anche all'interno dello stesso AS)
  - implementazione di sicurezza
  - supporto per il tunneling

# Connessioni di OSPF

- OSPF gestisce tre tipi di connessione:
  - linea **punto-punto tra router**
  - rete **locale multiaccesso**
    - una rete locale dotata di piu' accessi, quindi di piu' router
  - rete **geografica multiaccesso**
    - rete geografica connessa al resto dell'AS da piu' di un router
- OSPF considera la rete come un grafo con i router come punti, e le linee come archi
  - ogni **linea fisica** e' costituita da **due archi**, uno per ogni verso
  - una rete multiaccesso e' considerata come un **punto del grafo**, connessa a ciascun router della rete da una coppia di archi
- OSPF assegna un **costo** ad ogni arco, e determina il **cammino piu' breve** in base al costo complessivo del tragitto
  - i cammini nei due versi di una singola connessione **possono** avere costi **differenti**
- I nodi di rete non hanno peso in OSPF, ed la loro connessione al router viene valutata a **costo 0**

# Funzionamento OSPF

- Ciascun router invia a tutti gli altri router dell'area lo stato dei suoi collegamenti (**LSA**)
- Ogni router ha una visione completa della rete memorizzata in un suo **LS DB** (collezione di LSA) o database topologico
- Attraverso l'algoritmo di **Dijkstra** ogni nodo calcola individualmente il percorso di minor costo da se verso ogni altro nodo dell'area
- Eventuali modifiche vanno segnalate a tutti i nodi nell'area (broadcast/flooding)



# OSPF: Messaggi

- **LINK STATE UPDATE**

- Aggiornamenti sullo stato di link o nodi
- Ogni messaggio ha un numero di sequenza
- I messaggi vengono riscontrati (LS ACK)

- **LINK STATE REQUEST**

- E' una richiesta esplicita di informazioni

- **DATA BASE DESCRIPTION**

- Fornisce i numeri di sequenza delle informazioni link state possedute da chi lo spedisce
- E' una copia del LS DB usata per sincronizzare un nuovo nodo

0	8	16	24	31
LS age	Options	LS type		
Link State ID				
Advertising Router				
LS sequence number				
LS checksum		Length		



# OSPF: le aree

- Al crescere delle dimensioni anche alcuni AS sono diventati **troppo complessi** per essere gestiti in modo non gerarchico
- OSPF permette di dividere un AS in piu' **aree**
  - ogni area e' una rete o un insieme di reti **direttamente** connesse
  - le aree **non** hanno componenti **sovraposti**
  - in ciascun AS deve esistere un'area **dorsale** (l'area 0) a cui sono collegate **tutte** le altre aree dell'AS
- e' possibile passare da un'area qualunque ad un'altra area dell'AS **attraverso la dorsale**, che le connette tutte

# OSPF: tipi di router

- OSPF definisce quattro tipi di router
  - router interni (**internal router**): router interni ad un'area
  - router di confine (**border router**): router che collega due o piu' aree (una e' sempre l'area della dorsale)
  - router di dorsale (**backbone router**): router interno della dorsale
  - router di confine dell'AS (**boundary router**): router che collega l'AS ad uno o piu' AS differenti
- Naturalmente le funzioni possono **sovraporsi**
  - un router di confine e' anche router interno
  - un boundary router e' anche un backbone router
  - un backbone router non connesso ad altre aree o ad altri AS e' un router interno (per l'area 0)

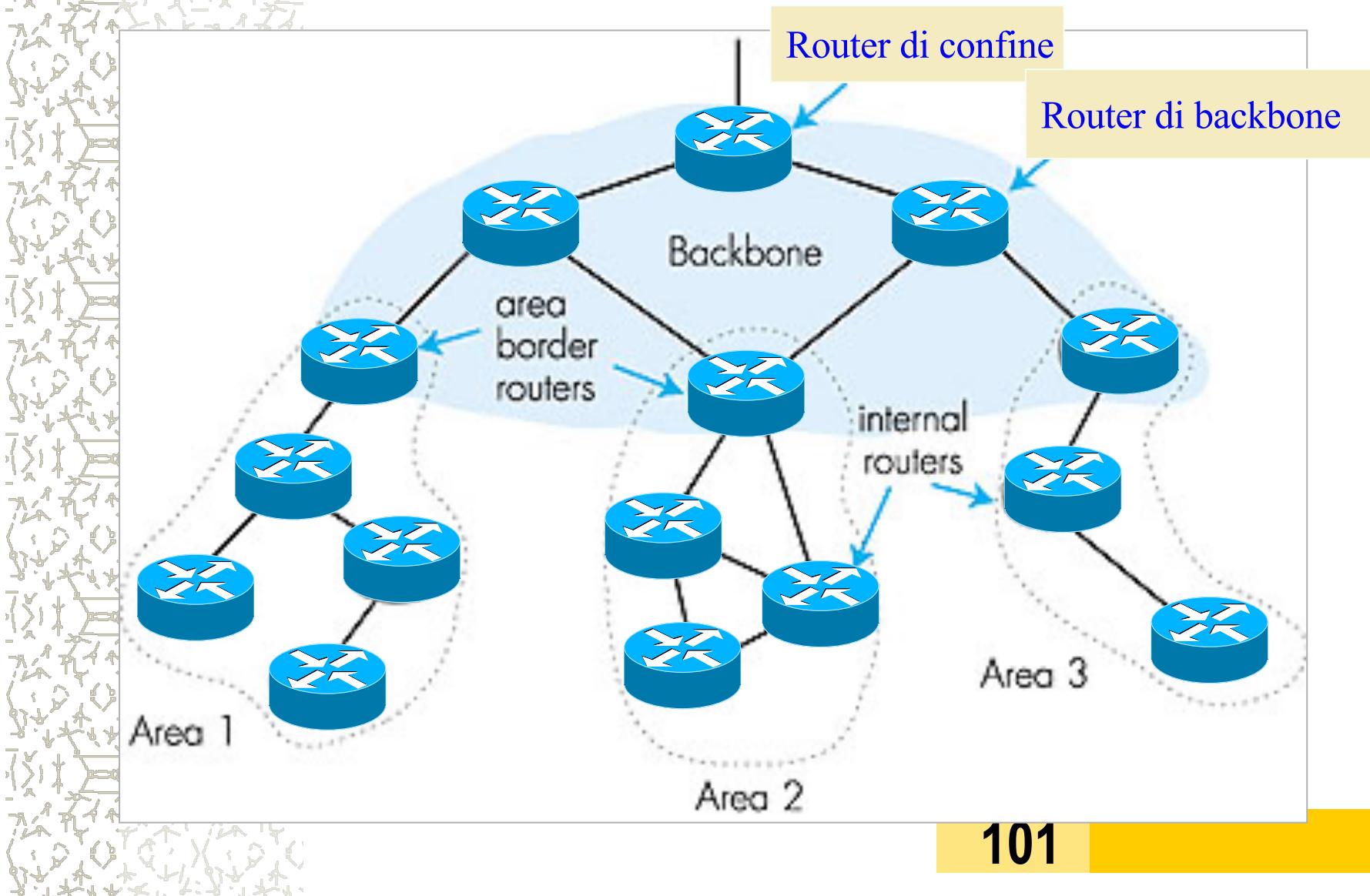
# OSPF: Instradamento fra aree

- Intra-area
  - Il router conosce già il cammino minimo
- Inter-area
  - Dalla sorgente al router del backbone
  - Dal router del backbone all'area destinazione
  - Alla destinazione
- Inter-AS
  - Si utilizza un protocollo di routing esterno

# OSPF: costruzione delle tabelle

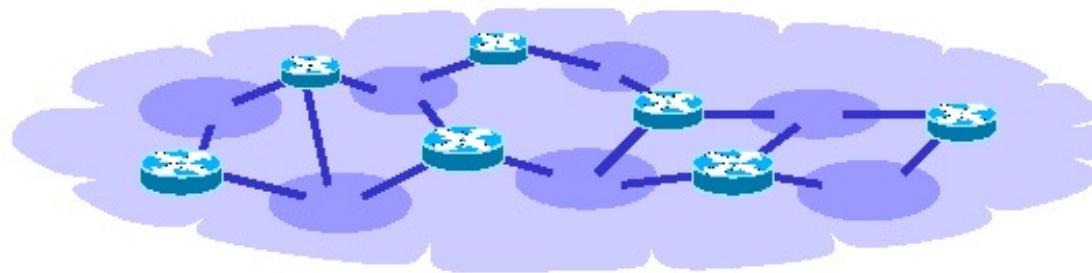
- Secondo la logica link state, ogni router di ogni area conosce la **topologia dell'area** e puo' effettuare routing interno
- All'interno di un'area i router hanno lo stesso insieme di dati link state
- Un router collegato a piu' aree mantiene i data base di ciascuna
- I router di **backbone** accettano le informazioni anche dai **router di confine** delle altre aree
  - in questo modo ogni router di backbone sa **a quale** router di confine di un'area inviare i pacchetti destinati a quella area
- Le informazioni delle adiacenze dei router di confine verso l'area di backbone sono propagate **dentro l'area**
  - in questo modo ogni router interno sa **a quale** router di confine della propria area inviare un pacchetto destinato ad un'altra area

# Struttura gerarchica di OSPF



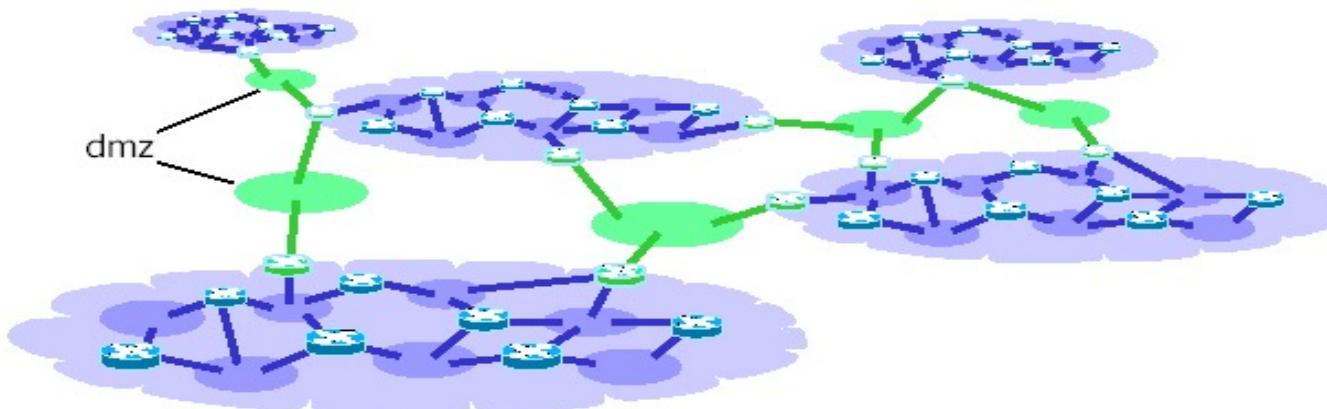
# I Sistemi Autonomi

- Ogni organizzazione è composta da un insieme di router e LAN sotto una singola amministrazione
- Un algoritmo di routing è prescelto per aggiornare automaticamente le tabelle di instradamento
- Un AS definisce in maniere coerente le politiche di instradamento all'interno della sua organizzazione

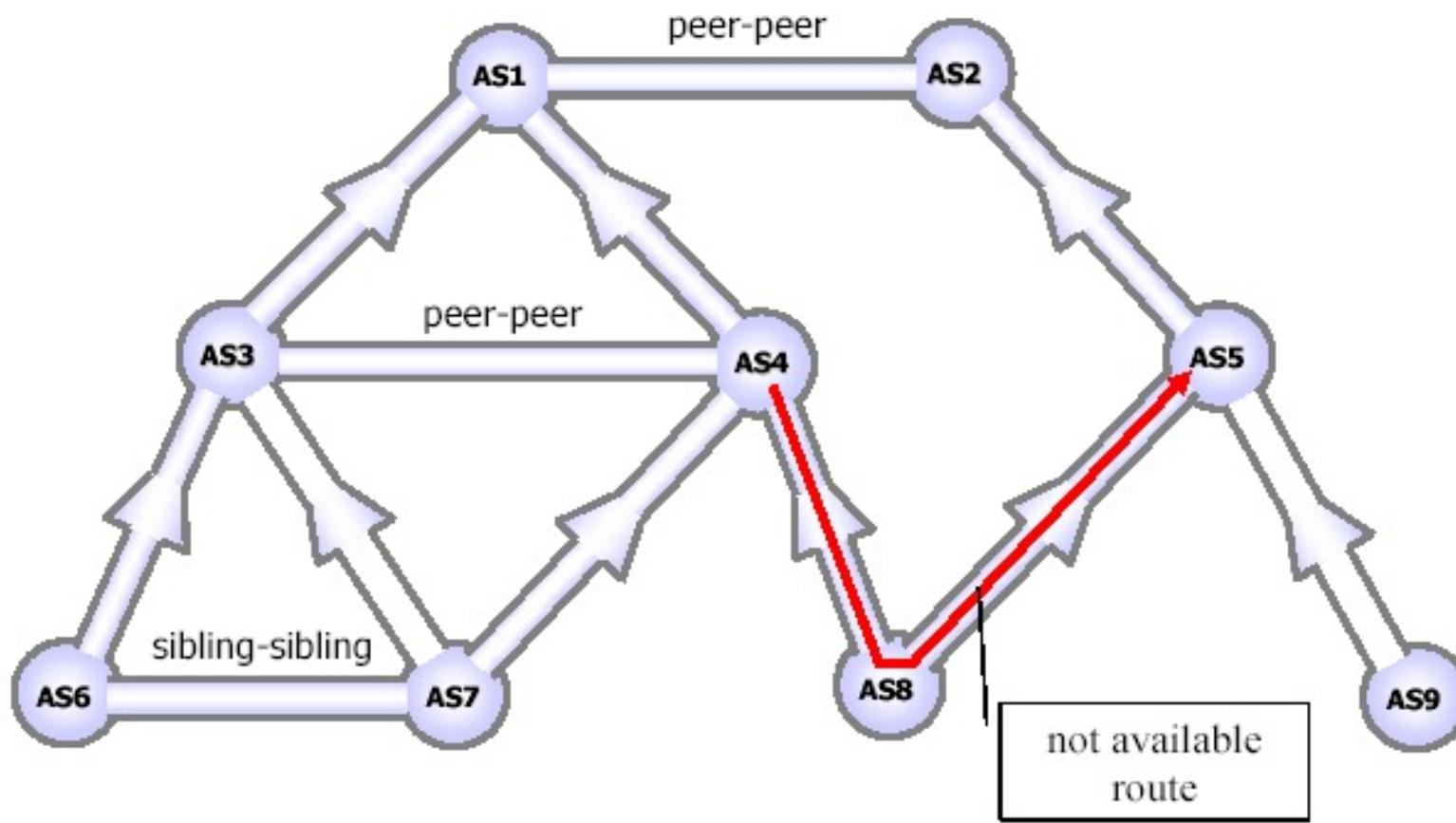


# L'interconnessione di Sistemi Autonomi

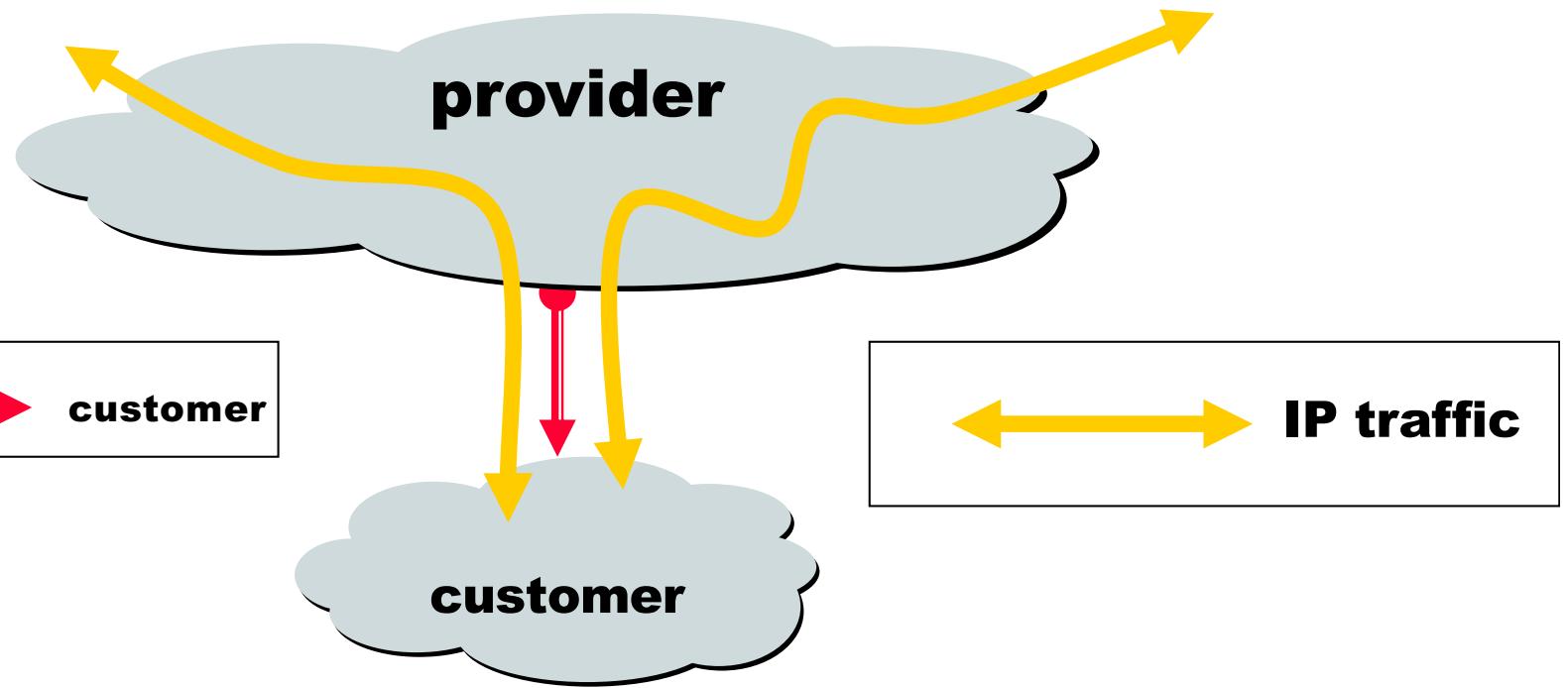
- Quando più organizzazioni si uniscono per formare una Inter-rete, occorre stabilire tra loro punti di collegamento o di **peering**



# Relazione fra Autonomous Systems

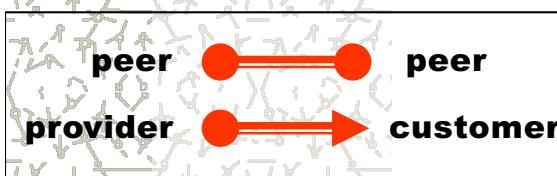
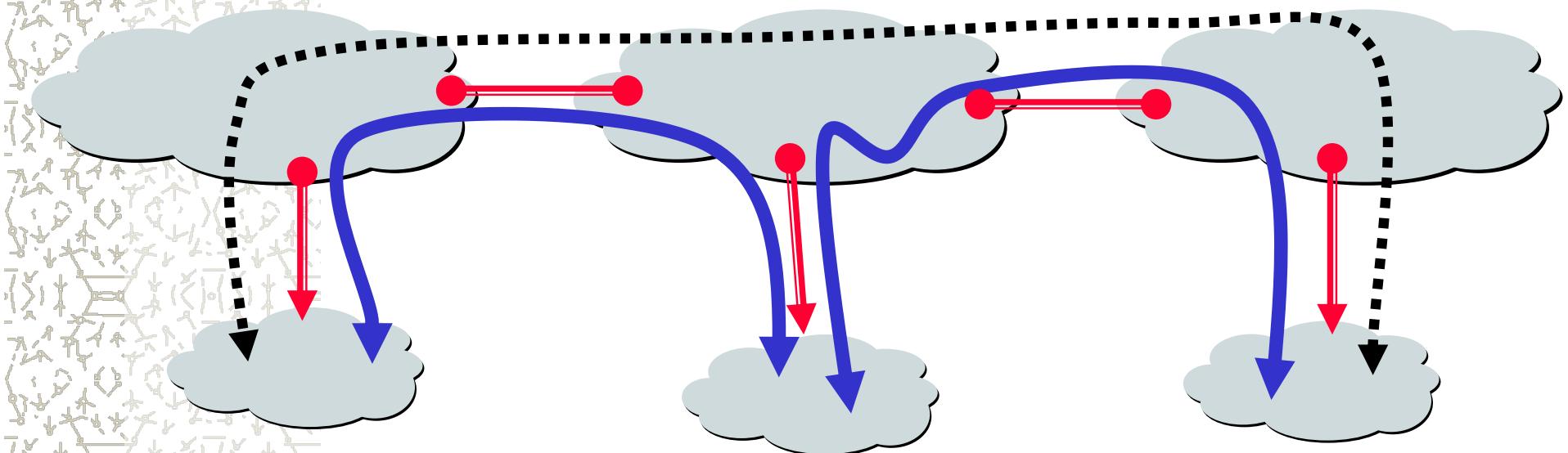


# Ruoli nel peering: Customers e Providers



**I Customer pagano i provider per accedere a Internet**

# Relazioni di Peering



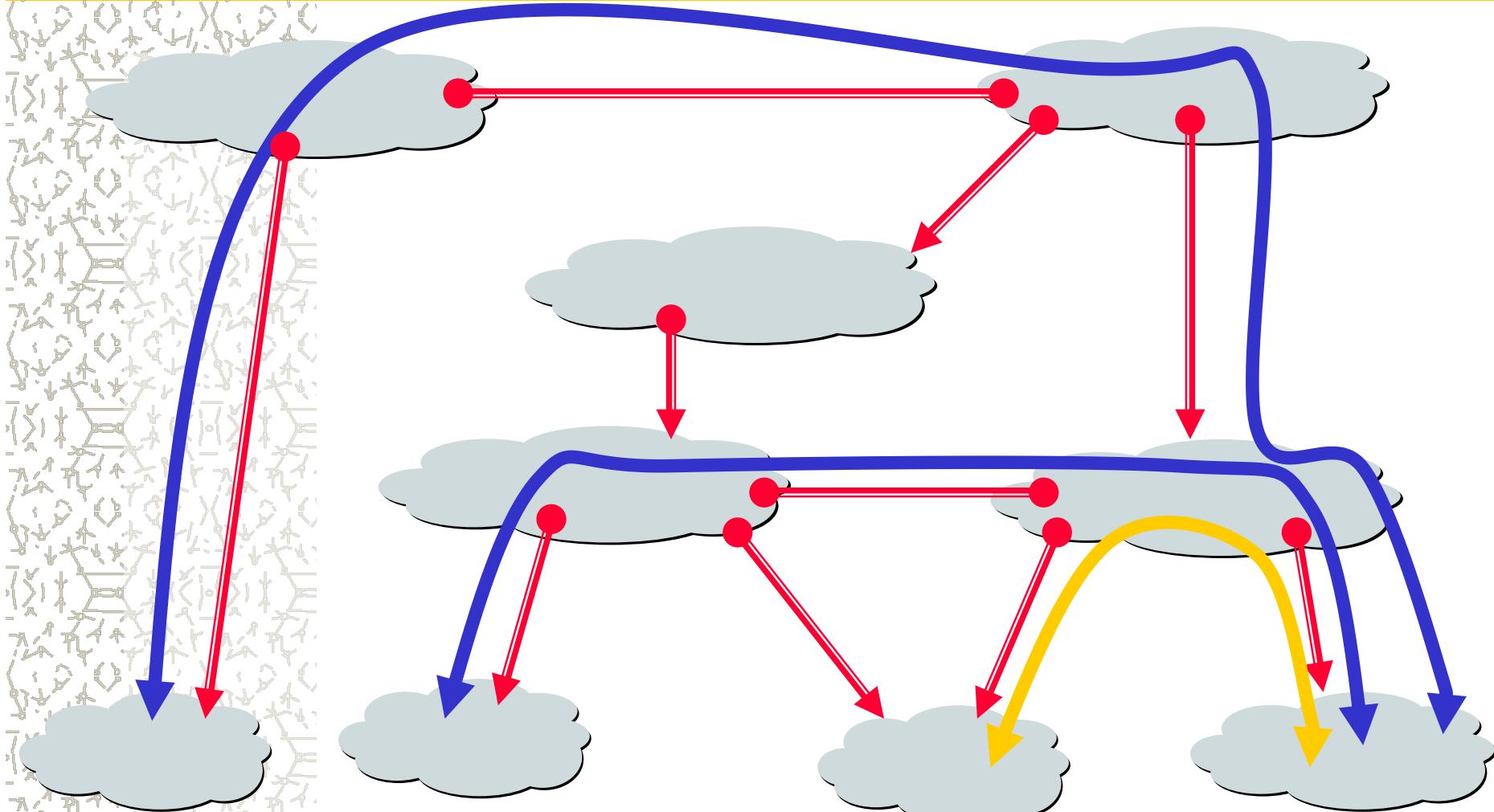
**traffic  
allowed**



**traffic NOT  
allowed**

**I provider di transito garantiscono il passaggio del traffico fra i rispettivi customers**

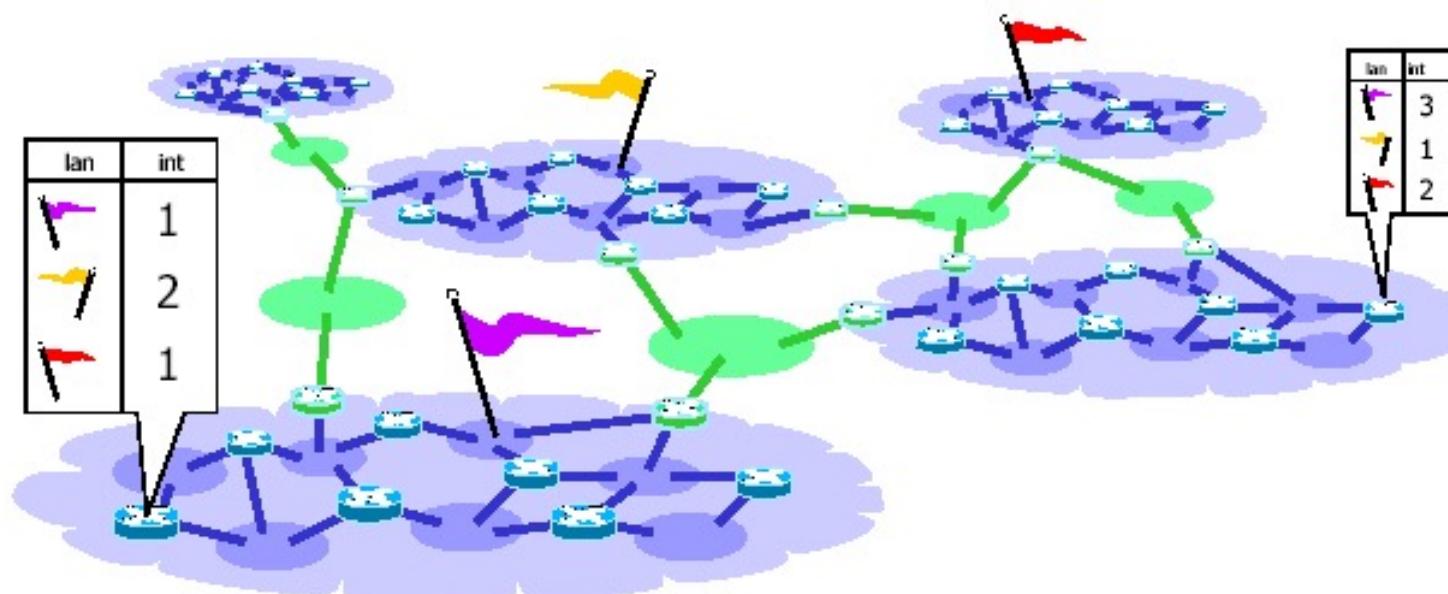
# Peering Per la connettività Tier1



**La pratica del Peering consente la connettività tra i “Tier 1” providers.**

# L'instradamento tra Sistemi Autonomi

- Ogni tabella deve avere un'entry per ogni possibile destinazione
- Questo deve valere sia per le destinazioni locali che per quelle globali



# Come aggiornare le tabelle di Instradamento?

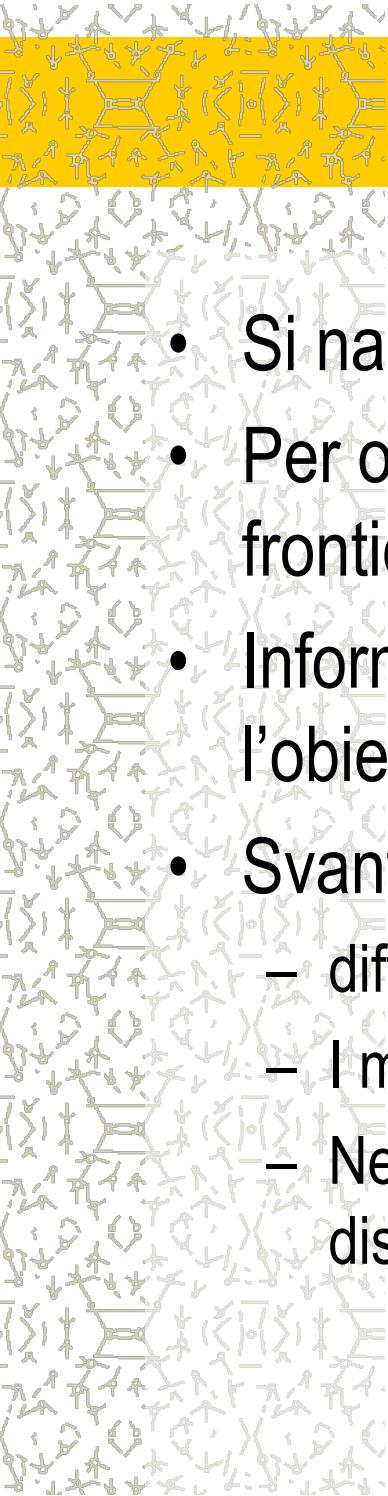
In generale vi sono tre opzioni:

1. Eseguire un unico algoritmo di instradamento tra organizzazioni adiacenti
2. Aggiornare le tabelle di instradamento manualmente aggiungendo percorsi statici predefiniti
3. Combinare un protocollo di instradamento intra-domain con un protocollo di instradamento inter-domain: Exterior gateway protocol



# 1. Unico algoritmo di Instradamento

- Molti Svantaggi:
  - Ritardo di propagazione, ex: distance vector
  - Rallentamento: messaggi di instradamento inviati agli altri routers con l'elenco delle possibili destinazioni
  - Tutte le organizzazioni sono forzate ad usare lo stesso algoritmo
  - Un nuovo algoritmo di instradamento è di difficile adozione
  - Non considera le relazioni politiche e commerciali tra sistemi autonomi

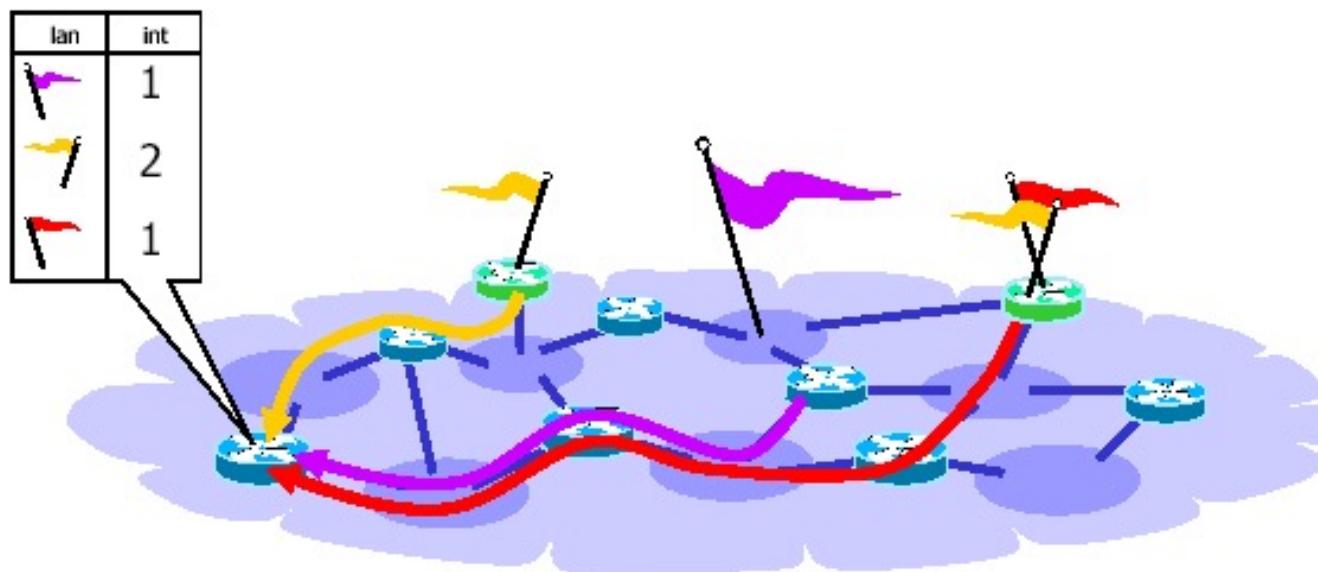


## 2. Percorsi statici

- Si nasconde la parte interna dell'AS
- Per ogni obiettivo esterno si identifica un router alla frontiera del Sistema Autonomo di destinazione
- Informazione sul cammino da seguire per raggiungere l'obiettivo
- Svantaggi:
  - difficile da aggiornare e da correggere
  - I malfunzionamenti non sono gestiti, non si ha backup
  - Nessuna garanzia che tutti i router del percorso sono in effetti disponibili per portare il traffico a destinazione

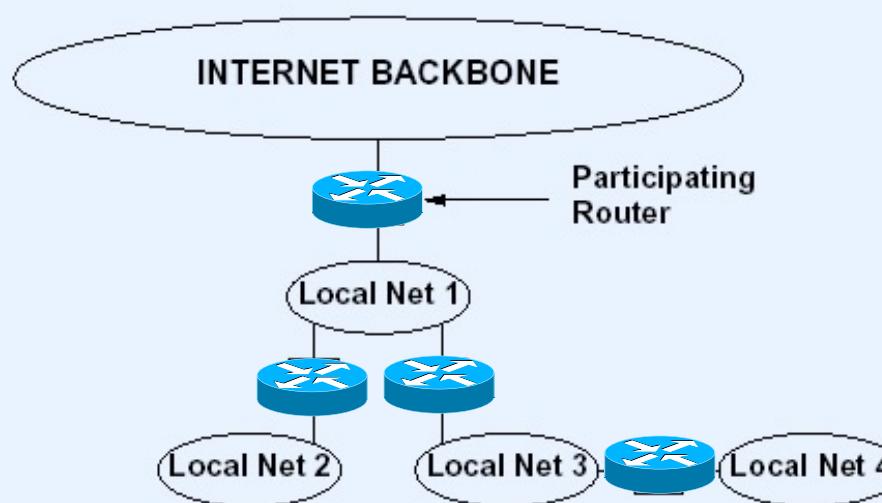
## 2. Percorsi statici

- L'algoritmo di instradamento diffonderà all'interno dell'AS il traffico locale come il traffico che segue i percorsi statici



# Le reti nascoste

- Ogni AS ha una topologia complessa, formata da diverse Reti Locali
- Non tutte le reti locali sono connesse ad un router di frontiera dell'AS
- Occorre informare l'esterno delle Reti Locali raggiungibili



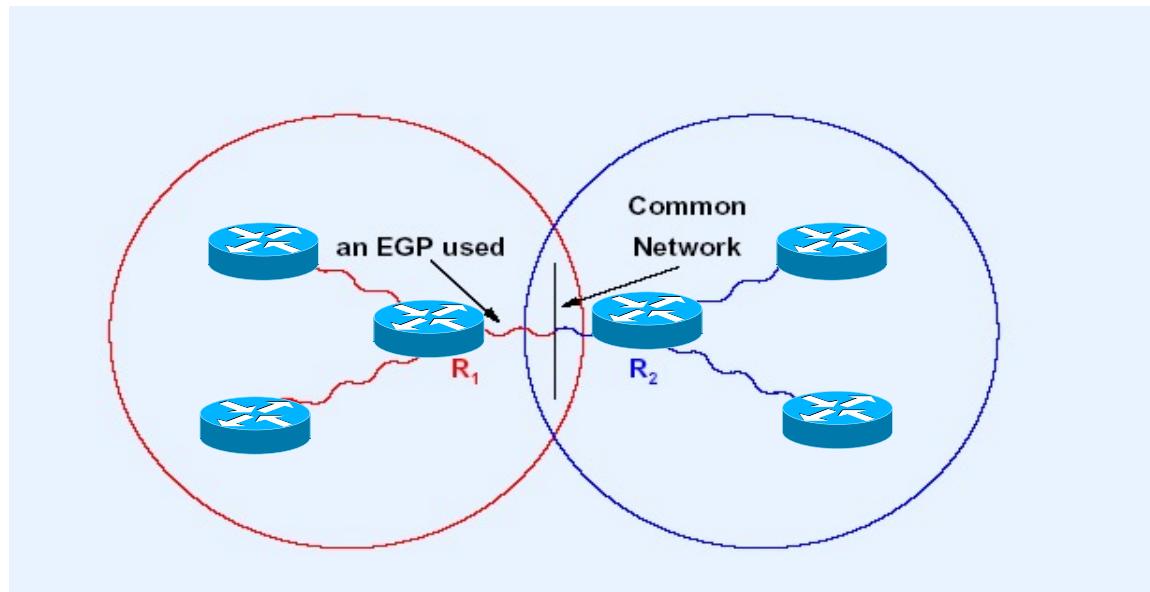


# Un approccio diverso

- Occorre avere un flusso informativo in due direzioni, sia dall'interno verso l'esterno che dall'esterno verso l'interno
- L'AS si deve far carico di garantire la consistenza degli instradamenti interni
- Occorre annunciare all'esterno quali reti interne sono raggiungibili
- Occorre assegnare le responsabilità per la diffusione delle informazioni riguardo l'instradamento

### 3. Exterior gateway protocol

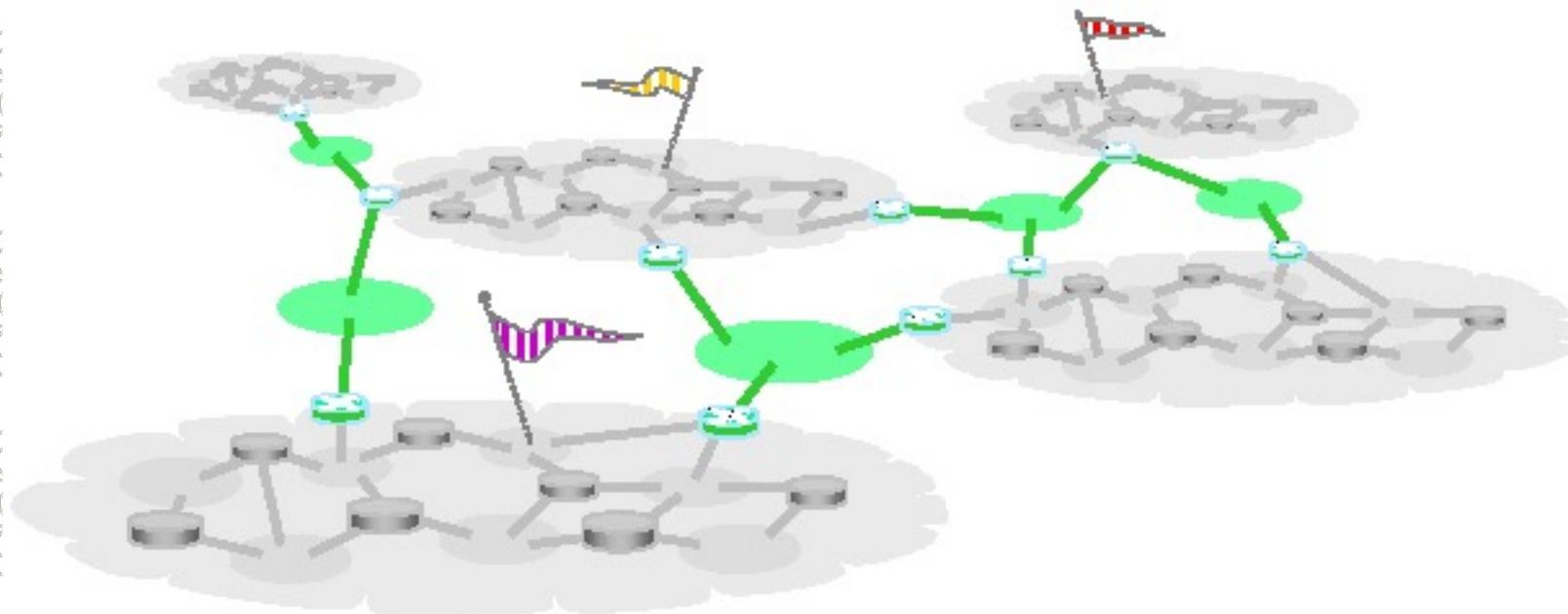
- Protocollo per lo scambio delle informazioni sull'instradamento tra Sistemi Autonomi
- BGP – Border Gateway Protocol
- Due AS che si scambiano informazioni di instradamento designano due router che stabiliscono una sessione di peering
- Router che partecipano a BGP sono detti Router di Confine o Gateway



# 3. Exterior Gateway Protocol

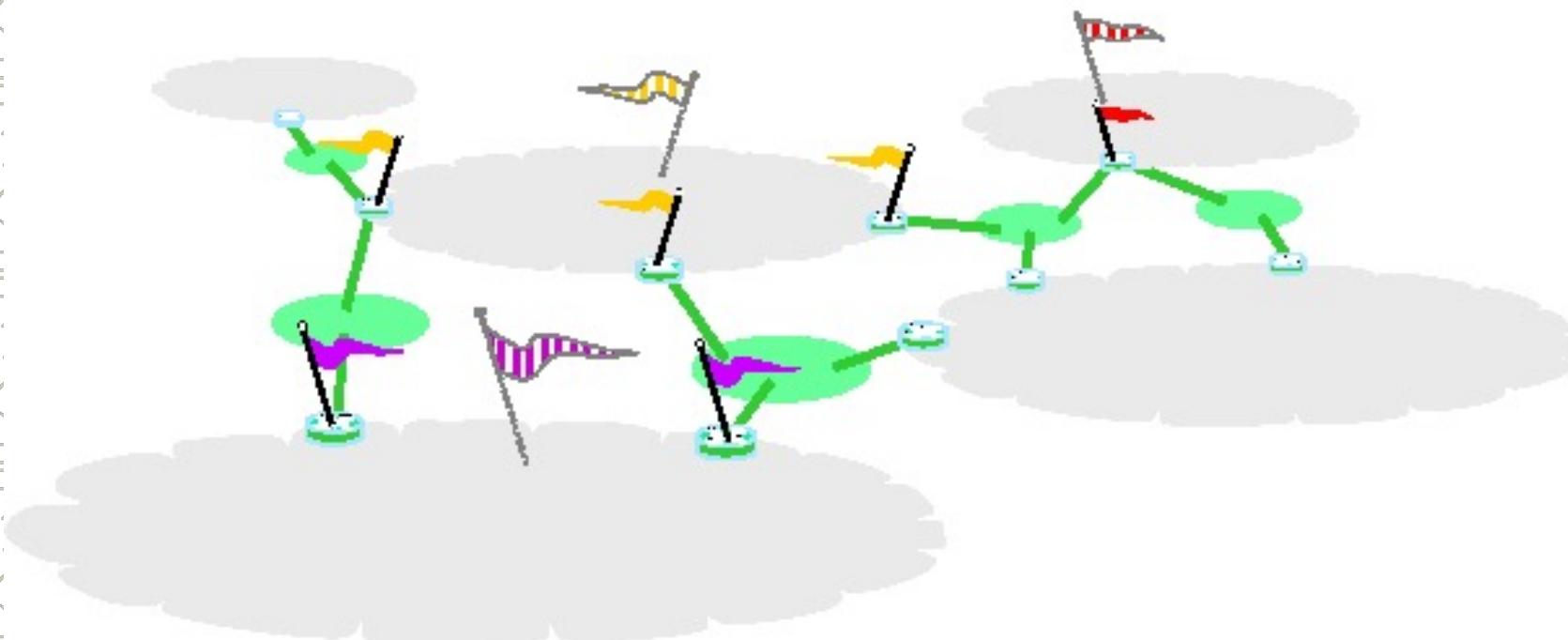
Approccio:

- Nascondi la parte interna degli AS
- Mantieni solo le zone di demarcazione e i router di frontiera degli AS



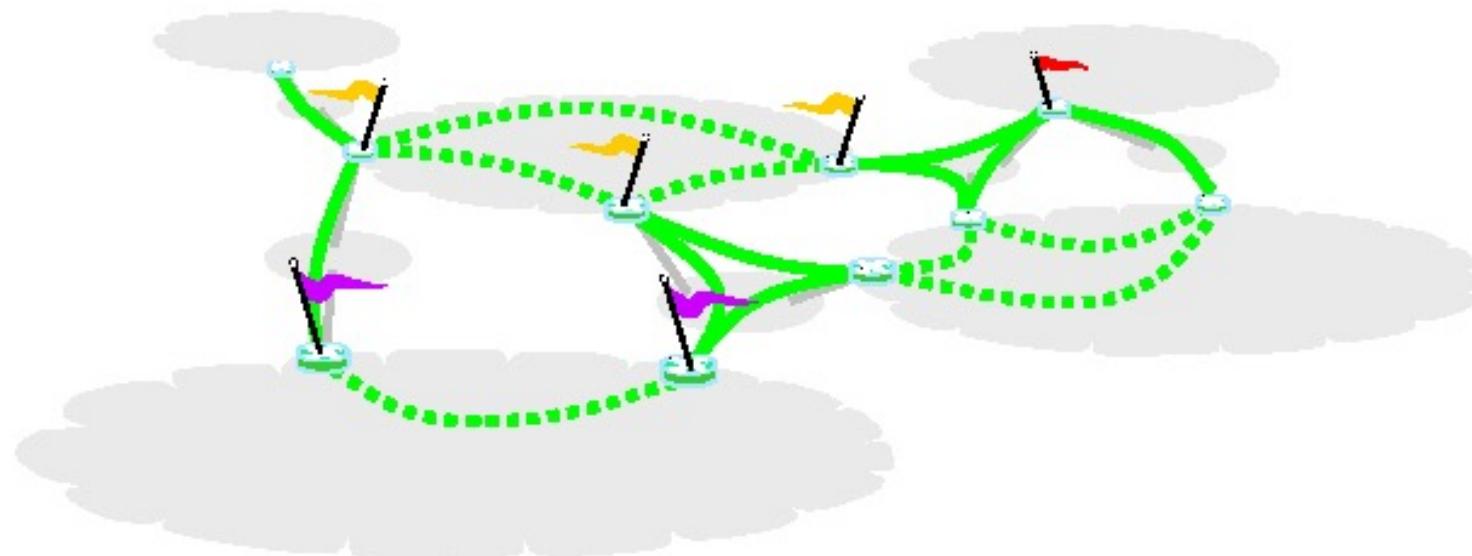
### 3. Exterior Gateway Protocol

- Ogni router di frontiera rappresenta le destinazioni interne come se fossero locali



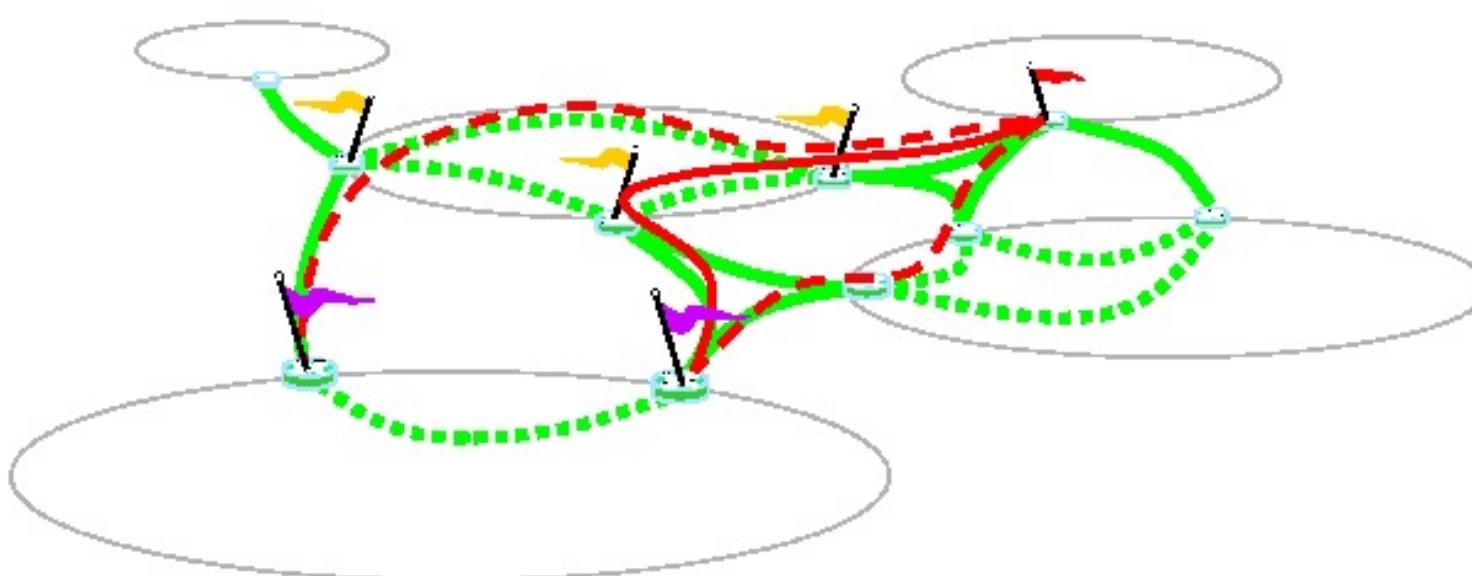
### 3. Exterior Gateway Protocol

- Semplifica il grafo considerando le informazioni sulla raggiungibilità sia interna che esterna all'AS
- Il grafo è gestito attraverso sessioni peering TCP



### 3. Exterior Gateway Protocol

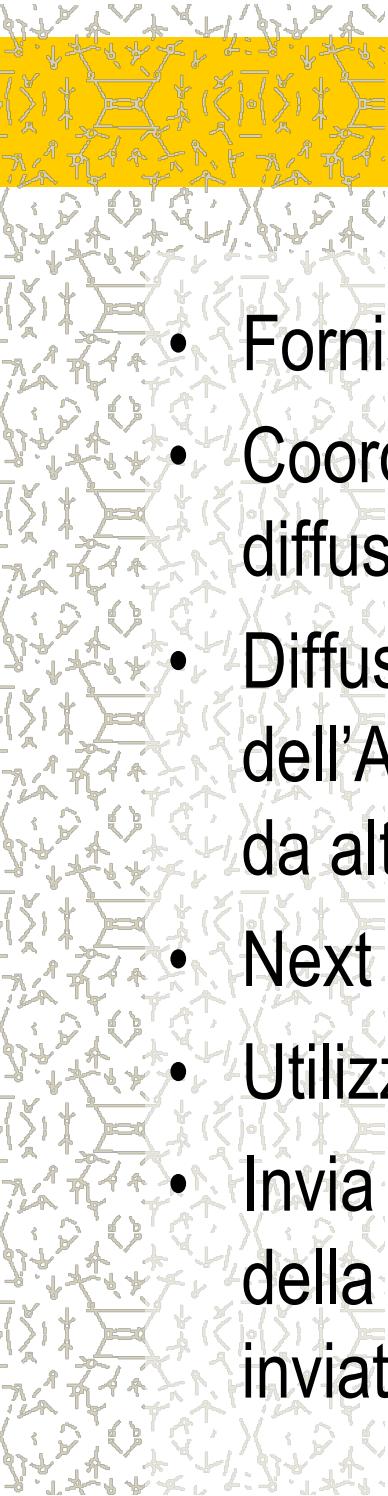
- Risolvi il problema dell'instradamento nel grafo così'
- Definisci anche percorsi prestabiliti sulla base di considerazioni politiche



# BGP v4 – Border Gateway Protocol

BGP mantiene aggiornate le tabelle di instradamento e propaga le informazioni sull'instradamento

BGP considera la disponibilità delle organizzazioni a cooperare nel processo di instradamento (accordi commerciali, questioni legali, preferenze locali)



# Caratteristiche di BGP i

- Fornisce comunicazione tra AS
- Coordinamento tra speaker di uno stesso AS -- diffusione di informazioni coerenti
- Diffusione dell'informazioni di raggiungibilità all'interno dell'AS e attraverso l'AS, e apprende tali informazioni da altri AS
- Next hop routing – simile a distance vector routing
- Utilizza TCP per le sessioni di peering
- Invia messaggi Keep-alive per informare dello stato della connessione anche se nessun messaggio è inviato



## Caratteristiche di BGP ii

- Informazioni sull'instradamento, router che saranno attraversati fino a destinazione
- Aggiornamenti incrementali per risparmiare banda
- Supporto CIDR – invio della maschera insieme all'indirizzo
- Aggregazione delle informazioni di instradamento per destinazione correlate
- Consente al destinatario di autenticare i messaggi

# Numerazione degli AS

- BGP richiede un numero identificativo per ogni AS (Autonomous System Number, asn) tra 1 and 65,535
- numeri maggiori di 64,511 sono detti “privati”
- Un asn può essere ottenuto da
  - asn globale – all'autorità internet regionale: ripe, arin, apnic
  - asn privato – all'isp

# Funzionalità BGP

1. Apertura connessione tra peers
2. Annuncio informazioni sulla raggiungibilità
3. Verifica corretto funzionamento

Quattro tipi di messaggio BGP

Type Code	Message Type	Description
1	OPEN	Initialize communication
2	UPDATE	Advertise or withdraw routes
3	NOTIFICATION	Response to an incorrect message
4	KEEPALIVE	Actively test peer connectivity

# Intestazione messaggi BGP

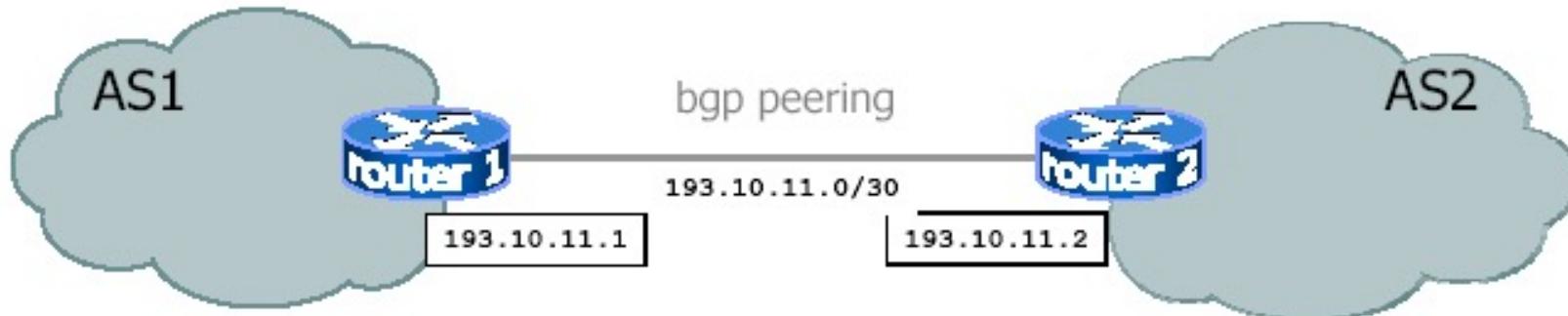
- Precede ogni messaggio BGP ed identifica il tipo di messaggio
- Marker (16 byte): scelto in accordo tra le due parti per sincronizzare i messaggi. Questa funzione non è fornita da TCP
- Length (2 byte): lunghezza del messaggio tra 19 e 4096 byte
- Type: tipo di messaggio BGP



# Peering tra due AS

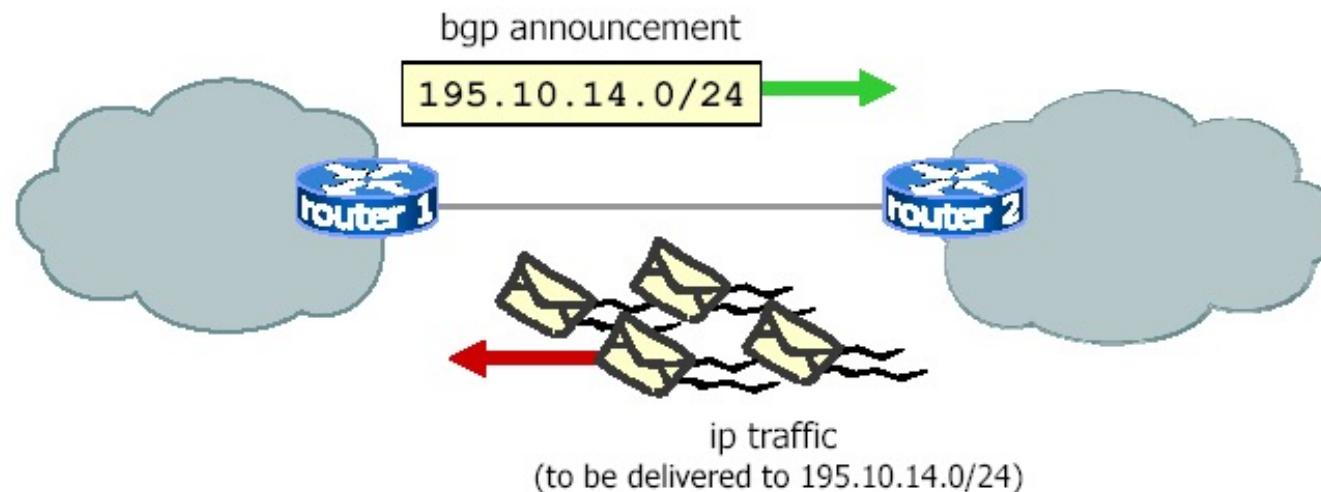
Le informazioni possono essere scambiate tra due AS solo se una sessione peering è attiva

La sessione peering è una connessione TCP tra i due AS



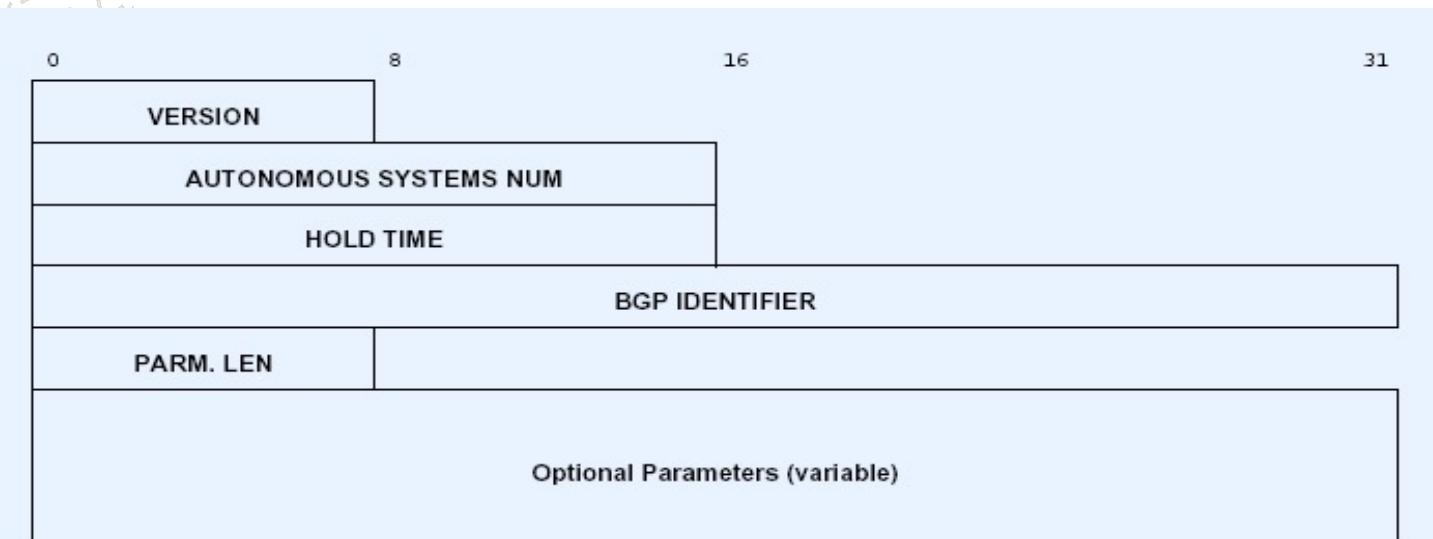
# Annunci BGP

- BGP permette ad un AS di offrire connettività ad un altro AS
- Offrire connettività significa promettere il recapito ad una specifica destinazione



# OPEN

- Utilizzato per aprire una connessione peer
- Il campo Hold specifica il massimo numero di secondi tra due messaggi successivi
- Un router bgp è caratterizzato dall'asn e da un identificatore unico a 32 bit che deve usare per tutte le connessioni peering
- Parametri opzionali: ad esempio per l'autenticazione

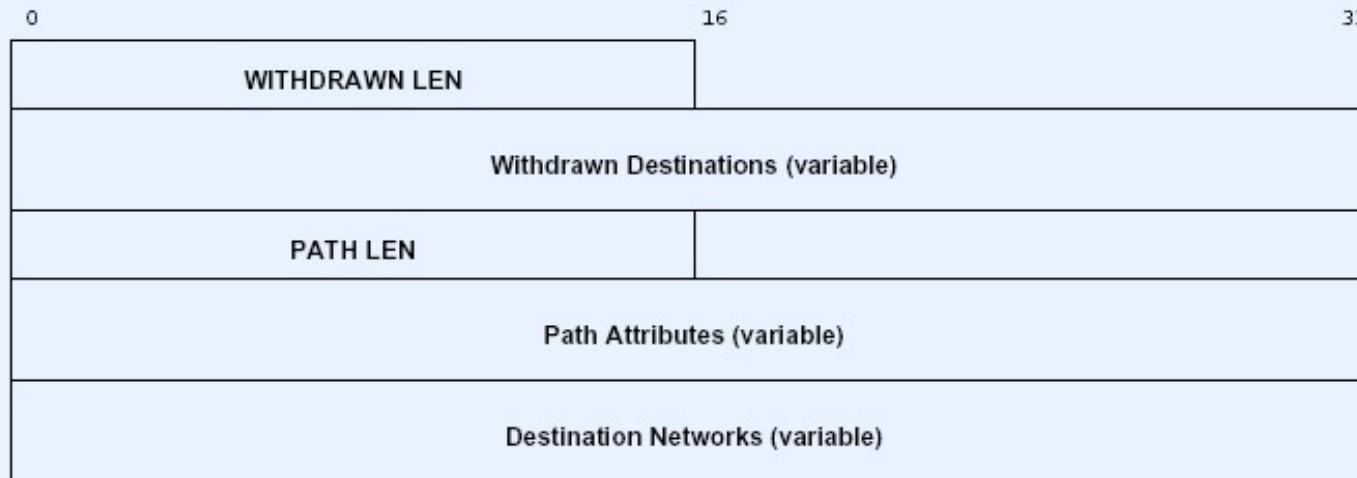


# OPEN

- Il router destinatario di un messaggio OPEN risponde con un KEEPALIVE
- Connessione aperta quando entrambi i router inviano un messaggio OPEN ed un messaggio KEEPALIVE

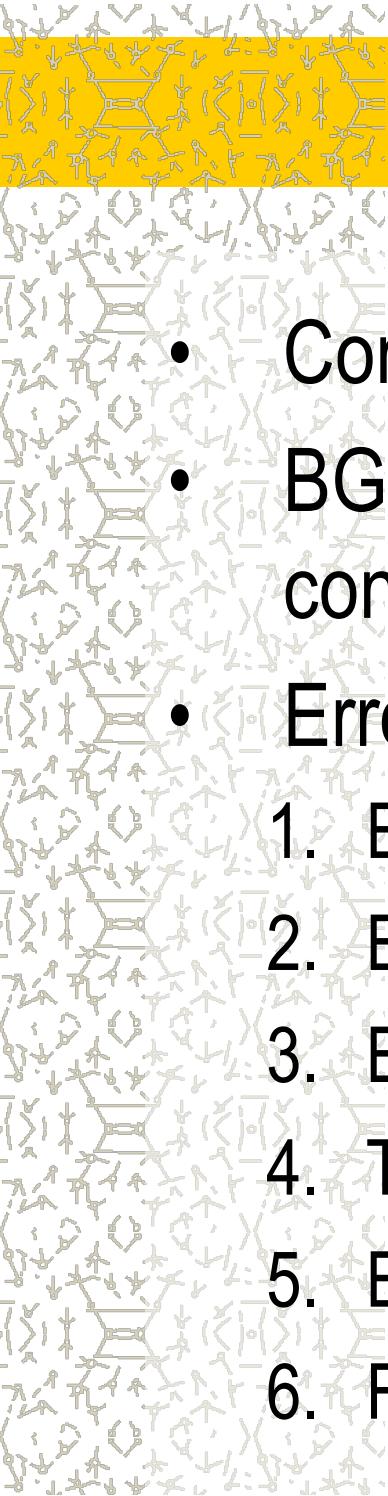
# UPDATE

- Announcement = prefix + attributes values
- Annuncia nuove reti raggiungibili ed eventualmente l'instradamento
- Annuncia reti precedentemente annunciate non più



# KEEPALIVE

- Verifica periodicamente la connessione TCP tra entità peer
- Più efficiente rispetto ad inviare periodicamente messaggi di instradamento
- Intervallo KEEPALIVE ogni 1/3 di HOLD time, mai inferiore a 1 sec.



# NOTIFICATION

- Controllo o segnalazione errori
- BGP invia un messaggio di notifica e chiude la connessione TCP
- Errori:
  1. Errore nell'intestazione del messaggio
  2. Errore nel messaggio OPEN
  3. Errore nel messaggio UPDATE
  4. Timer di attesa scaduto
  5. Errore nella macchina a stati finiti
  6. Fine (connessione terminata)

# Prefissi di Rete Compressi

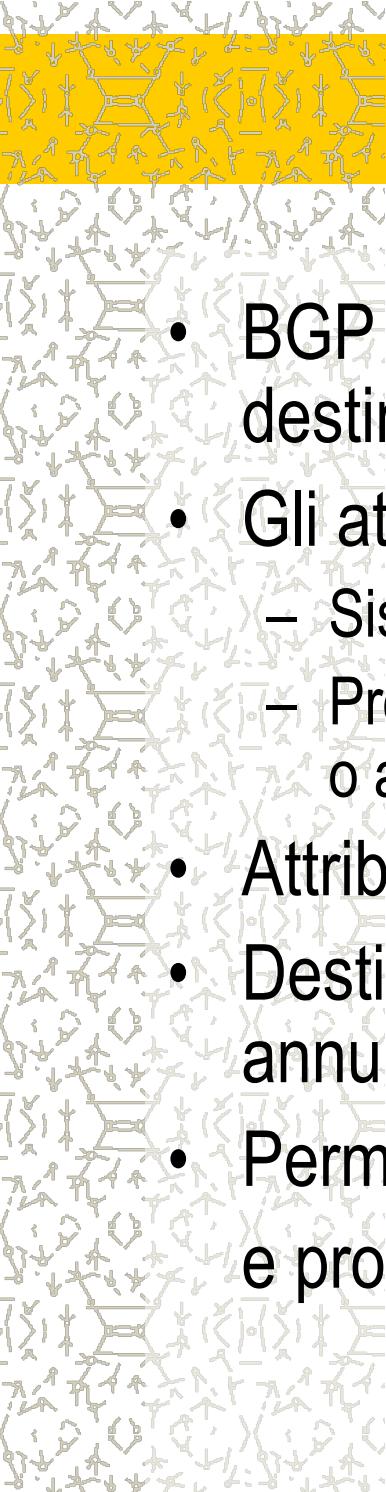
- Specifica solo i bytes ,corrispondenti al prefisso
- 1 – 4 byte: maschere fino a 8,16,24 bit
- Ex: 220.123

220.16.128



# Filtraggio degli annunci

- Gli annunci sono inviati e/o accettati solo se alcune condizioni sono verificate
- Gli annunci possono essere filtrati sulla base di:
  - Una lista di prefissi validi
  - Una lista di numeri di AS



# Path attributes

- BGP specifica più di un salto successivo verso la destinazione
- Gli attributi possono indicare:
  - Sistemi autonomi attraversati verso la destinazione
  - Provenienza delle informazioni sull'instradamento: locali (igp) o apprese da altri sistemi autonomi (egp)
- Attributi sono comuni a tutte le destinazioni annunciate
- Destinazioni con attributi diversi devono essere annunciate con messaggi diversi
- Permette di individuare cicli sugli instradamenti e provenienza dei messaggi

# Path attributes

- Codice tipo:
  1. Origine informazione instradamento
  2. Elenco sistemi autonomi sul percorso
  3. Salto successivo
  4. Discriminazione tra più punti di uscita all'AS
  5. Preferenza all'interno dell'AS
  6. Indicazione di percorsi riuniti
  7. ID dell'AS che ha riunito i percorsi

# Route Selection basata su Path Attributes

**Highest Local Preference**

Enforce relationships

**Shortest AS PATH**

**Lowest MED**

**i-BGP < e-BGP**

**Lowest IGP cost  
to BGP egress**

traffic engineering

**Lowest router ID**

Throw up hands and  
break ties

# Uso degli Attributi BGP

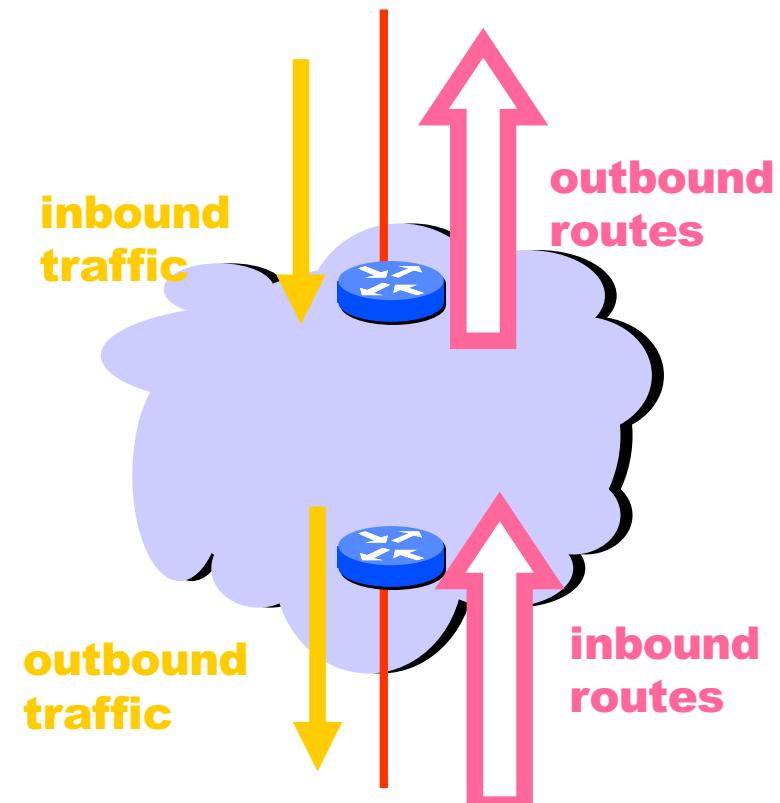
- In ingresso

- Utilizzabili per il Filtraggio delle routes in ingresso
- La manipolazione di attributi sulle routes uscenti consente **di tentare** di influenzare la selezione dei percorsi dagli altri AS verso l'interno

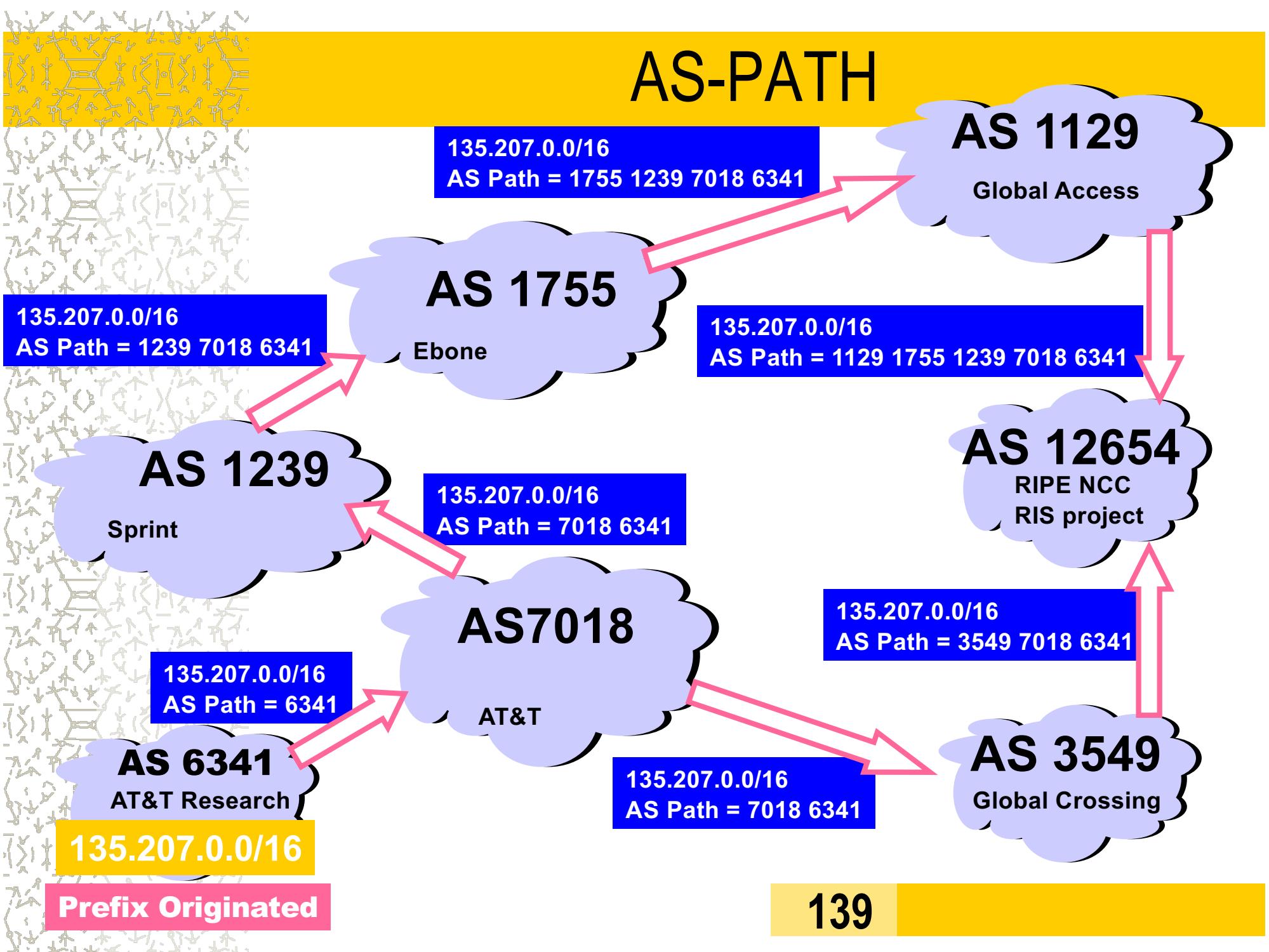
- In Uscita

- Utilizzabili per il Filtraggio delle routes in uscita
- La manipolazione di attributi sulle routes entranti consente **di influenzare** la selezione dei percorsi verso l'esterno

In general, an AS has more control over outbound traffic



# AS-PATH

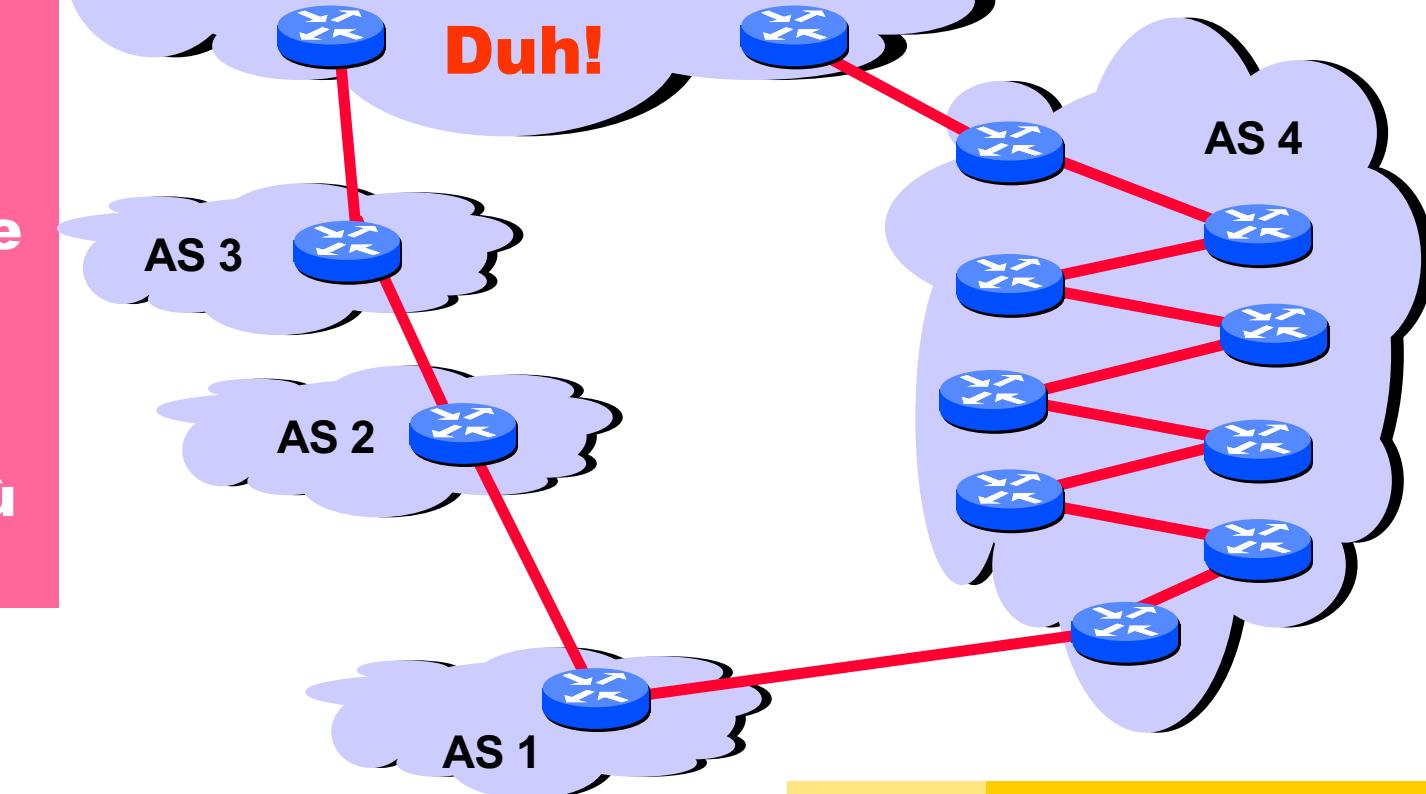


# Equivoci sul percorso minimo

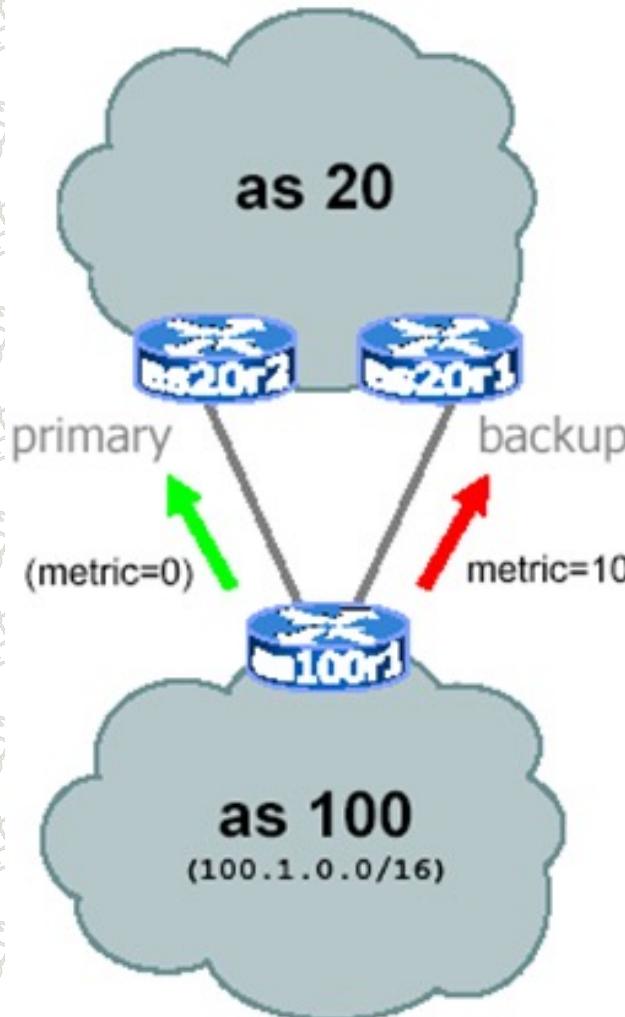
**Il path più  
corto non  
corrisponde  
necessariame-  
nte al  
percorso di  
rete  
realmente più  
breve**

**Per BGP il  
path 4 1 è meglio  
del path 3 2 1**

# Duh!

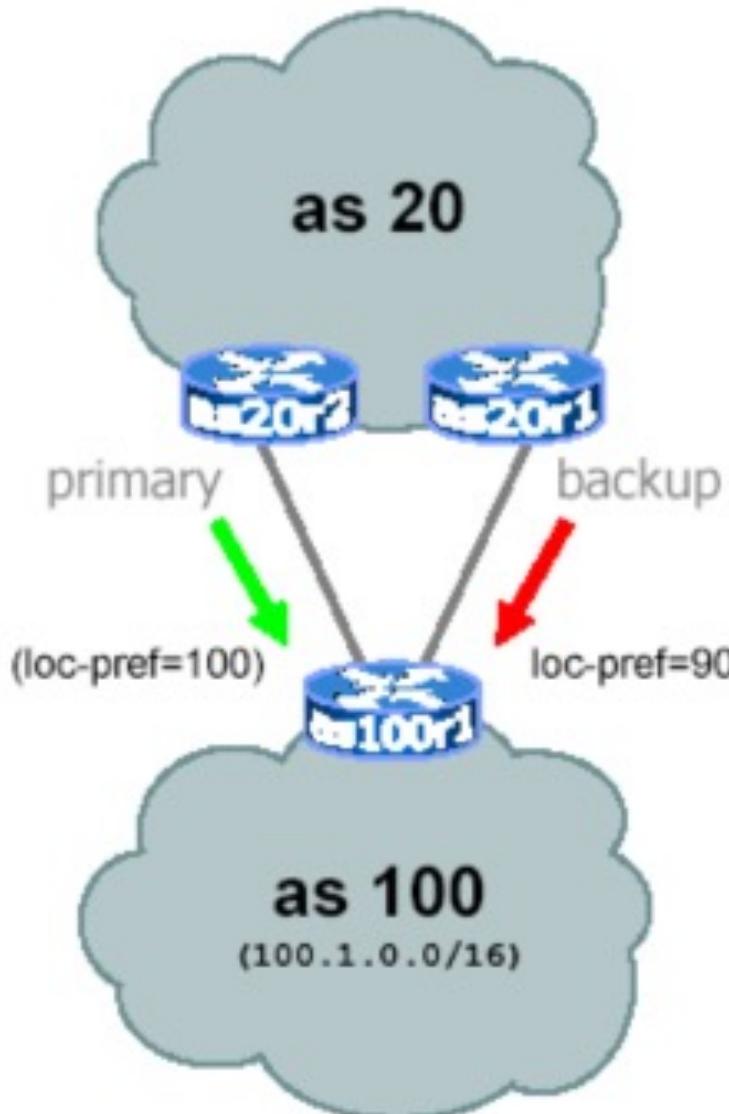


# Metric o MED



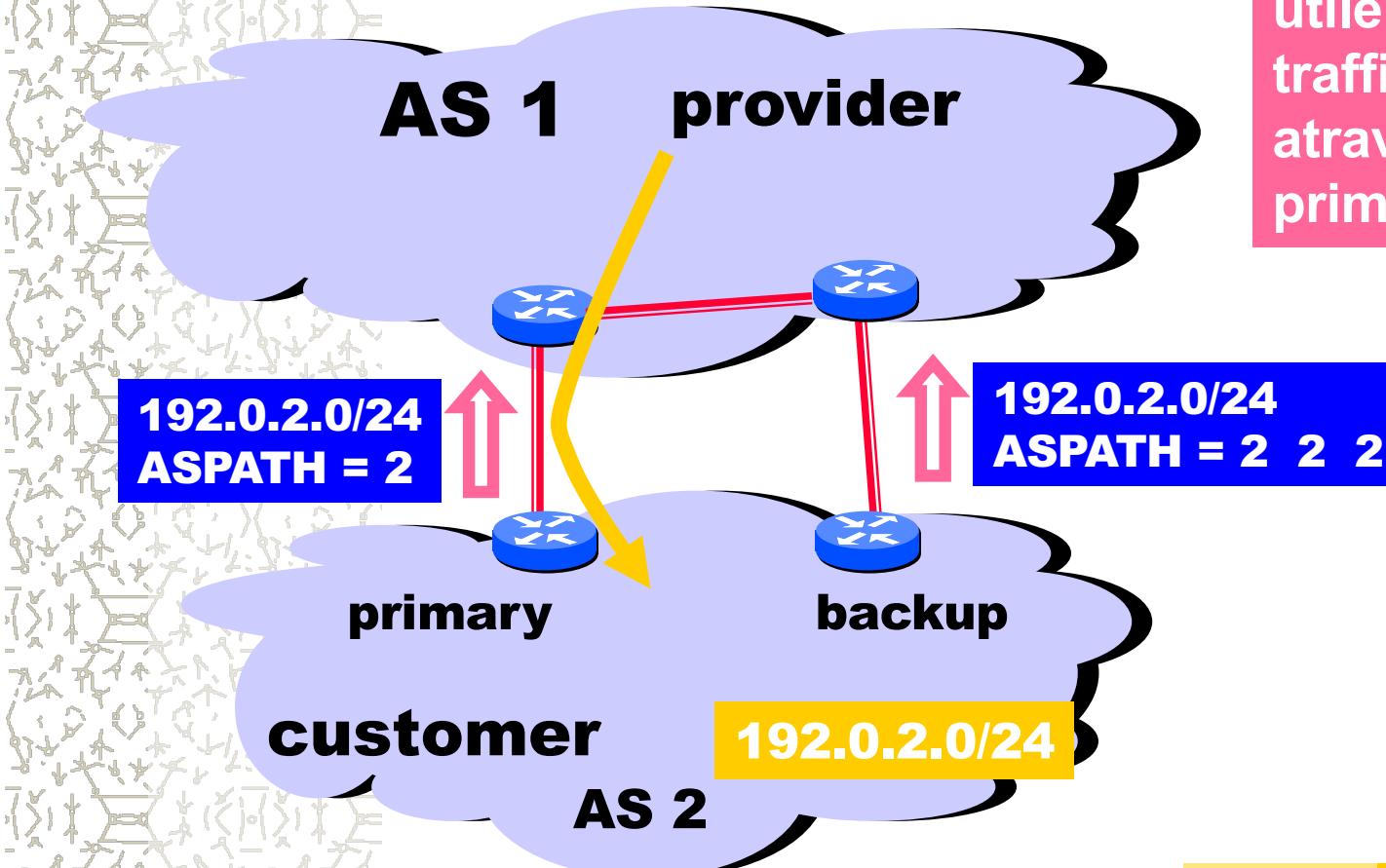
- L'attributo noto come «metrica» è anche chiamato «multi-exit-discriminator» (MED)
- L'AS ricevente in presenza di 2 MED differenti dovrebbe adottare la route associata a quello più piccolo
- Il valore di default del MED è 0
- È usato sugli annunci in uscita per condizionare il comportamento del traffico in ingresso

# Local Preference



- L'attributo noto come «local preference» è associabile agli annunci in ingresso ed è utile per gestire politiche di scelta sul traffico in uscita
- Nell'algoritmo di decisione BGP è verificato prima della lunghezza dell'AS-PATH
- Il valore di default è 100
- Valori più elevati sono preferibili

# AS path prepend

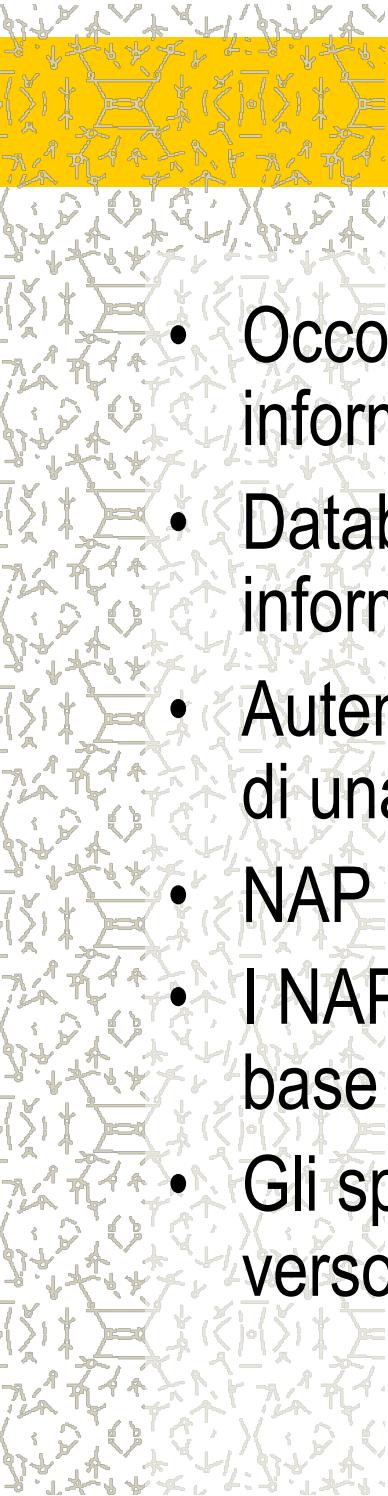


Il Padding del path con AS numbers ripetuti può essere utile per forzare il traffico da AS 1 attraverso il link primario



# Limiti di BGP

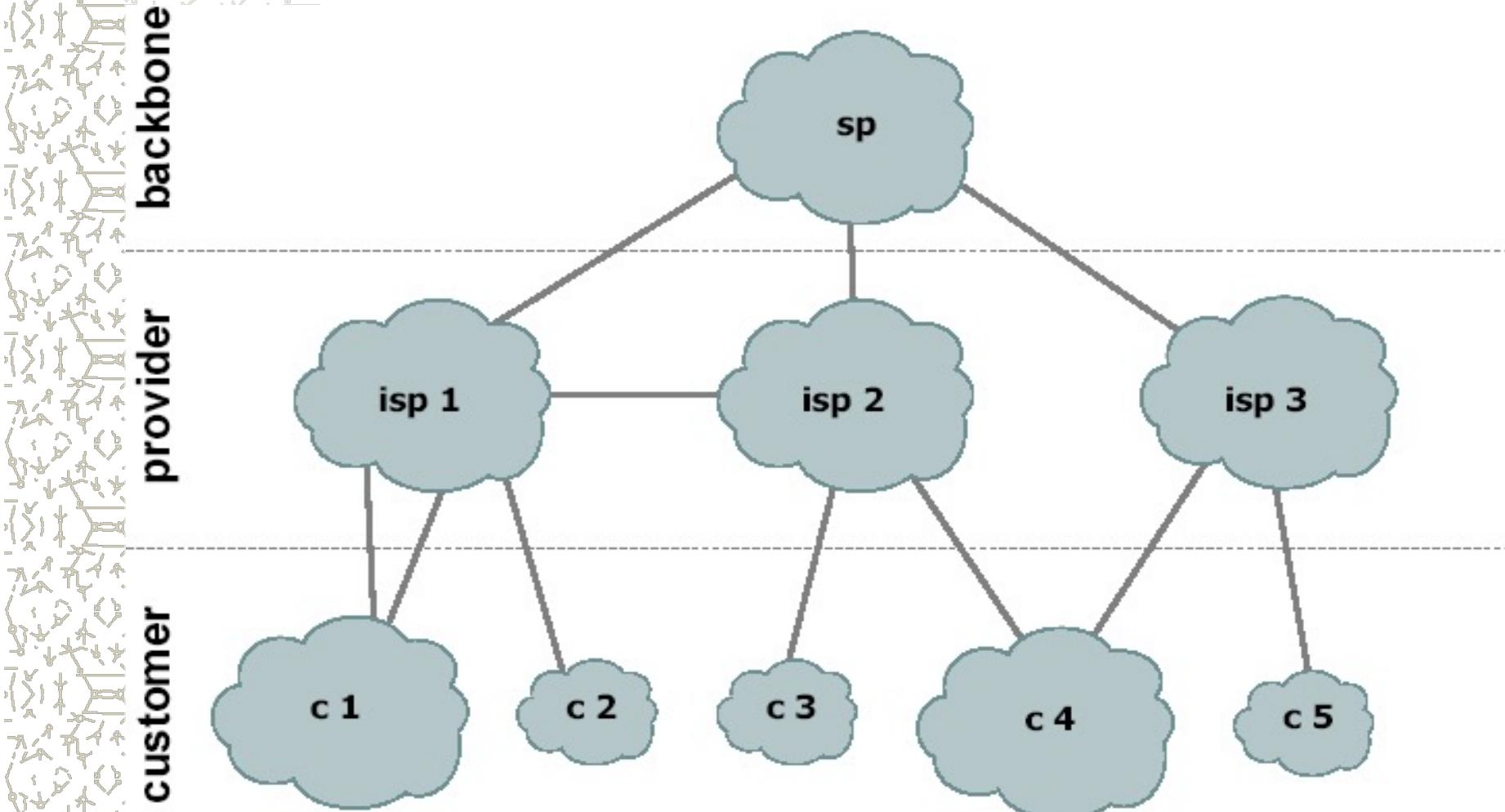
- BGP non può discriminare tra due percorsi sulla base della distanza o della congestione
- L'informazione di raggiungibilità fornita da due AS è indistinguibile
- BGP sceglie uno dei due percorsi possibili non sulla base di una metrica di costo
- BGP permette di suddividere il carico attraverso la rete ma non in modo dinamico
- Occorre configurare manualmente quale reti sono annunciate da quali routers esterni
- Tutti i sistemi autonomi devo concordare su uno schema coerente per annunciare la raggiungibilità

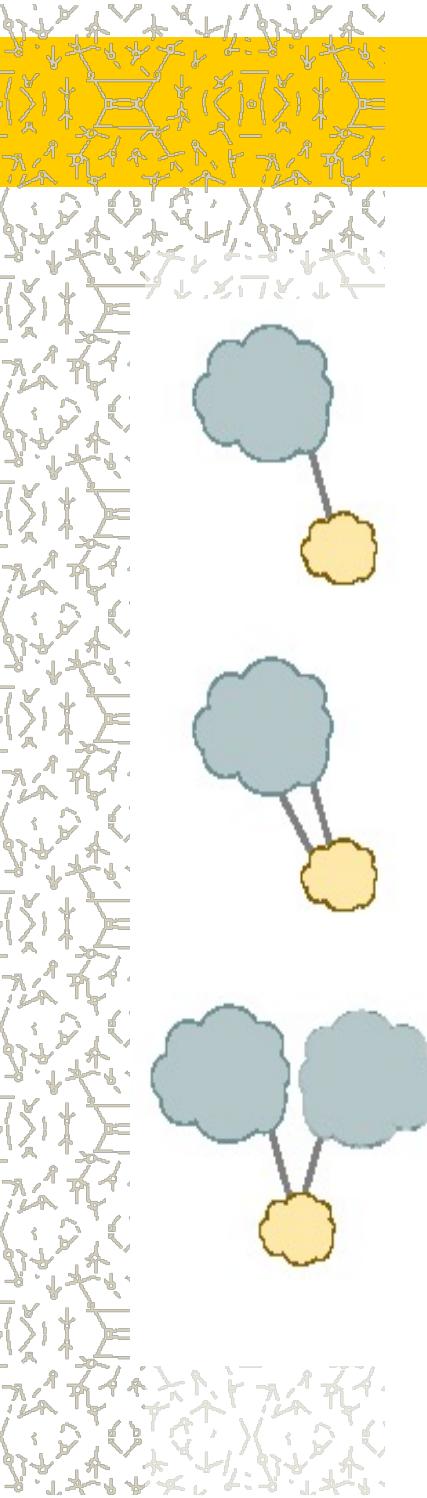


# Instradamento con arbitraggio

- Occorre un sistema per garantire la coerenza sulle informazioni di instradamento
- Database autenticato e replicato che contiene le informazioni sulla raggiungibilità
  - Autenticazione permette di annunciare la raggiungibilità di una rete solo al SA che la possiede
- NAP sono i router di interconnessione tra ISP
- I NAP hanno un Router Server che mantiene il database BGP ma non sono necessariamente speaker BGP
- Gli speaker BGP mantengono aperto un collegamento verso il Router Server

# Uno scenario BGP complesso

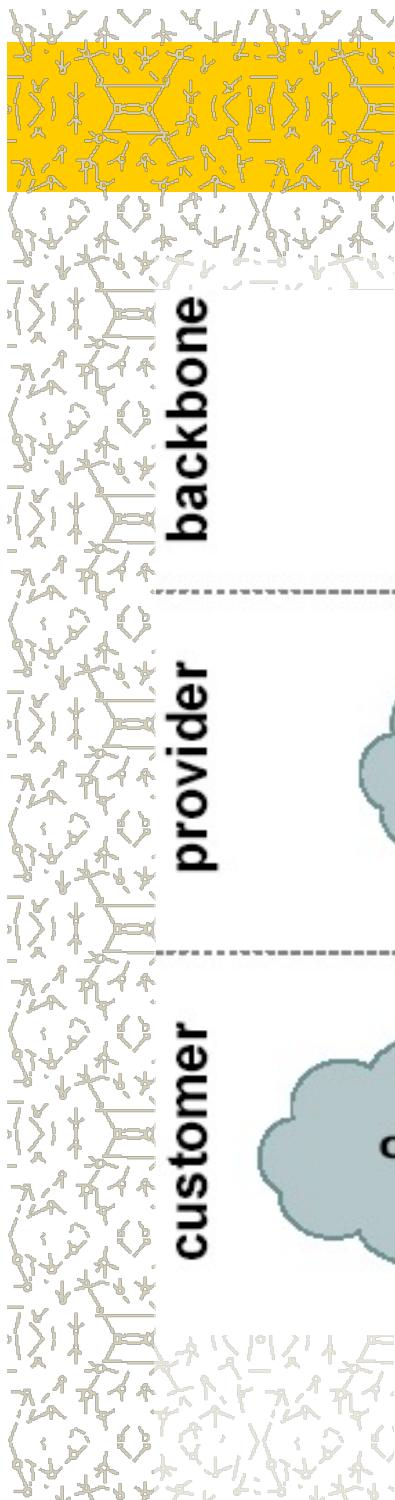




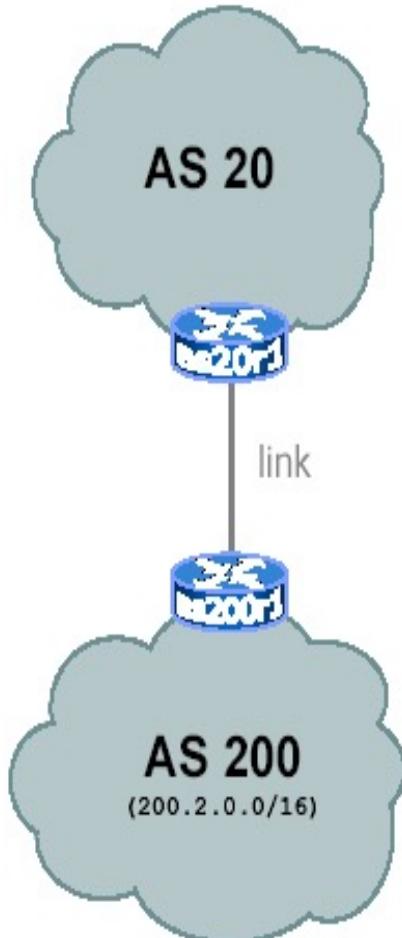
# Classificazione delle Reti

- Stub network:  
un collegamento ad un singolo isp
- Multi-homed stub network:  
due o più collegamenti allo stesso isp  
backup o divisione del carico
- Multi-homed network:  
due o più collegamenti a isp differenti  
backup o divisione del carico

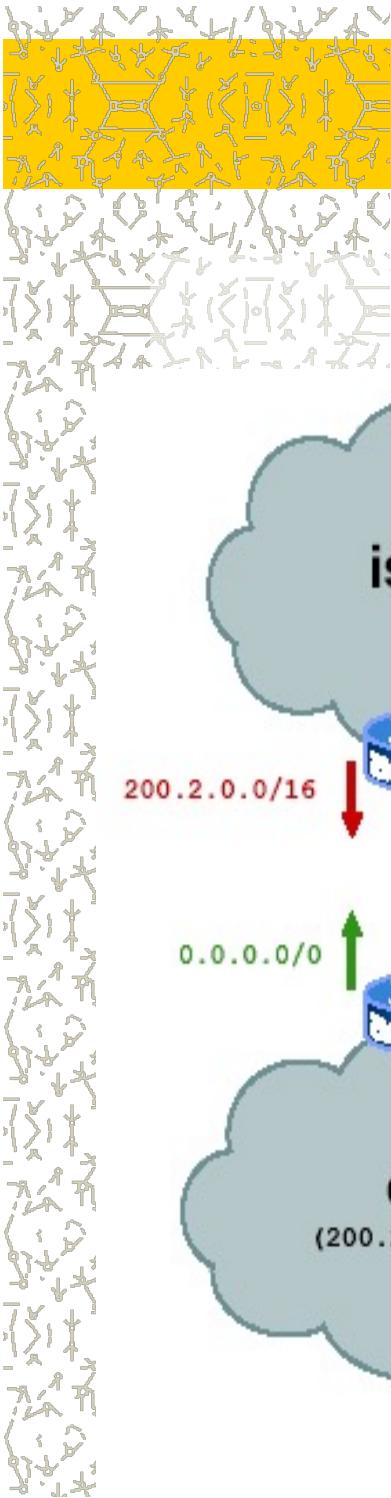
# Stub network



# Stub network, architettura

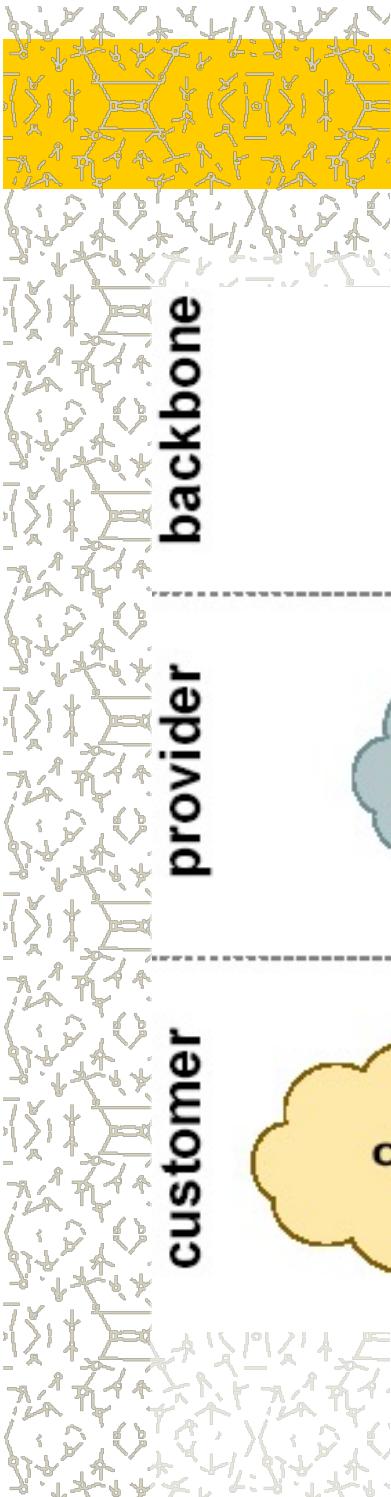


- Un router della rete è scelto come gateway di default è connesso ad un singolo router dell'isp con una o più connessioni
- Una singola sessione di peering in cui as200 annuncia la sua raggiungibilità e accetta l'instradamento di default sul router

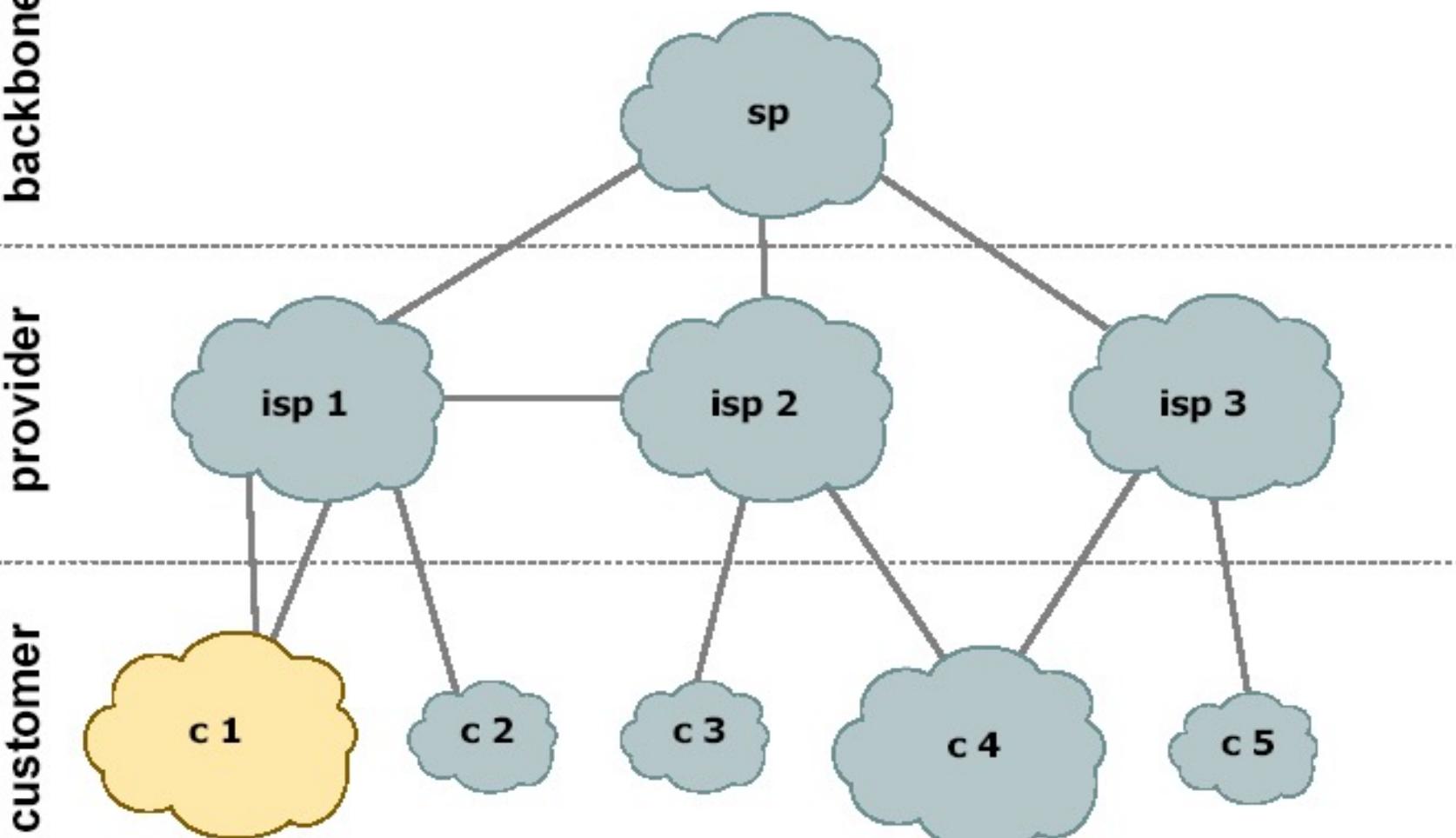


# Instradamento statico per stub network

- Un instradamento statico di default è sufficiente per i pacchetti in uscita per essere inviati su internet attraverso la connessione all'isp
- Un instradamento statico è anche sufficiente per i pacchetti in ingresso per raggiungere la rete attraverso la connessione all'isp
- Non vi è alcun bisogno di BGP

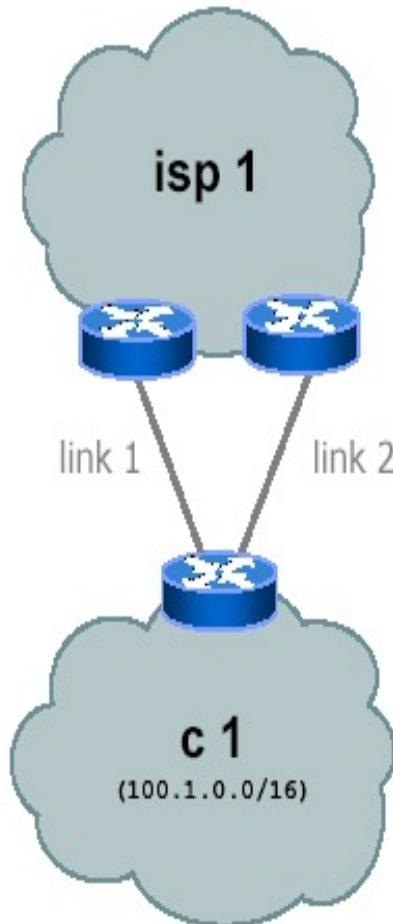


# Multi-homed stub networks



# Multi-homed stub networks

- Due collegamenti allo stesso isp
- Due routers della rete customer sono di solito coinvolti



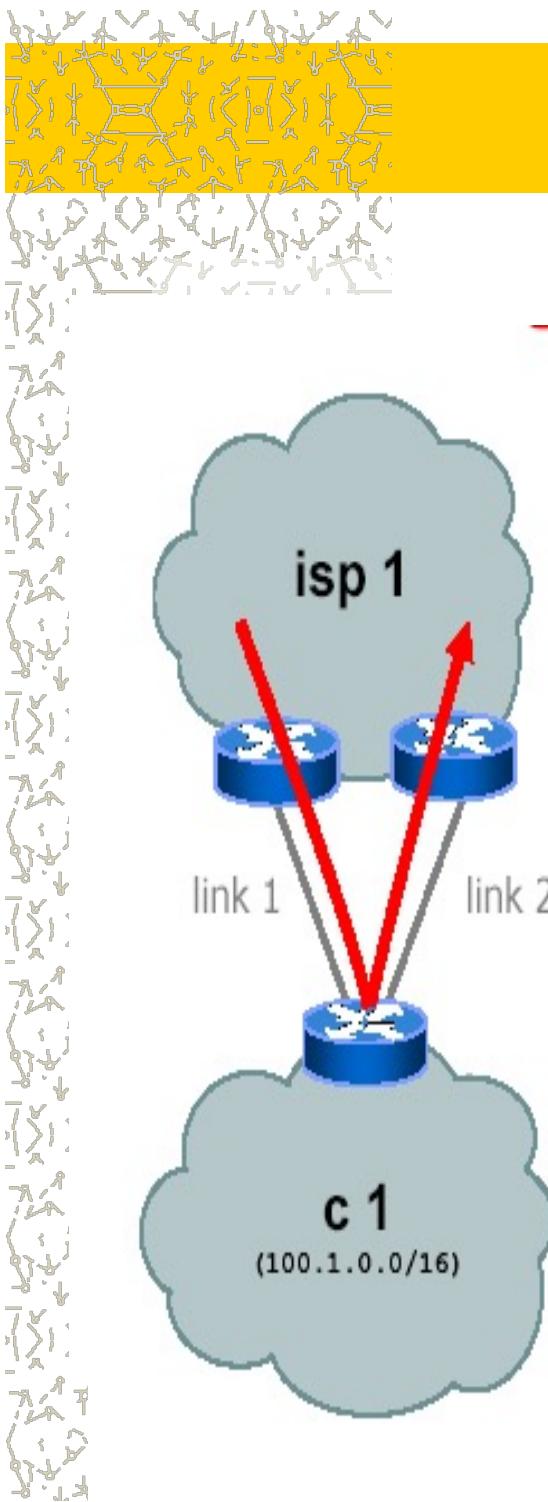
single point  
of failure



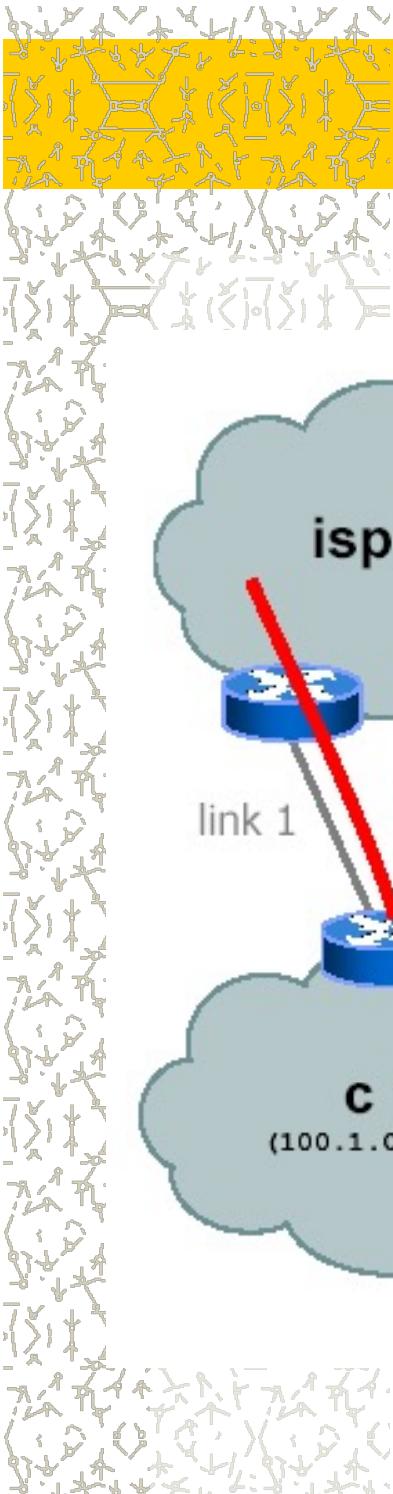
augmented  
redundancy



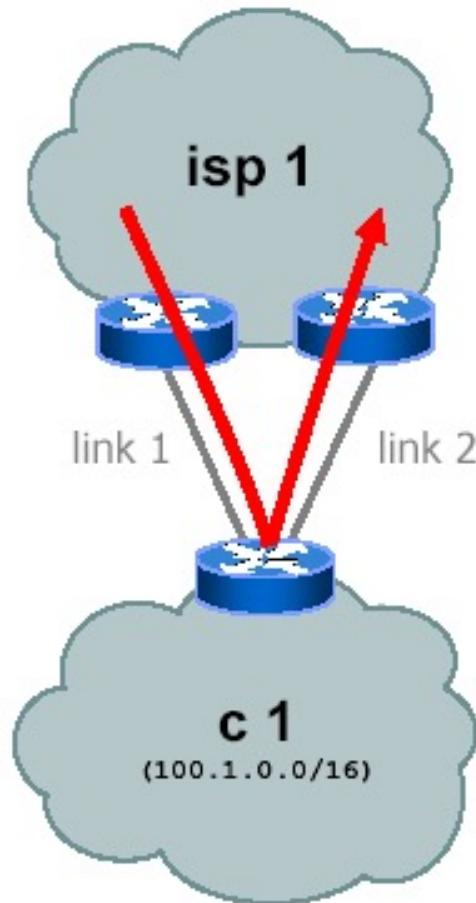
# Instradamento



- Un pacchetto diretto ad Internet può attraversare uno dei due link
- Un pacchetto proveniente da Internet può attraversare uno dei due link
- Un pacchetto in transito può attraversare entrambi i link

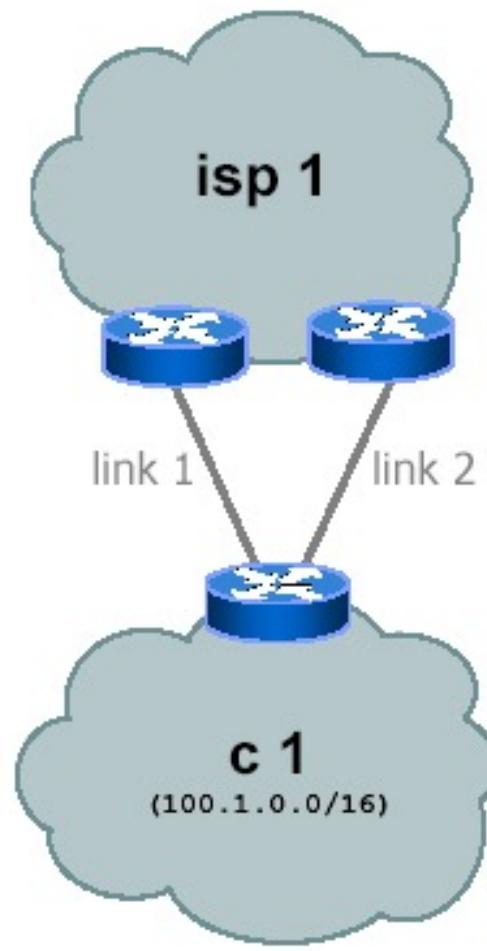


# Politiche desiderate - Backup



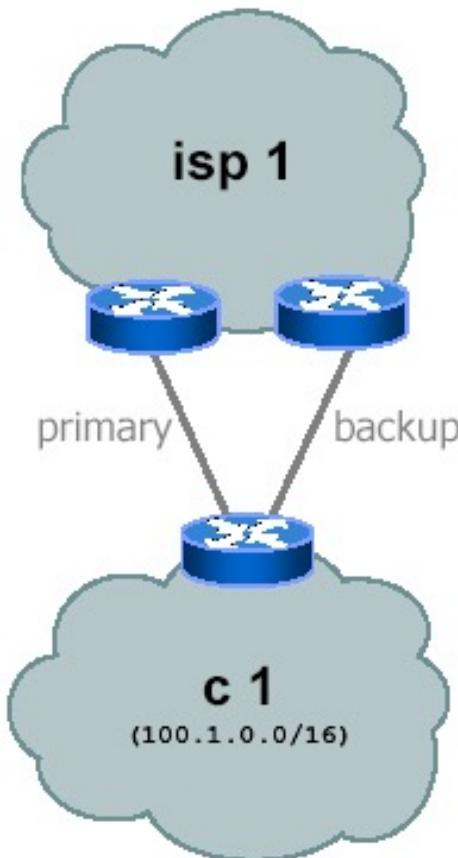
- Eliminare traffico in transito
- Traffico in ingresso:
  - Utilizzare link 1
  - Utilizzare link 2 in caso di fault su link 1
- Traffico in uscita:
  - Utilizzare link 1
  - Utililizzare link2 in caso di fault su link 1

# Alternative a BGP



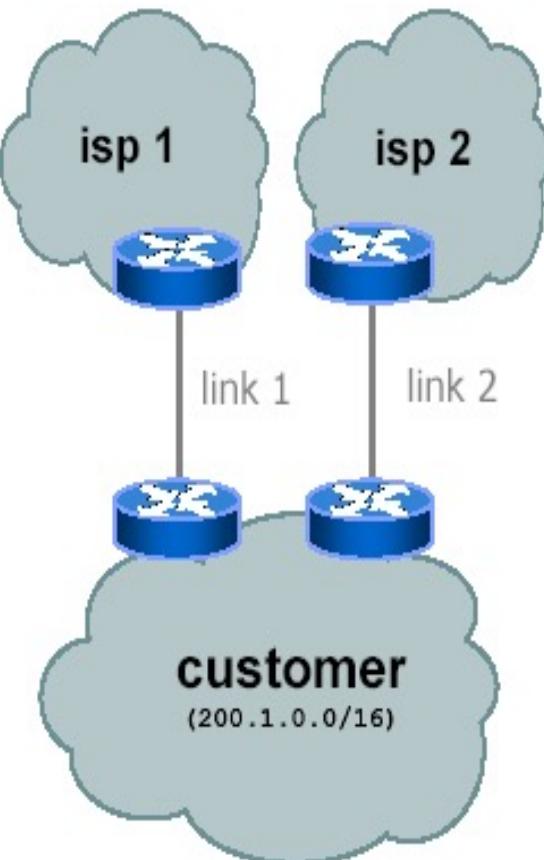
- Usare un igp:
  - Pacchetti usano link 1 o link 2 a seconda dello shortest path verso c1
  - Non è possibile escludere pacchetti in transito quando link 1 e link 2 sono sul cammino minimo tra sorgente e destinazione
- Usare cammini statici:
  - I routers dell'isp e la rete devono essere configurati manualmente in modo coerente.
  - Non è possibile gestire un meccanismo di backup automatico

# La strategia usata da BGP

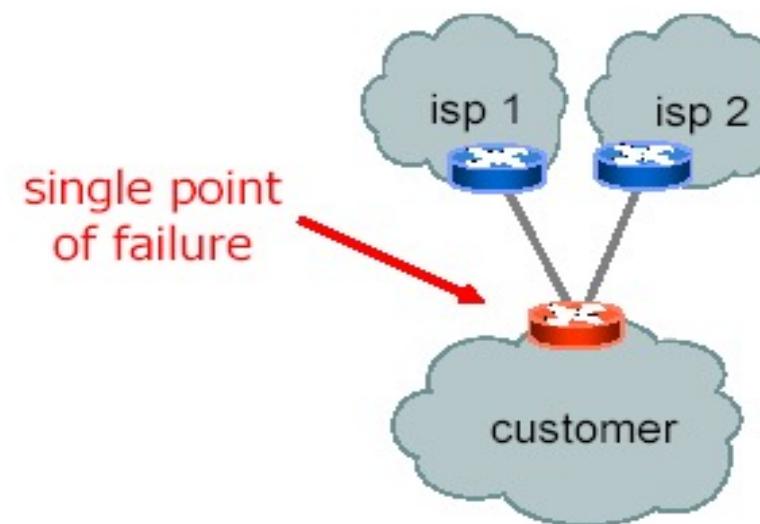


- Annuncio /16 aggregato su ogni arco:
  - Link primario invia un announcement standard
  - Il link di backup aumenta il costo sugli annunci in uscita e riduce la preferenza sugli annunci in ingresso
- Quando occorre un fault su un link, l'annuncio del /16 aggregato sull'altro link assicura la connettività

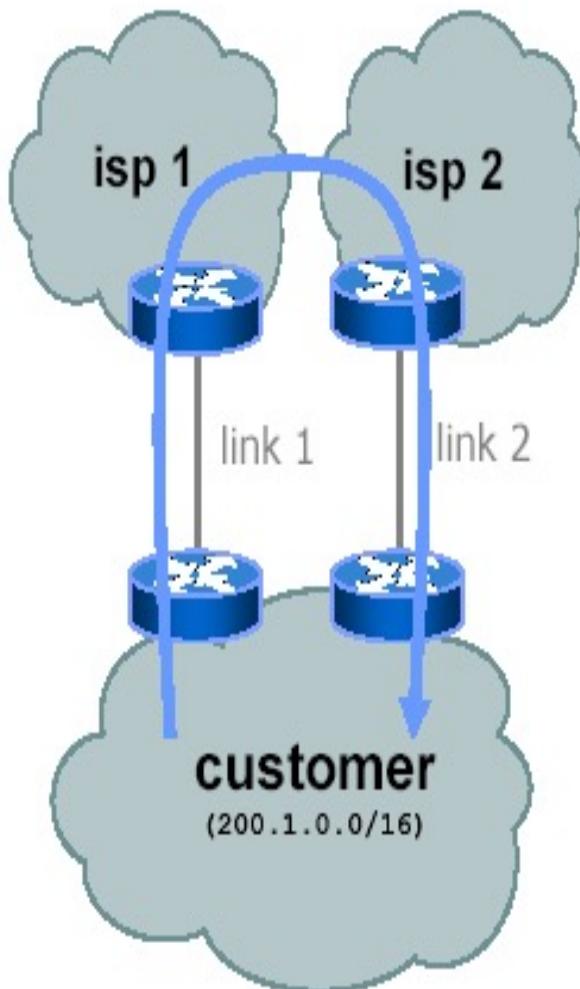
# Multi-homed network



- Due link a due providers differenti
- In genere, due routers sono coinvolti in modo tale da evitare singoli punti di rottura

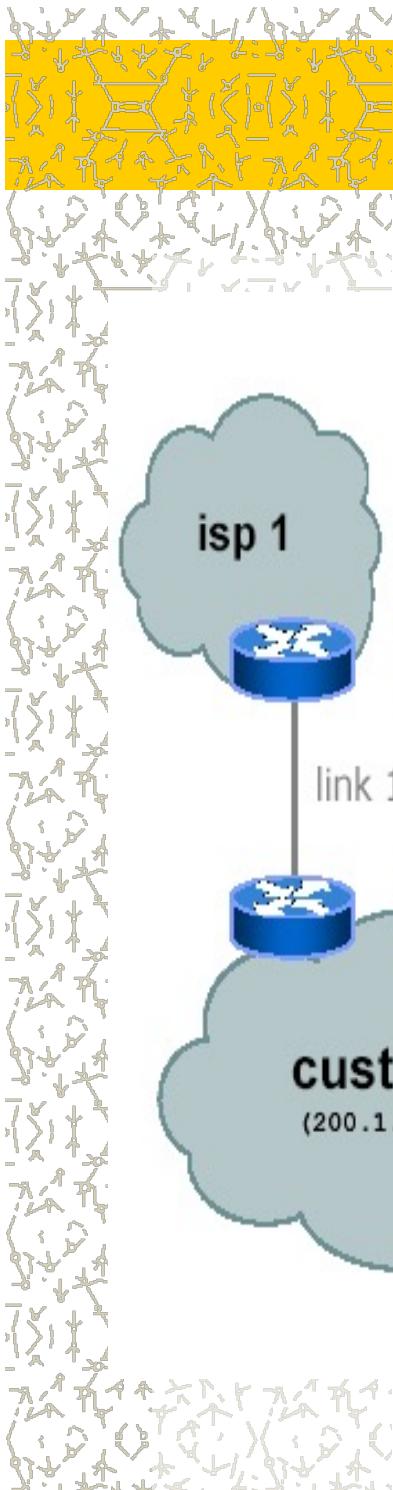


# Instradamento



- Un pacchetto in uscita può essere inviato attraverso uno dei due link per raggiungere Internet
- Un pacchetto in ingresso può usare uno dei due link per raggiungere la rete
- Un pacchetto internet può attraversare il link 1 ed il link 2
- Un pacchetto interno può attraversare entrambi i link

# Partizione del carico

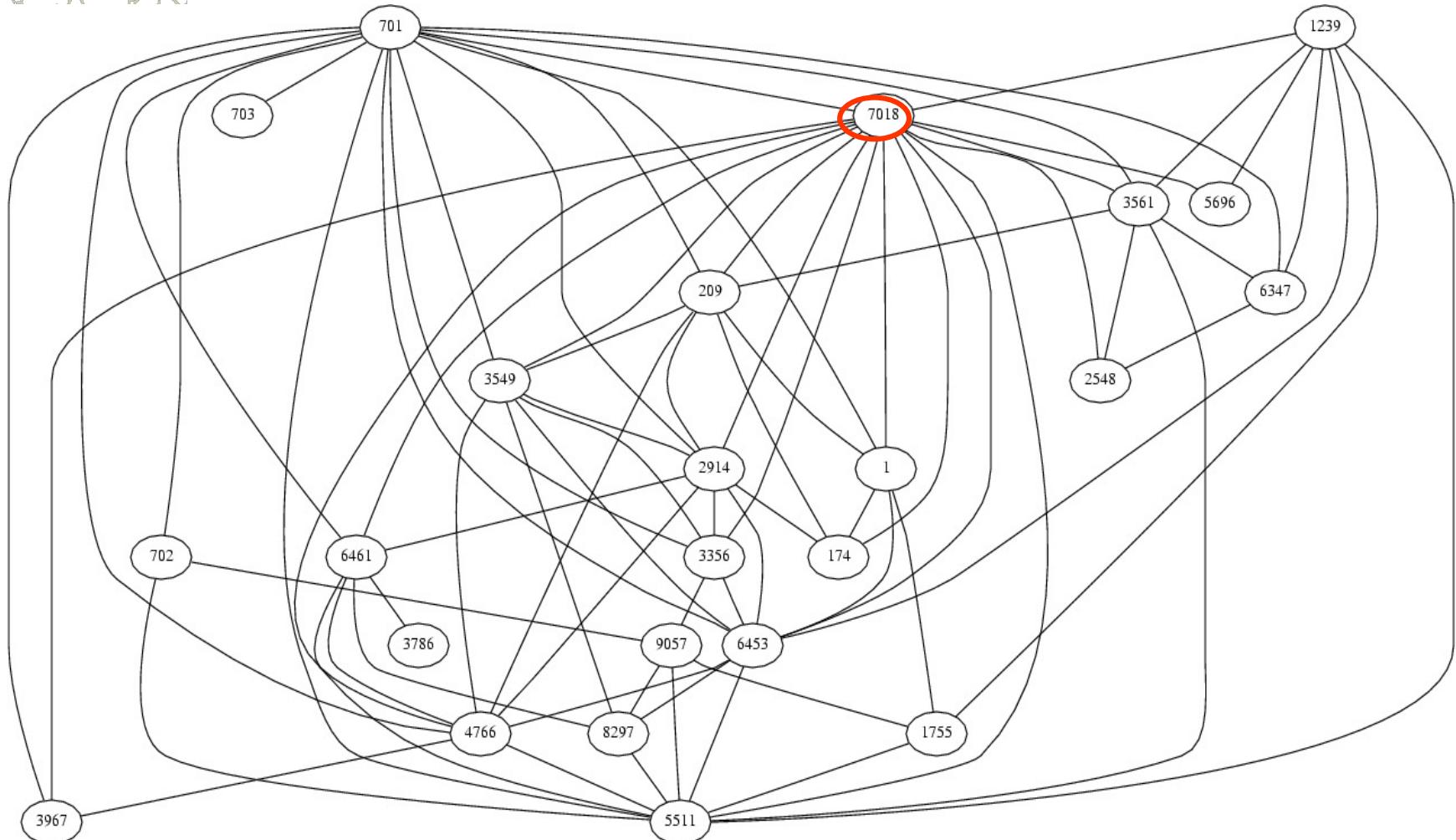


- Elimina il traffico in transito
- Traffico in uscita:
  - Metà degli host interni usano link 1,
  - l'altra metà usa link 2
- Traffico in ingresso:
  - usa link 1 per raggiungere metà degli host interni
  - Usa link 2 per l'altra metà

# Uso di BGP per il partizionamento

- Annuncia /19 aggregato su ogni link
- split /19 e annuncia due /20, uno per ogni link:
  - Partizionamento del traffico approssimato sul traffico in ingresso
  - Assume uguale capacità ed anche distribuzione del traffico sul blocco di indirizzi
- Modifica lo split finchè un partizionamento perfetto è ottenuto
- Accetta l'instradamento di default upstream:
  - Partizionamento del traffico con instradamento verso l'uscita più vicina (igp)
  - Una buona approssimazione poiché molto del traffico è diretto verso la rete

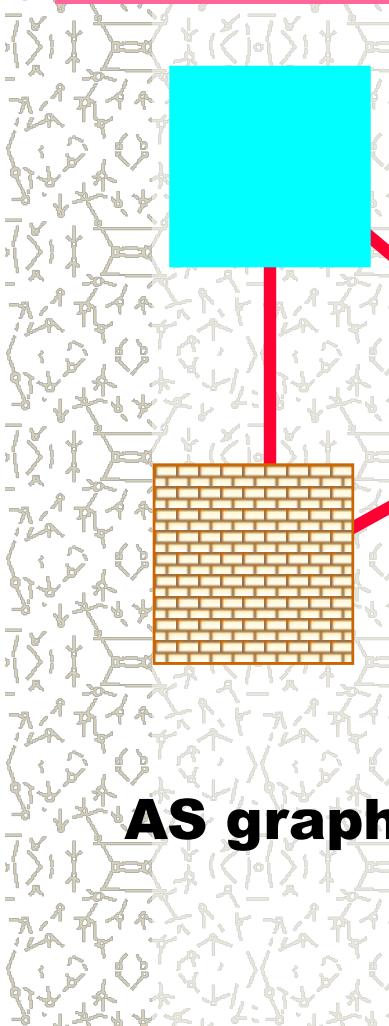
# AS Graph



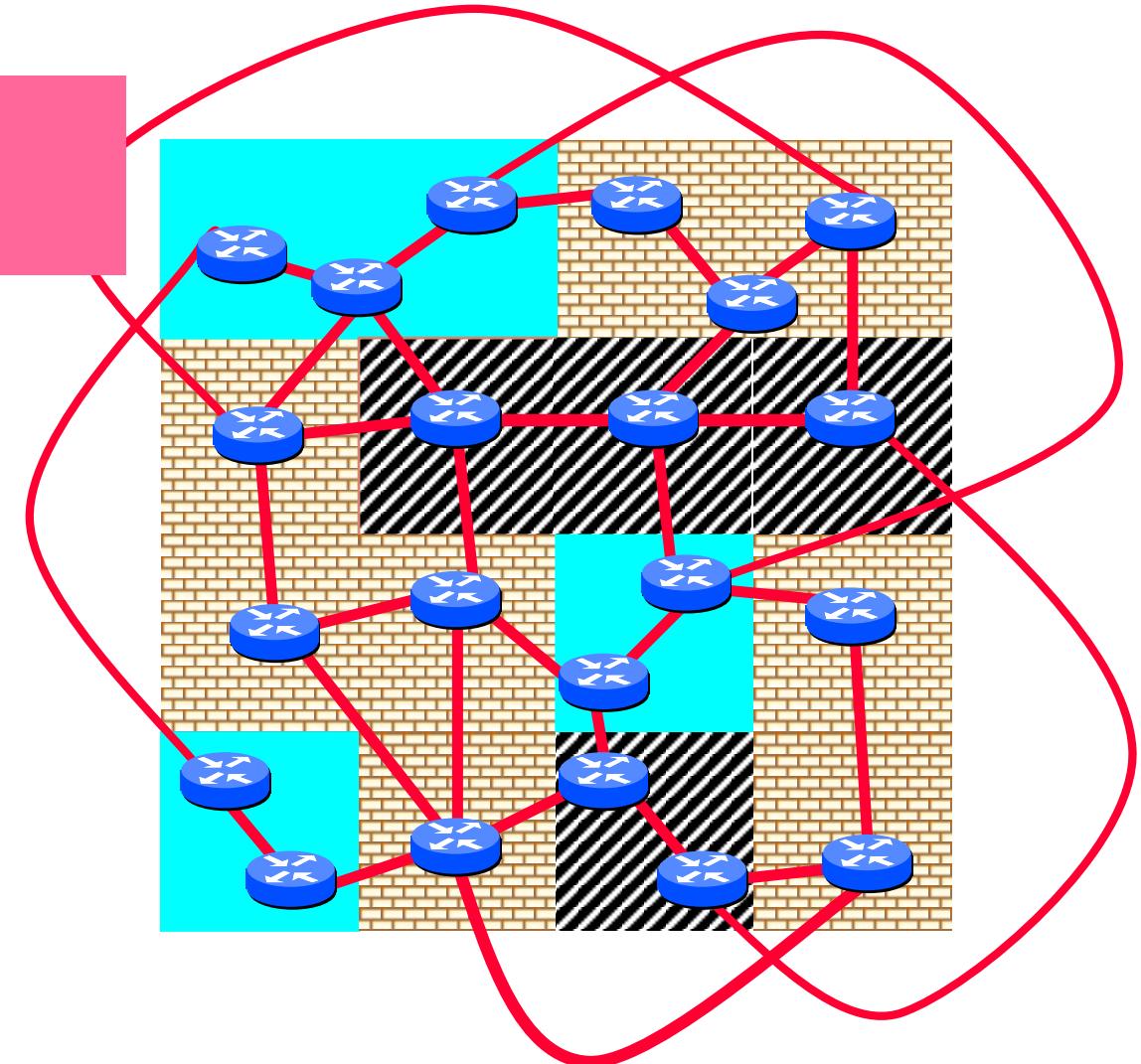
**Questo grafo rappresenta tutti gli Ases che hanno più di 100 peers in un grafo completo di 11,158 nodei: AT&T route-server**

# AS Graph vs Internet Topology

BGP è stato disegnato per ridurre la quantità di informazioni trattate

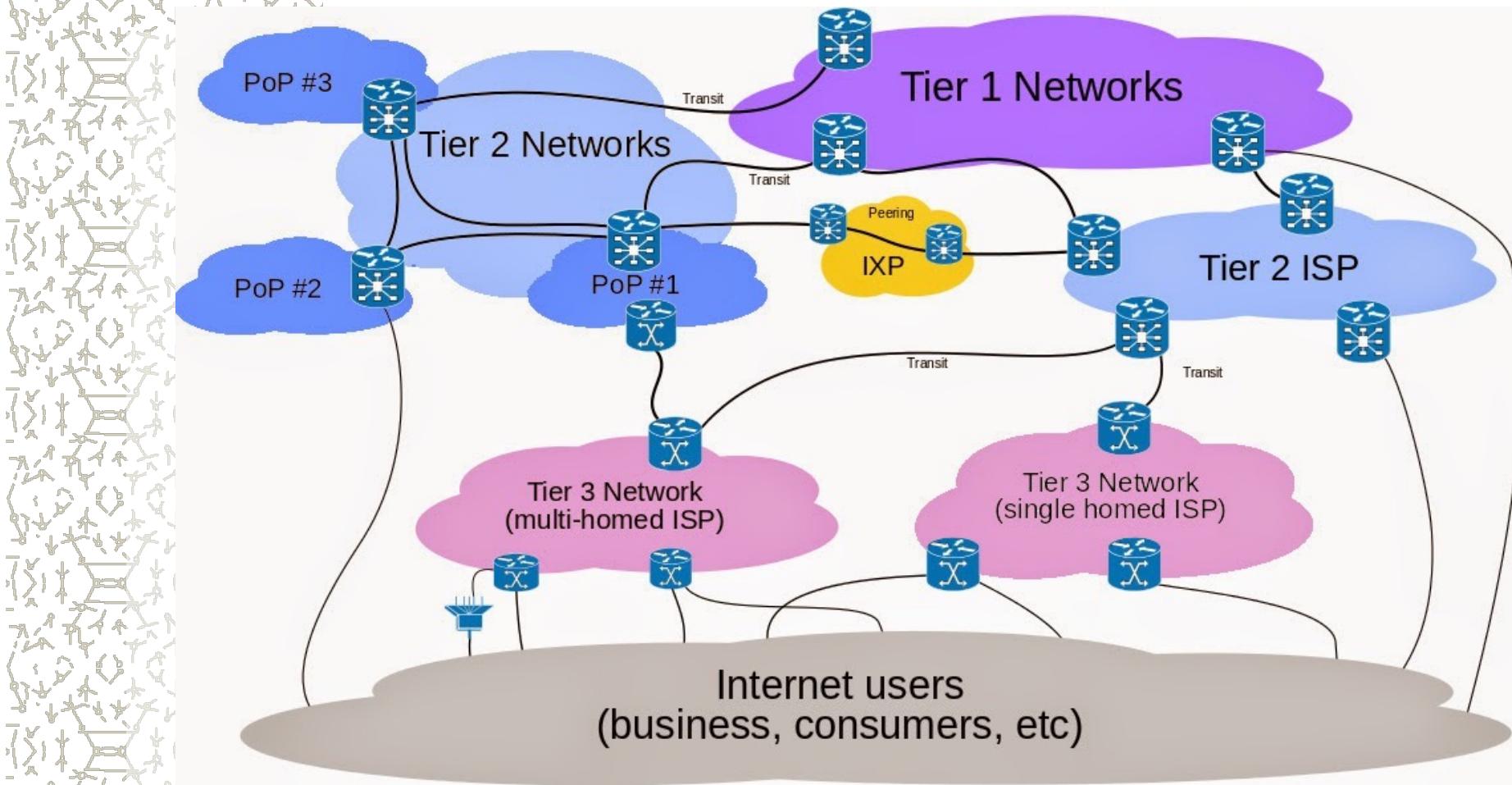


**AS graph**



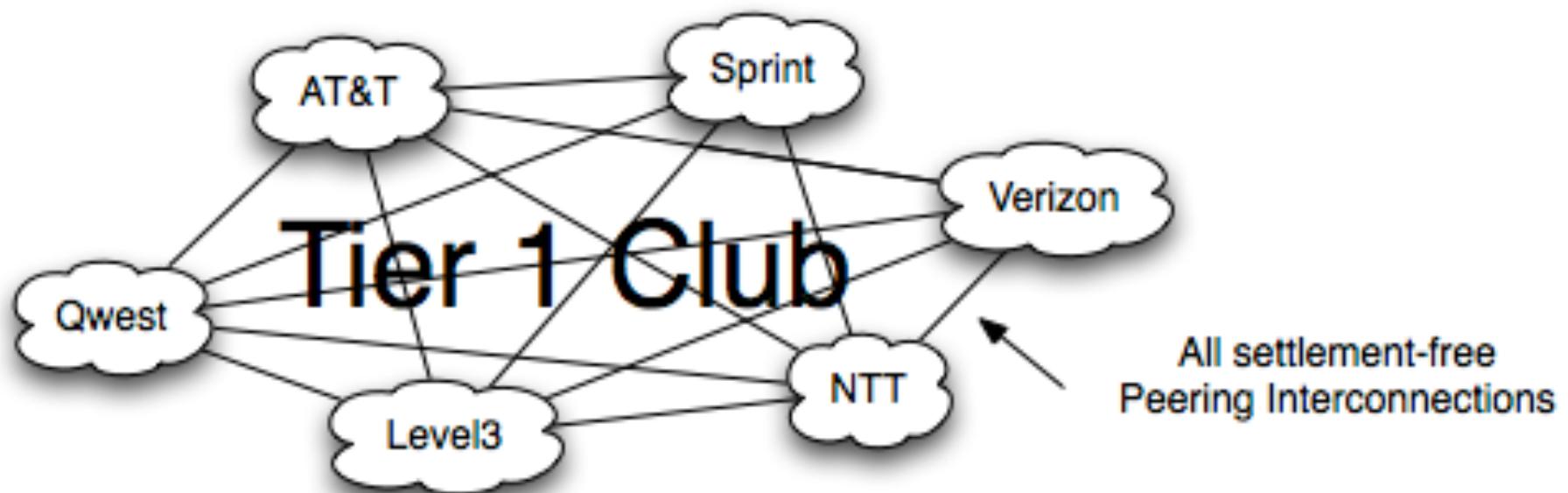
162

# La struttura 3-tiers di internet



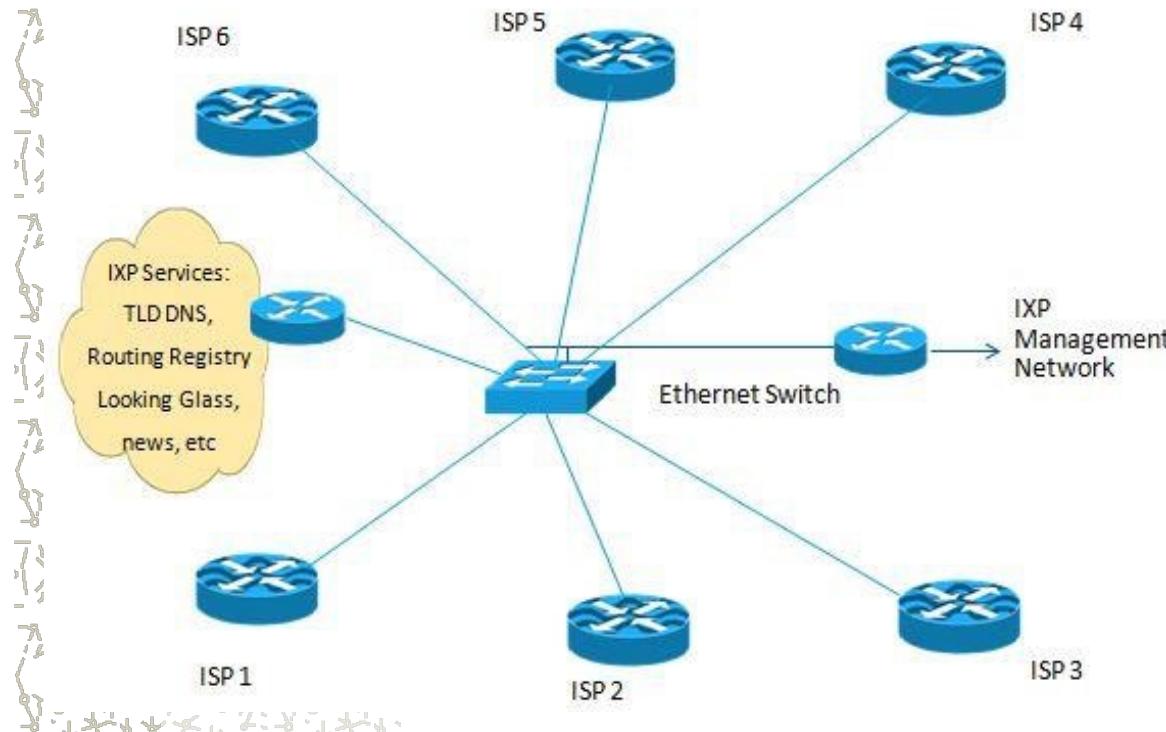
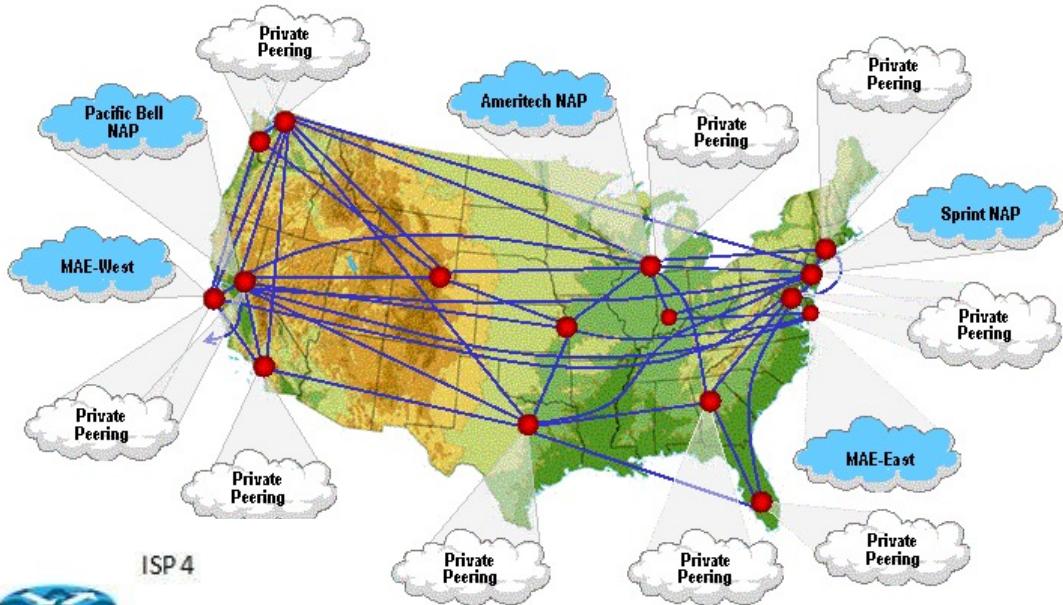
- **Tier3**: providers regionali e reti enterprise
- **Tier2**: providers a grande copertura
- **Tier1**: providers di providers

# Il Club dei Tier 1



# Punti di interscambio: IX e NAP

- Neutralità
- Scambio del transito fra ISP
- Possibilità di stabilire peering

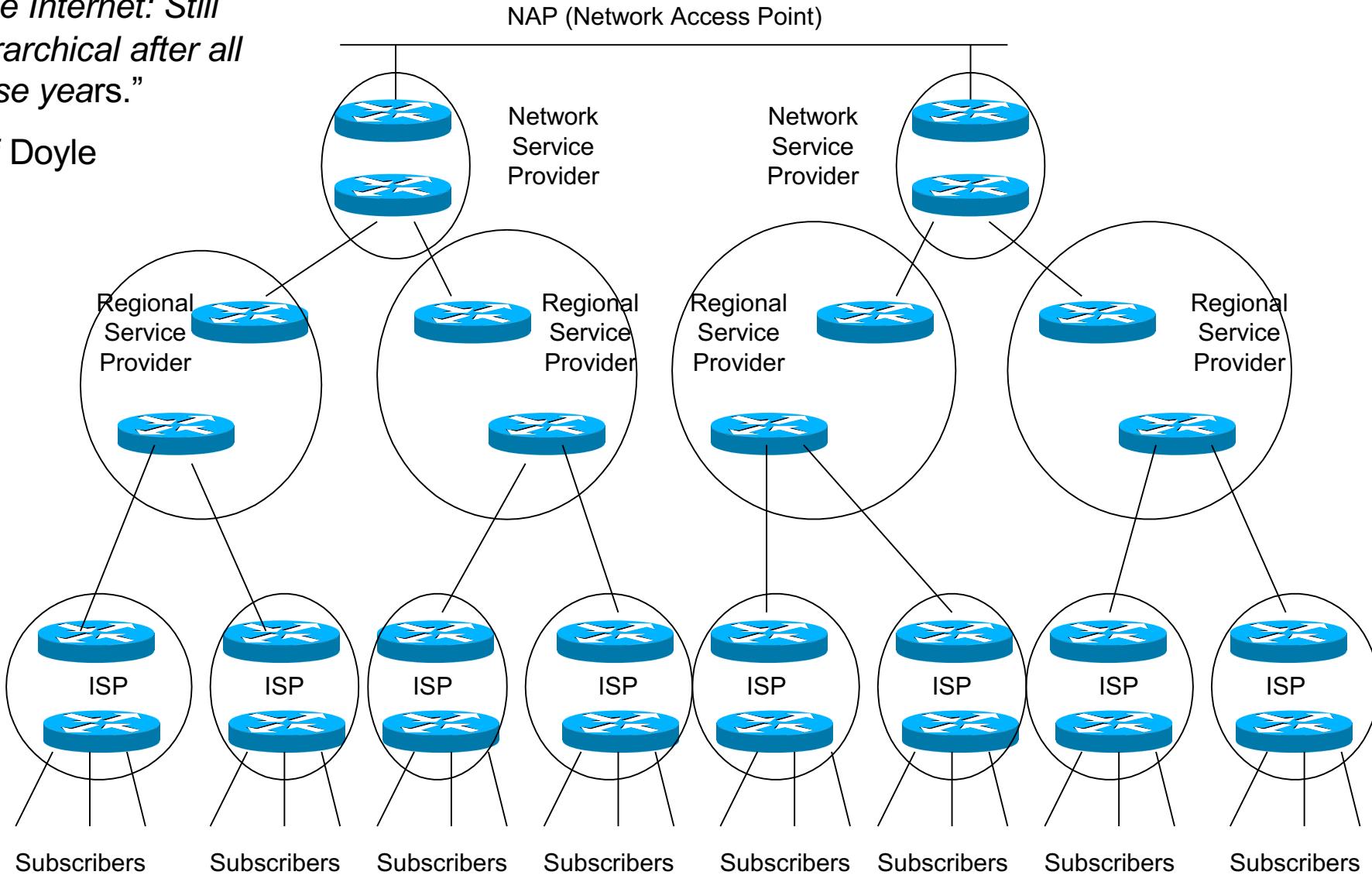


- LAN che interconnette terminazioni di ogni ISP
- Realizzati a livello 2
- Switch ad altissime prestazioni

# Ancora una visione gerarchica

*"The Internet: Still  
hierarchical after all  
these years."*

Jeff Doyle



# Il punto di partenza e quello di arrivo

