



ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

# Instrandamento nelle reti a pacchetto e in Internet

Franco CALLEGATI

Walter CERRONI



# Funzioni di IP

- **Indirizzamento** (Identifica un Calcolatore tramite l'uso di Indirizzi IP)
- **Frammentazione** (Dividere Pacchetti grandi in più piccoli)
- **Instradamento**
  - Decidere che percorso un datagramma deve seguire per raggiungere la destinazione dalla sorgente
  - Utilizza le PCI dei datagrammi, in particolari l'indirizzo destinazione
  - Determina il comportamento della funzione di commutazione nei nodi
- Il problema dell'instradamento è più generale rispetto allo specifico protocollo di livello 3



# Algoritmi e protocolli

- Instradamento = scelta del percorso
- La scelta del percorso spesso significa semplicemente scegliere il prossimo router a cui inviare un pacchetto (scelta del *next hop*)
- Algoritmo di instradamento
  - Metodologia di scelta del next hop
  - Ha obiettivi di ottimalità
    - Semplicità = bassa complessità computazionale
    - Robustezza = capacità di adeguarsi a cambiamenti
    - Stabilità = consistenza di risultato
    - Efficienza = buon uso delle risorse disponibili senza sprechi



# Tabella?

- I nodi di commutazione per applicare l'algoritmo possono utilizzare informazioni predisposte localmente tipicamente sotto forma di tabella
- Algoritmi senza tabella
  - Non fanno uso di tabelle di instradamento
- Algoritmi con tabella
  - Fanno uso di tabelle di instradamento



# Algoritmi di instradamento

- Senza tabella

- Flooding
- Random
- Deflection routing (hot potato)
- Source routing

- Con tabella

- Instradamento fisso e centralizzato (tipicamente utilizzato nelle reti telefoniche)
- Instradamento dinamico a distanza minima (tipicamente utilizzato nei protocolli di rete come l'IP)

- Instradamento fisso e centralizzato:

È una tecnica in cui tutte le decisioni di instradamento sono prese da un'unità centrale. Una volta stabilito il percorso, rimane fisso, indipendentemente dalle condizioni della rete. Questo metodo è efficiente per reti telefoniche tradizionali, dove i percorsi sono statici e prevedibili.

È utilizzato principalmente nelle reti a commutazione di circuito, come le reti telefoniche, poiché garantisce che il percorso scelto sia riservato per l'intera durata della comunicazione.

- Instradamento dinamico a distanza minima:

Nelle reti IP, l'instradamento dinamico si basa su algoritmi che utilizzano la distanza minima per determinare il percorso migliore per inoltrare i pacchetti. Questo metodo è dinamico e può adattarsi ai cambiamenti della rete, come la congestione o la caduta di nodi.

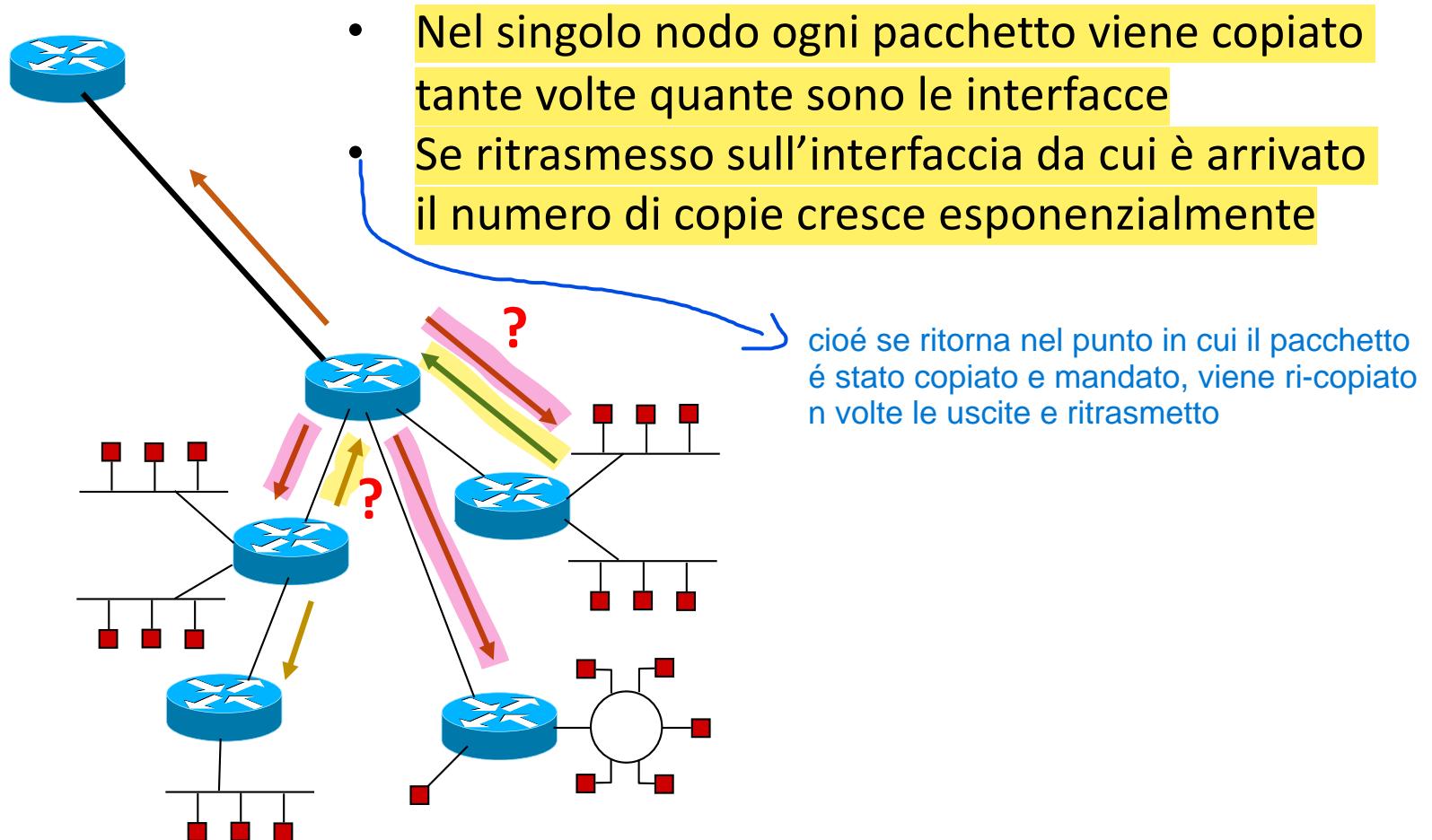


# 1 Flooding

- **Flooding**
  - ogni nodo ritrasmette su tutte le porte di uscita ogni pacchetto ricevuto
  - Prima o poi
    - un pacchetto viene sicuramente ricevuto da tutti i nodi della rete e quindi anche da quello a cui è effettivamente destinato
  - Tutte le strade possibili sono percorse
    - il primo pacchetto che arriva a destinazione ha fatto la strada più breve possibile
  - L'elaborazione associata è pressoché nulla
- Molto adatto quando si desidera inviare una certa informazione a tutti i nodi della rete (*broadcasting*)

# Problema

## Proliferazione dei pacchetti (ritrasmissione di pacchetti già trasmessi)



Ogni pacchetto IP include un campo TTL, che rappresenta il numero massimo di salti (hop) che un pacchetto può effettuare prima di essere scartato. Ogni volta che il pacchetto attraversa un router o nodo intermedio, il valore TTL viene decrementato di 1. Se il valore del TTL scende a 0, il pacchetto viene scartato e non viene più inoltrato.



# Soluzioni

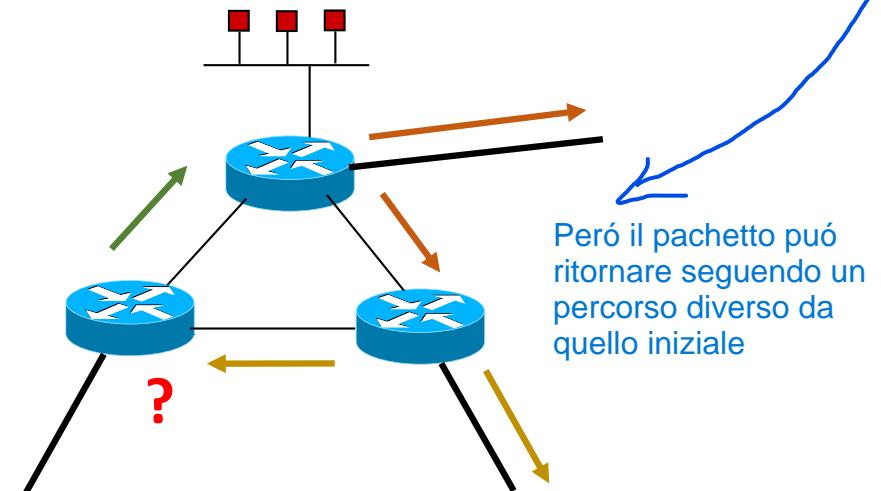
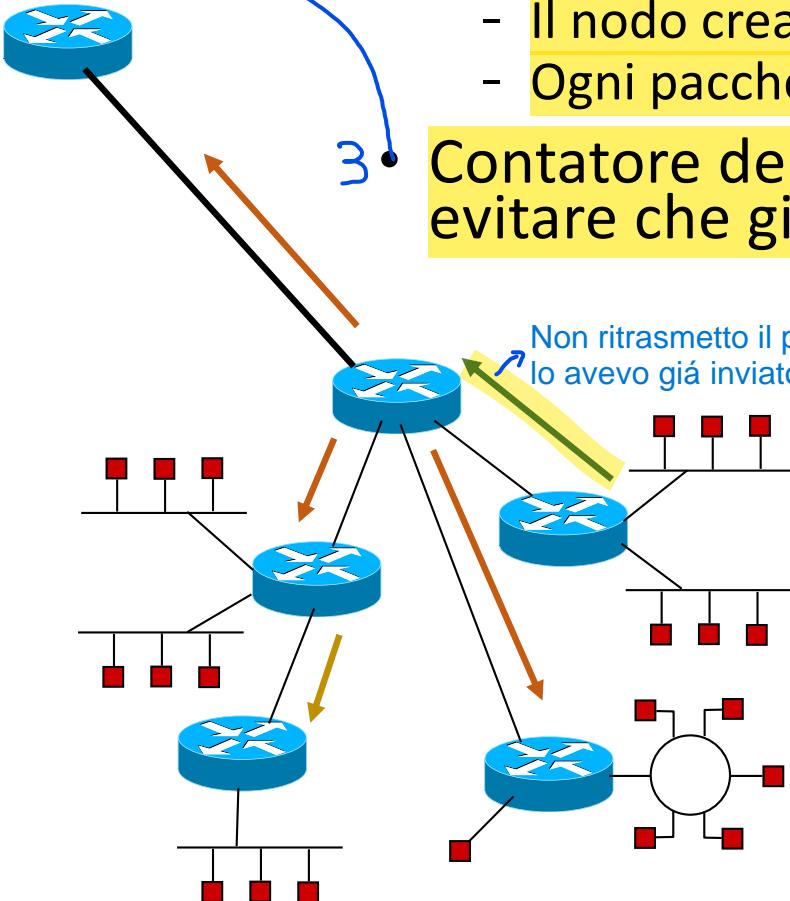
(ho 3 soluzioni diverse)

- 1 • Un nodo non ritrasmette il pacchetto nella direzione dalla quale è giunto

- 2 • Identificazione dei pacchetti

- Ad ogni pacchetto viene associato un identificativo unico (l'indirizzo della sorgente e un numero di sequenza)
- Il nodo crea una lista dei pacchetti ricevuti e ritrasmessi
- Ogni pacchetto già trasmesso, viene ignorato

- 3 • Contatore del tempo di vita (TTL) di un pacchetto per evitare che giri all'infinito





# Dinamicità

Il concetto di dinamicità nell'instradamento si riferisce alla capacità delle reti di adattarsi ai cambiamenti nella loro topologia. Le metodologie di instradamento possono essere classificate in due categorie principali: statico (o fisso) e dinamico.

- Tutte le metodologie di instradamento dovrebbero adattarsi agli eventuali cambiamenti topologici della rete
- Lo possono fare più o meno velocemente per cui si parla di instradamento
- **Statico (o fisso)** In un instradamento statico, i percorsi sono decisi al momento dell'inizializzazione della rete e rimangono invariati per tutto il tempo, a meno che non si decida di aggiornare manualmente la configurazione.
  - I percorsi sono decisi in momenti specifici (inizializzazione della rete) e non cambiano sul breve periodo
  - Se c'è un cambiamento repentino della topologia questo viene recepito solamente alla prossima inizializzazione
- **Dinamico**
  - I percorsi vengono modificati periodicamente per adattarsi velocemente ad eventuali cambiamenti della rete

Nell'instradamento dinamico, le tabelle di instradamento sono aggiornate periodicamente in base ai cambiamenti della rete. Questo consente di adattarsi rapidamente a modifiche topologiche come l'aggiunta o la rimozione di nodi, variazioni di carico di traffico o guasti dei collegamenti.



## 2 Random

- Il next hop viene scelto a caso fra quelli possibili
- Le probabilità possono essere diverse e modificabili nel tempo
- Problemi
  - Non garantisce la consegna in tempi certi
  - Potrebbe dar luogo a comportamenti instabili (loop)

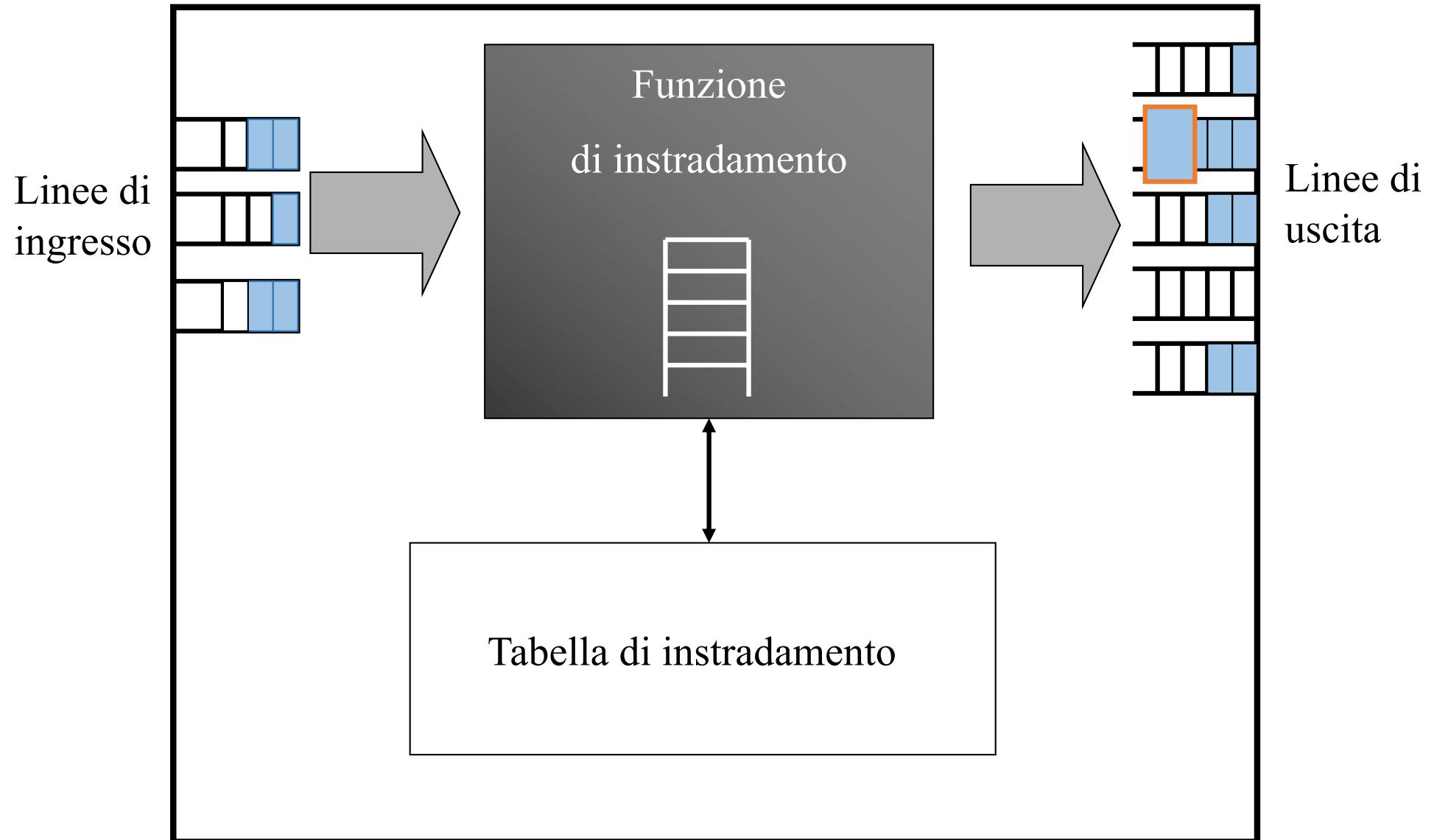


# Deflection routing (hot potato)

- Quando un nodo riceve un pacchetto lo ritrasmette sulla linea d'uscita avente il minor numero di pacchetti in attesa di essere trasmessi
- E' adatto a reti in cui
  - i nodi di commutazione dispongono di spazio di memorizzazione molto limitato
  - si desidera minimizzare il tempo di permanenza dei pacchetti nei nodi
- Problemi
  - I pacchetti possono essere ricevuti fuori sequenza
  - Alcuni pacchetti potrebbero percorrere all'infinito un certo ciclo in seno alla rete, semplicemente perché le sue linee sono poco utilizzate In alcuni casi, i pacchetti potrebbero circolare all'infinito in determinati cicli all'interno della rete, poiché vengono continuamente inoltrati su linee poco utilizzate.
- Si deve prevedere un meccanismo per limitare il tempo di vita dei pacchetti
- Non tiene conto della destinazione finale del pacchetto



# Instradamento con tabella



Lo Store-and-Forward viene comunemente utilizzato in switch di rete, router

# Store-and-Forward



Lo Store-and-Forward è un meccanismo di instradamento dei pacchetti in cui ciascun pacchetto viene prima completamente ricevuto, memorizzato e verificato da un nodo prima di essere ritrasmesso verso il prossimo nodo. Questo processo garantisce che ogni pacchetto sia corretto e completo prima di proseguire, minimizzando la possibilità di trasmissione di dati corrotti.

- Il pacchetto entrante è verificato e memorizzato
- Si estraggono le informazioni di instradamento dall'intestazione (indirizzo, priorità, classe di servizio)
- Si confrontano queste informazioni con la tabella di instradamento
  - Identificando una o più uscite su cui inviare il pacchetto
- Il pacchetto è inserito nella coda relativa all'uscita prescelta, in attesa della effettiva trasmissione

Il pacchetto viene prima memorizzato interamente nel nodo e quindi ritrasmesso nella direzione opportuna

In generale dovrebbe esistere una base dati per il confronto che è la tabella di instradamento



# Shortest path routing

- Si assume che ad ogni collegamento della rete possa essere attribuita una lunghezza
  - Lunghezza
    - è un numero che serve a caratterizzare il peso di quel collegamento nel determinare la funzione di costo del percorso totale di trasmissione
  - L'algoritmo cerca la strada di lunghezza minima fra ogni mittente e ogni destinatario
  - Si applicano algoritmi di calcolo dello shortest path (Bellman-Ford e Dijkstra)
  - L'implementazione può avvenire in modalità
    - Centralizzata
      - Un solo nodo esegue i calcoli per tutti
    - Distribuita
      - Ogni nodo esegue i calcoli per sé stesso
      - Sincrona
        - Tutti i nodi eseguono gli stessi passi dell'algoritmo nello stesso istante
      - Asincrona
        - I nodi eseguono lo stesso passo dell'algoritmo in momenti diversi
- Ogni nodo esegue i calcoli di instradamento per sé stesso. In questa modalità, i nodi cooperano tra loro scambiando informazioni sulla rete e aggiornando periodicamente le proprie tabelle di instradamento.
- Un singolo nodo della rete esegue tutti i calcoli necessari per determinare i percorsi ottimali tra tutti i mittenti e i destinatari. Questo nodo ha la responsabilità di aggiornare e distribuire le informazioni di instradamento agli altri nodi.



# Rappresentazione della rete

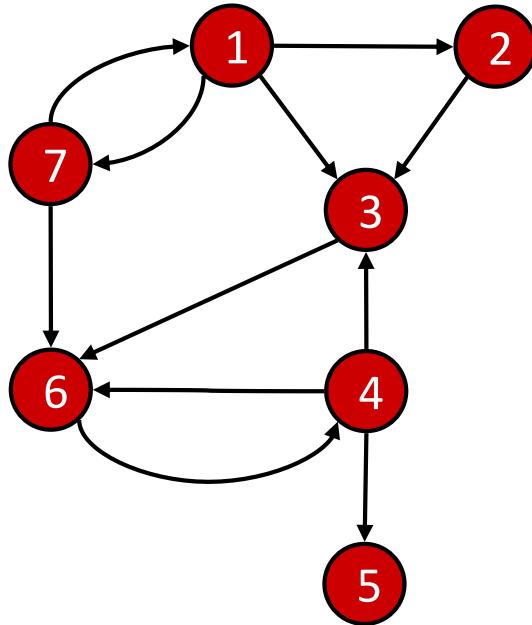
- Ad una generica rete si può facilmente associare un grafo orientato:
    - i nodi rappresentano i terminali ed i commutatori
    - gli archi rappresentano i collegamenti
    - L'orientazione degli archi rappresenta la direzione di trasmissione
    - il peso degli archi rappresenta il costo dei collegamenti, che può essere espresso in termini di
      - numero di nodi attraversati (ogni arco ha peso unitario)
      - distanza geografica
      - ritardo introdotto dal collegamento
      - inverso della capacità del collegamento
      - costo di un certo instradamento
      - una combinazione dei precedenti
- Quando si usa l'inverso della capacità, l'idea è che un collegamento con una maggiore capacità abbia un costo inferiore, perché è più efficiente e può trasportare più dati con meno congestione.



# Il grafo della rete

- Una rete è un insieme di nodi di commutazione interconnessi da collegamenti
- Per rappresentarla si possono usare i modelli matematici della teoria dei grafi
  - Sia  $V$  un insieme finito di **nodi**
  - Un **arco** è definito come una coppia di nodi  $(i,j)$ ,  $i,j \in V$
  - Sia  $E$  un insieme di archi
  - Un **grafo**  $G$  è definito come la coppia  $(V,E)$  e può essere
    - **orientato** se  $E$  consiste di coppie ordinate, cioè se  $(i,j) \neq (j,i)$
    - **non orientato** se  $E$  consiste di coppie non ordinate, cioè se  $(i,j) = (j,i)$
  - Se  $(i,j) \in E$ , il nodo  $j$  è **vicino** del nodo  $i$

# Rappresentazione di grafi



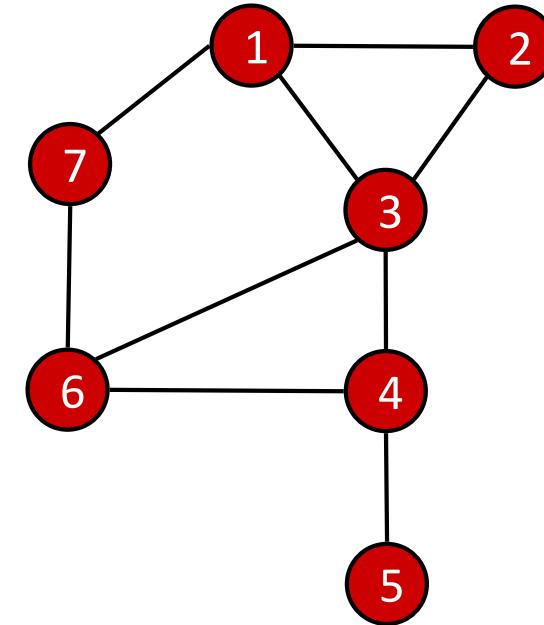
Coppie Ordinate

## Grafo orientato

$$V = \{1, 2, 3, 4, 5, 6, 7\}$$

$$E = \{(1,2), (1,3), (1,7), (2,3), (3,6), (4,3), (4,5), (4,6), (6,4), (7,1), (7,6)\}$$

Dimensioni:  $|V|=7$ ,  $|E|=11$



Coppie non ordinate

## Grafo non orientato

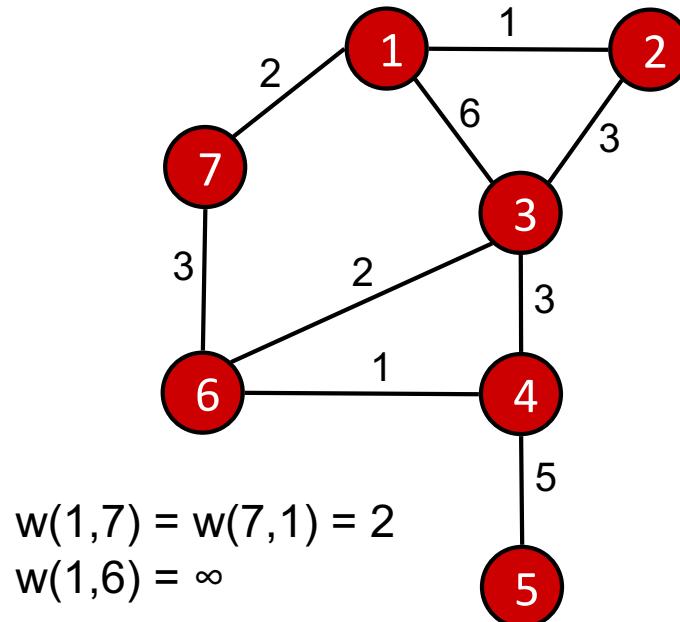
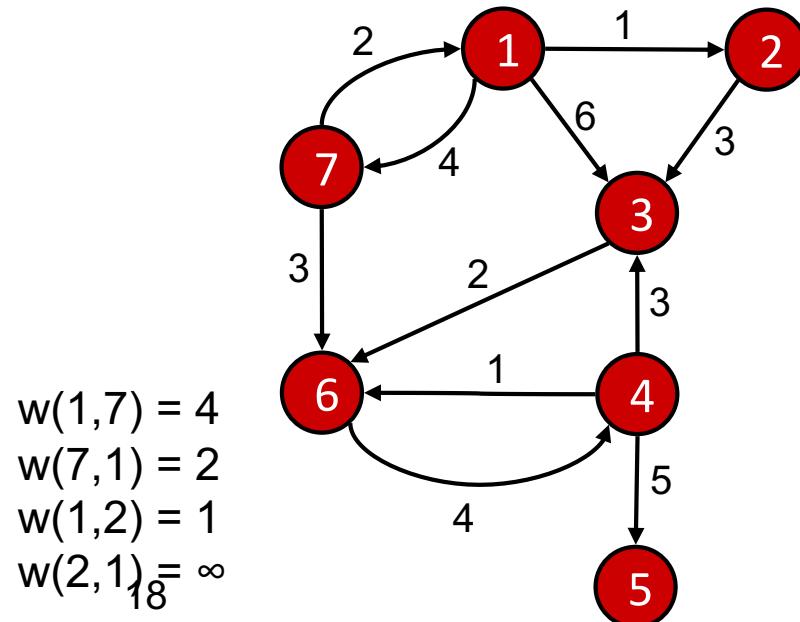
$$V = \{1, 2, 3, 4, 5, 6, 7\}$$

$$E = \{(1,2), (1,3), (1,7), (2,3), (3,4), (3,6), (4,5), (4,6), (6,7)\}$$

Dimensioni:  $|V|=7$ ,  $|E|=9$

# Grafo pesato

- Un **grafo pesato** è un grafo  $G=(V,E)$  tale che ad ogni arco  $(i,j) \in E$  è associato un numero reale  $w(i,j)$  chiamato **peso** (o costo, o distanza)
  - In un grafo non orientato vale sempre  $w(i,j) = w(j,i)$
  - In un grafo orientato vale in generale  $w(i,j) \neq w(j,i)$
  - Se  $(i,j) \notin E$ , allora  $w(i,j) = \infty$
  - Per semplicità si assume  $w(i,j) > 0$  per ogni arco  $(i,j) \in E$





# Routing shortest path nel mondo IP

- Quando i nodi di rete vengono accesi conoscono solamente la configurazione delle loro interfacce
  - Statica Configurazione statica: L'utente o l'amministratore configura manualmente l'interfaccia di rete.
  - Dinamica con DHCP Configurazione dinamica tramite DHCP: Il dispositivo può ottenere automaticamente l'indirizzo IP e le altre informazioni di rete tramite un server DHCP
- Con queste informazioni popolano la tabella di instradamento iniziale
- Per implementare il routing shortest path verso una qualunque destinazione devono utilizzare
  - Uno o più protocolli di routing per scambiarsi informazioni ed apprendere la topologia della rete
  - Uno o più algoritmi per il calcolo degli SP sulla base delle informazioni ottenute

Il Distance Vector non è un protocollo specifico, ma un metodo di instradamento utilizzato da alcuni protocolli di routing per determinare il percorso migliore per trasmettere i dati in una rete.



# 1 Routing Distance Vector

(Utile se si ha una rete con poche network)

- Basato su Bellman-Ford, in versione dinamica e distribuita proposta da Ford-Fulkerson
- Implementa meccanismi di dialogo per fare sì che
  - Ogni nodo scopre i suoi vicini e ne calcola la distanza da se stesso
  - Ad ogni passo, ogni nodo invia ai propri vicini un vettore contenente la stima della sua distanza da tutti gli altri nodi della rete (quelli di cui è a conoscenza)
- È un protocollo semplice e richiede poche risorse
- Problemi:
  - convergenza lenta, partenza lenta (cold start)
  - problemi di stabilità: conteggio all'infinito

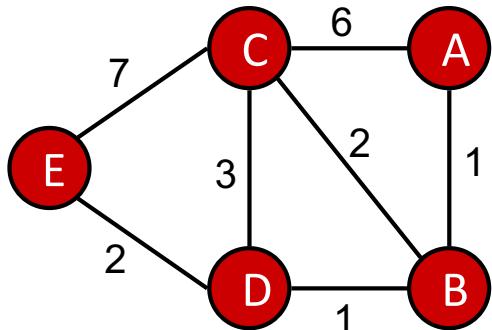
Meccanismo di Funzionamento:

Scoperta dei vicini: Ogni nodo inizia calcolando la distanza dai propri vicini, ossia i nodi direttamente connessi. Questa distanza è generalmente misurata in termini di numero di hop o in base a metriche come la larghezza di banda, il ritardo o altri fattori (quindi in base al peso degli archi).

Invio del vettore di distanza: Ad ogni iterazione, ogni nodo invia ai suoi vicini un vettore che contiene la sua stima della distanza da ogni altro nodo della rete di cui è a conoscenza. Questo vettore viene utilizzato dai nodi vicini per aggiornare le loro stime e trovare percorsi più brevi.

Iterazione continua: Il processo viene ripetuto ad intervalli regolari. Ogni nodo riceve i vettori dai propri vicini e aggiorna la sua tabella di instradamento se trova un percorso più corto attraverso uno dei suoi vicini.

# Esempio



Distance Vector iniziali:  $DV(i) = \{(i, 0)\}$ , per  $i = A, B, C, D, E$

Distance Vector dopo la scoperta dei vicini:

$$DV(A) = \{(A, 0), (B, 1), (C, 6)\}$$

$$DV(B) = \{(A, 1), (B, 0), (C, 2), (D, 1)\}$$

$$DV(C) = \{(A, 6), (B, 2), (C, 0), (D, 3), (E, 7)\}$$

$$DV(D) = \{(B, 1), (C, 3), (D, 0), (E, 2)\}$$

$$DV(E) = \{(C, 7), (D, 2), (E, 0)\}$$

## Evoluzione delle tabelle di routing

1. A riceve DV(B)

dest	Costo, next hop
A	0
B	1, B
C	3, B
D	2, B

2. A riceve DV(C)

dest	Costo, next hop
A	0
B	1, B
C	3, B
D	2, B
E	10, B

3. B riceve DV(D)

dest	Costo, next hop
A	1, A
B	0
C	2, C
D	1, D
E	3, D

4. A riceve DV(B)

dest	Costo, next hop
A	0
B	1, B
C	3, B
D	2, B
E	4, B

Tabella di A

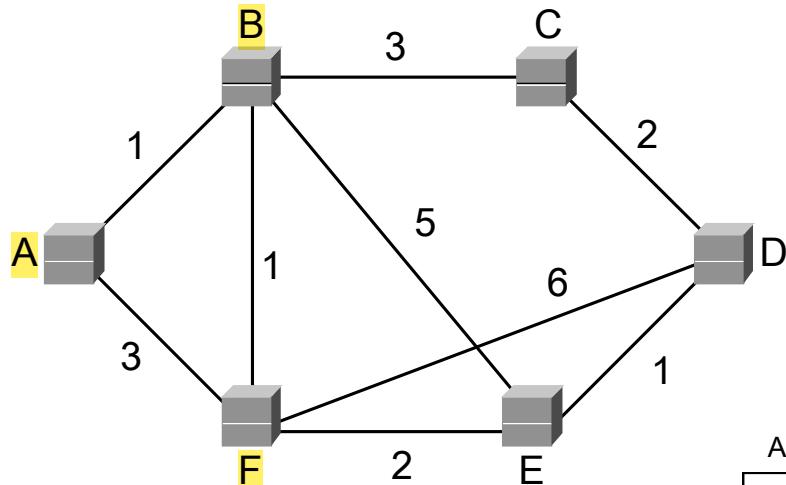
Tabella di A

Tabella di B

Tabella di A  
21



# Esempio 3.18 dal libro



## Evoluzione del distance vector di A

A riceve i DV sempre solo da B e F  
(suo diretti vicini)

I DV di B ed F sono progressivamente più completi e permettono ad A di venire a conoscenza dell'intera rete

**B - Time  $T$**

D	C
A	1
B	-
C	3
E	5
F	1

**B - Time  $2T$**

D	C
A	1
B	-
C	3
D	5
E	3
F	1

**B - Time  $3T$**

D	C
A	1
B	-
C	3
D	4
E	3
F	1

**B - Time  $4T$**

D	C
A	1
B	-
C	3
D	5
E	4
F	1

**A - Time  $T$**

D	C	NH
A	-	
B	1	B
F	3	F

**A - Time  $2T$**

D	C	NH
A	-	
B	1	B
C	4	B
D	9	F
E	5	F
F	2	B

**A - Time  $3T$**

D	C	NH
A	-	
B	1	B
C	4	B
D	6	F
E	4	B
F	2	B

**A - Time  $4T$**

D	C	NH
A	-	
B	1	B
C	4	B
D	5	B
E	4	B
F	2	B

**A - Time  $5T$**

D	C	NH
A	-	
B	1	B
C	4	B
D	6	B
E	5	B
F	2	B

**F - Time  $T$**

D	C
A	3
B	1
D	6
E	2
F	-

**F - Time  $2T$**

D	C
A	2
B	1
C	4
D	3
E	2
F	-

**F - Time  $3T$**

D	C
A	2
B	1
C	4
D	6
E	3
F	-

**F - Time  $4T$**

D	C
A	2
B	1
C	4
D	6
E	3
F	-



# Algoritmo

Con la ricezione di DV, i nodi conoscono altri nodi che non sono connessi al nodo sorgente

- Nodo sorgente del traffico denominato  $s$
- $D_i^h$  costo del percorso di lunghezza minima da  $s$  a  $j$  in al più  $h$  salti ( $h$  indica il numero di salti dal nodo sorgente al nodo  $j$ )
- $d_{ij}$  costo del collegamento diretto fra  $i$  e  $j$  (si scambiano informazioni di routing)
- $d_{ij} = \infty$  se  $i$  e  $j$  non sono connessi direttamente (non si scambiano informazioni di routing)
- Per  $h=1$  
$$D_j^h = d_{sj} \quad \forall j \neq s$$
 (Se sono vicini/adiacenti,  $h = 1$ , quindi collegamento diretto)
- Per  $h = h+1$  
$$D_j^h = \min_i \{ D_i^{h-1} + d_{ij}, D_j^{h-1} \}$$

(Se non sono vicini/adiacenti,  $h \neq 1$ , quindi devo valutare se è minima la distanza che ho già nella tabella oppure quella che mi è stata inviata dal nodo vicino/adiacente)



# Cold start e tempo di convergenza

- Allo start-up le tabelle dei singoli nodi contengono solo l'indicazione delle distanze dagli immediati vicini
- Da qui in poi lo scambio dei distance vector permette la creazione di tabelle sempre più complete (conosco sempre di più la rete)
- L'algoritmo converge al più dopo un numero di passi pari al numero di nodi della rete
- Se la rete è molto grande il tempo di convergenza può essere lungo.
- Cosa succede se lo stato della rete cambia in un tempo inferiore a quello di convergenza dell'algoritmo?
  - Risultato imprevedibile → si ritarda la convergenza

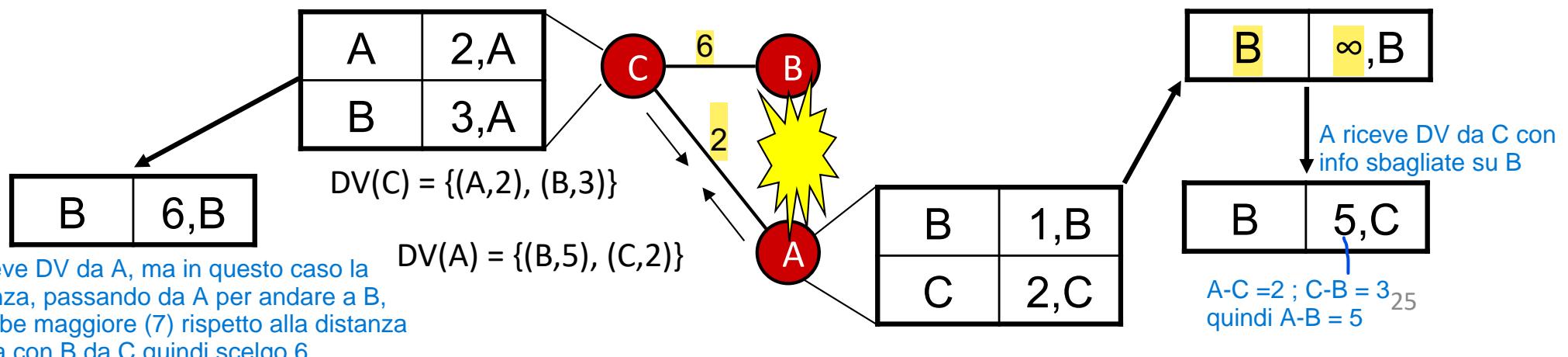
La convergenza nell'algoritmo di instradamento Distance Vector si verifica quando ogni nodo della rete ha una tabella di instradamento aggiornata e completa. Questo significa che ciascun nodo ha calcolato il percorso minimo per raggiungere ogni altro nodo nella rete, sulla base delle informazioni fornite dai suoi vicini.

# Bouncing effect

Il bouncing effect rallenta la convergenza complessiva della rete e spreca risorse, specialmente in topologie più grandi e complesse dove la propagazione degli aggiornamenti richiede più tempo. Abbiamo, in questo caso, pacchetti che continuano a circolare tra i nodi, fino all'esaurimento del TTL (Time-To-Live) o finché i nodi non convergono nuovamente su informazioni aggiornate.

- Il link fra due nodi A e B cade
    - A e B si accorgono che il collegamento non funziona e immediatamente pongono ad infinito la sua lunghezza
- (Modifica Tabella Distanze dei Nodi A e B)
- - Se altri nodi hanno nel frattempo inviato anche i loro vettori delle distanze, <sup>con informazioni obsolete</sup> si possono creare delle incongruenze temporanee, di durata dipendente dalla complessità della rete
  - ad esempio A crede di poter raggiungere B tramite un altro nodo C che a sua volta passa attraverso A (causato dal fatto che si mandano DV con info sbagliate)
- Queste incongruenze possono dare luogo a cicli, per cui due o più nodi si scambiano datagrammi fino a che non si esaurisce il TTL o finché non si converge nuovamente

(perché la convergenza dipende dalla dimensione della rete)

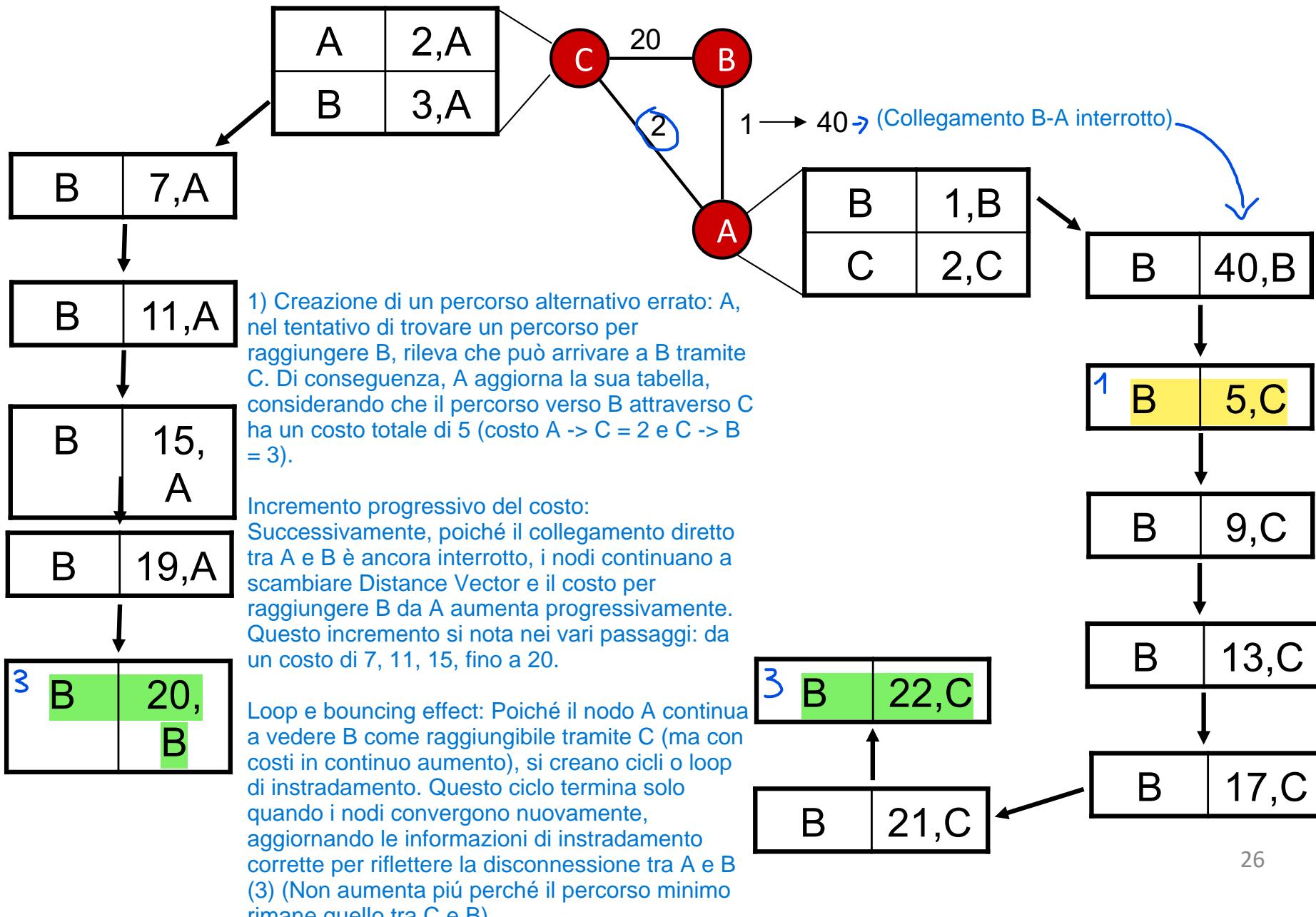


In questo caso, si verifica "Bouncing Effect", perché C manda il proprio DV con informazioni obsolete prima che A o B comunichino le informazioni aggiornate (essendo che i DV vengono mandati periodicamente)



# Convergenza lenta

(stessa cosa della slide precedente ma più tempo di convergenza per completare le tabelle)

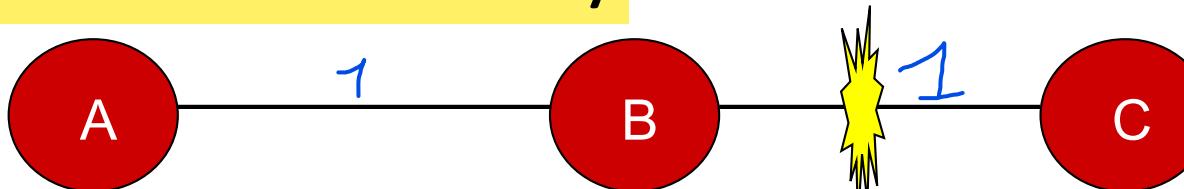


Pacchetti mandati da A e B verso C quando collegamento B-C fuori servizio, rimangono in vita e rimangono nel canale A-B occupando la capacità del canale, fino al termine del TTL.



# Count to infinity

(non si converge proprio)



- Situazione iniziale:  $D_{AB} = 1$ ,  $D_{BC} = 1$  e  $D_{AC} = 2$

- Link BC va fuori servizio (B mette nella Tabella il collegamento con C ad infinito)
- B riceve il DV di A che contiene l' informazione  $D_{AC} = 2$ , per cui esso computa una nuova  $D'_{BC} = D_{BA} + D_{AC} = 3$  (Il problema si crea perché dopo che il canale B-C scompare, A manda DV a B)
- B comunica ad A la sua nuova distanza da C
- A calcola la nuova distanza  $D_{AC} = D_{AB} + D'_{BC} = 4$  (A convinto che la distanza da C fosse 2, quindi aggiorna a 3)
- ... così all'infinito

- La cosa può andare avanti all' infinito

- Si può interrompere imponendo che quando una distanza assume un valore  $D_{IJ} > D_{\max}$  allora si suppone che il nodo destinazione J non sia più raggiungibile

- Inoltre si possono introdurre meccanismi migliorativi

- **Split horizon**
- **Triggered update**



# Split horizon

A manda DV a B omettendo le distanze che hanno come next hop B (forma semplice) oppure mettendole ad infinito (poisonous reverse)

- Split horizon è una tecnica molto semplice per risolvere in parte i problemi suddetti
  - se A instrada i pacchetti verso una destinazione X tramite B, non ha senso per B cercare di raggiungere X tramite A
  - di conseguenza non ha senso che A renda nota a B la sua distanza da X
- Un algoritmo modificato di questo tipo richiede che un router invii informazioni diverse ai diversi vicini
- Split horizon in forma semplice:
  - A omette la sua distanza da X nel DV che invia a B
- Split horizon with poisonous reverse:
  - A inserisce tutte le destinazioni nel DV diretto a B, ma pone la distanza da X uguale ad infinito



# Triggered update

- Una ulteriore modifica per migliorare i tempi di convergenza è relativa alla tempistica con cui inviare i DV ai vicini
  - i protocolli basati su questi algoritmi richiedono di inviare periodicamente le informazioni delle distanze ai vicini
  - è possibile che un DV legato ad un cambiamento della topologia parta in ritardo e venga sopravanzato da informazioni vecchie inviate da altri nodi Per evitare ciò: ↗
- Triggered update: un nodo deve inviare immediatamente le informazioni a tutti i vicini qualora si verifichi una modifica della propria tabella di instradamento

(così si evita "Count to Infinity" perché alla modifica del collegamento B-C, il nodo B manda subito DV evitando che arrivi prima quello di A)

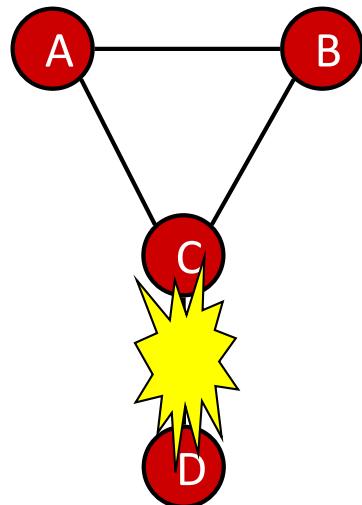


# Ma non basta....

riducono la possibilità di errori,  
ma non li risolvono al 100%

- I diversi rimedi proposti in realtà non sono davvero risolutivi

- sono ancora presenti situazioni patologiche in cui i protocolli Distance Vector convergono troppo lentamente o non convergono affatto



- Inizialmente, A e B raggiungono D tramite C
- Dopo il guasto, C mette a  $\infty$  la sua dist. da D
- Dopo aver ricevuto il DV da C, A crede di poter raggiungere comunque D tramite B
- Idem per B che crede di poter usare A
- Stavolta A e B trasmettono i propri DV a C
- Si crea di nuovo un loop e un problema di convergenza



# Il routing link state

Separa radicalmente algoritmo da protocollo di invio di informazioni di routing.

- Utilizzando il protocollo di routing ogni nodo si costruisce un'immagine del grafo della rete
- Il protocollo di routing ha come scopo fondamentale quello di permettere ad ogni nodo di crearsi l'immagine della rete
  - scoperta dei nodi vicini
  - raccolta di informazioni dai vicini
  - diffusione delle informazioni raccolte a tutti gli altri nodi della rete
- Noto il grafo della rete ogni nodo calcola le tabelle di routing utilizzando un opportuno algoritmo di routing

Maggior costo computazionale e memoria (memorizza mappa), ma garantire maggior prestazioni alle variazioni della rete



# Raccolta delle informazioni

- Ogni router deve comunicare con i propri vicini ed “imparare” i loro indirizzi
  - **Hello Packet**

(Questo processo è essenziale affinché ciascun router possa “scoprire” i suoi vicini, identificandoli e salvandone gli indirizzi.)
- Deve poi misurare la distanza dai vicini
  - **Echo Packet** Una volta identificati i vicini, ogni router misura la distanza verso ciascun vicino, inviando pacchetti di "Echo" per calcolare il ritardo o la metrica del collegamento, un valore che rappresenta il costo del collegamento in termini di latenza, larghezza di banda, o altri parametri definiti dalla rete.
- In seguito ogni router costruisce un pacchetto con lo stato delle linee (**Link State Packet** o LSP) che contiene
  - la lista dei suoi vicini
  - le lunghezze dei collegamenti per raggiungerli

Dopo aver identificato i vicini e misurato le distanze, il router crea un Link State Packet. Questo pacchetto contiene informazioni dettagliate sullo stato dei collegamenti, inclusa la lista di tutti i vicini e i costi per raggiungerli. LSP viene poi diffuso attraverso la rete in modo che ogni router riceva una copia della topologia aggiornata della rete.



# Diffusione ed elaborazione delle informazioni

- I pacchetti LSP devono essere trasmessi da tutti i router a tutti gli altri router della rete
  - si usa il Flooding
  - a tal fine nel pacchetto LSP occorre aggiungere
    - l'indirizzo del mittente (identifica il router che ha originato il pacchetto)
    - un numero di sequenza
    - una indicazione dell'età del pacchetto
- Avendo ricevuto LSP da tutti i router, ogni router è in grado di costruirsi un'immagine della rete
  - tipicamente si usa l'algoritmo di Dijkstra per calcolare i cammini minimi verso ogni altro router

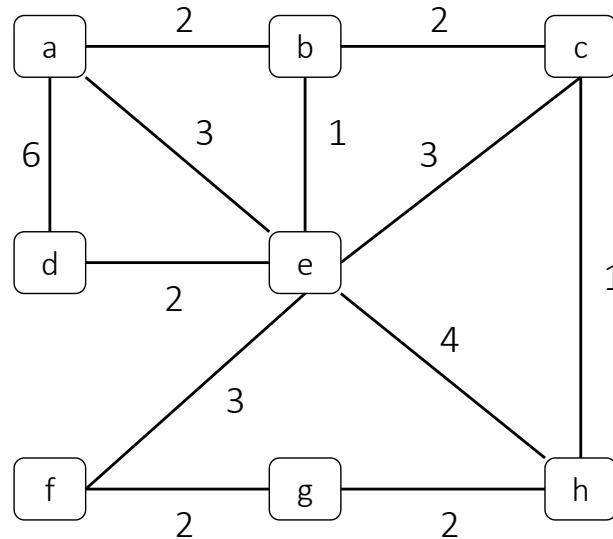
(ogni volta che un router invia un nuovo pacchetto LSP, incrementa questo numero. In questo modo, i router possono distinguere i pacchetti più recenti dai vecchi. Solo i pacchetti con il numero di sequenza più alto vengono mantenuti e propagati)

(Il campo dell'età consente di gestire la scadenza dei pacchetti per evitare che informazioni obsolete restino nella rete: indica il tempo trascorso da quando il pacchetto è stato generato, non è il TTL)

(viene applicato Dijkstra dopo aver ricevuto la mappa della rete)

Se collegamento A-B sparisce, rieseguo algoritmo, senza aspettare che i nodi di rete mi diano delle informazioni ed alla fine invio LSP a tutti i nodi della rete

# Esempio



- Passo 1: il nodo a minima distanza da a è b
- Passo 2: il nodo a minima distanza da a avente b come predecessore è e
- Passo 3: il nodo a minima distanza da a avente e come predecessore è c

È dimostrato che, procedendo in questo modo, le distanze determinate sono minime e non possono essere migliorate nei passi successivi dell'algoritmo.

Nella riga gialla viene riassunta la soluzione dell'algoritmo. Essa riporta la distanza minima verso X ed il nodo “predecessore” di X sul relativo percorso.

Nella riga arancio viene indicata la distanza da A al nodo X e il gateway da A verso X (ossia il nodo a cui A deve inviare i dati per raggiungere X)

 Scrivo in Grigio Scuro la distanza piú piccola tra quelle disponibili e uso la destinazione selezionata come prossimo passo

Determinare il percorso di lunghezza minima dal nodo **a** verso tutti gli altri.

Nelle righe grigio chiare della tabella indichiamo le distanze determinate a quel passo dell'algoritmo

Nelle righe grigie scure indichiamo la distanza che viene ritenuta la migliore determinando il nodo a cui fare riferimento per i calcoli al prossimo passo dell'algoritmo

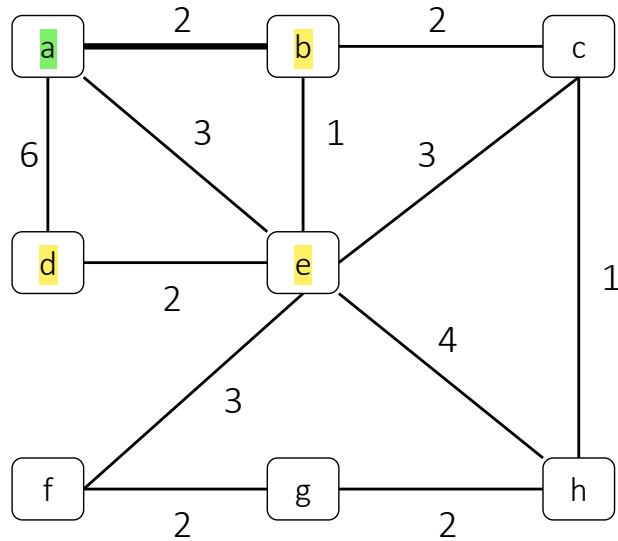
(sarebbero le distanze dal nodo A dei vicini del nodo selezionato per quel passo: quando uno dei vicini ha già la sua distanza migliore, non viene più scritto nelle caselle grigio chiare)

Mantengo tutta la tabella in memoria nel nodo di rete



 = indica nodi disponibili di cui non  
é stato scelto il percorso minimo

## Esempio



Passo 1: il nodo a minima distanza da a è b  
Passo 2: il nodo a minima distanza da a avente b come predecessore è e  
Passo 3: il nodo a minima distanza da a avente e come predecessore è c

.....  
È dimostrato che, procedendo in questo modo le distanze determinate sono minime e non possono essere migliorate nei passi successivi dell'algoritmo

Nella riga gialla viene riassunta la soluzione dell'algoritmo. Essa riporta la distanza minima verso X ed il nodo “predecessore” di X sul relativo percorso

Nella riga arancio viene indicata la distanza da A al nodo X e il gateway da A verso X (ossia il nodo a cui A deve inviare i dati per raggiungere X)

Determinare il percorso di lunghezza minima dal nodo a verso tutti gli altri.

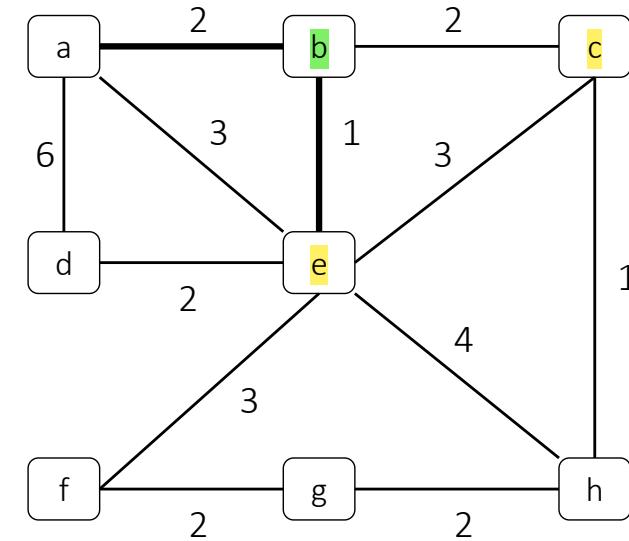
Nelle righe grigio chiare della tabella indichiamo le distanze determinate a quel passo dell'algoritmo

Nelle righe grigio scure indichiamo la distanza che viene ritenuta la migliore determinando il nodo a cui fare riferimento per i calcoli al prossimo passo dell'algoritmo

Scelgo il percorso minimo disponibile (i disponibili sono i vicini di A): tra questi viene scelto B come distanza minima tra i 3.

→ Oltre alla distanza per arrivare a B, ci metto anche il predecessore per arrivare a B, in modo da poter andare a ritroso dalla destinazione (es: per arrivare a B devo passare per A). Le distanze sono calcolate TUTTE dal nodo sorgente A (perché è la tabella di routing del nodo A)

# Esempio



Passo 1: il nodo a minima distanza da  $a$  è

Passo 2: il nodo a minima distanza da a avente b come predecessore è e

Passo 3: il nodo a minima distanza da a avente e come predecessore è c

.....  
È dimostrato che, procedendo in questo modo le distanze determinate sono minime e non possono essere migliorate nei passi successivi dell'algoritmo

Nella riga gialla viene riassunta la soluzione dell'algoritmo. Essa riporta la distanza minima verso X ed il nodo “predecessore” di X sul relativo percorso

Nella riga arancio viene indicata la distanza da A al nodo X e il gateway da A verso X (ossia il nodo a cui A deve inviare i dati per raggiungere X)

A non viene considerato perché sto calcolando i percorsi minimi per il Nodo A. Allora scelgo, tra le caselle bianche, come percorso minimo di questo passo, il nodo E.



Determinare il percorso di lunghezza minima dal nodo a verso tutti gli altri.

Nelle righe grigio chiare della tabella indichiamo le distanze determinate a quel passo dell'algoritmo

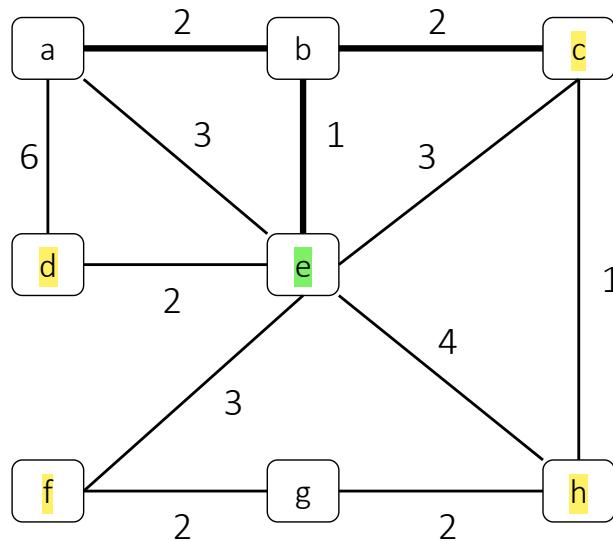
Nelle righe grigio scure indichiamo la distanza che viene ritenuta la migliore determinando il nodo a cui fare riferimento per i calcoli al prossimo passo dell'algoritmo

Spesso, a parità di distanza, alcuni algoritmi di Dijkstra scelgono il nodo in ordine alfabetico o nell'ordine in cui i nodi sono processati.



## Esempio

↓ = indica nodi disponibili di cui non  
é stato scelto il percorso minimo



- Passo 1: il nodo a minima distanza da a è b
- Passo 2: il nodo a minima distanza da a avente b come predecessore è e
- Passo 3: il nodo a minima distanza da a avente e come predecessore è c

.....

È dimostrato che, procedendo in questo modo, le distanze determinate sono minime e non possono essere migliorate nei passi successivi dell'algoritmo

Nella riga gialla viene riassunta la soluzione dell'algoritmo. Essa riporta la distanza minima verso X ed il nodo “predecessore” di X sul relativo percorso

Nella riga arancio viene indicata la distanza da A al nodo X e il gateway da A verso X (ossia il nodo a cui A deve inviare i dati per raggiungere X)

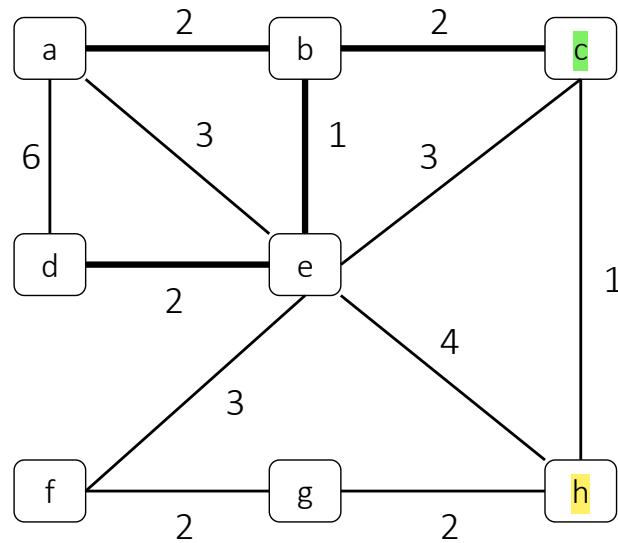
Determinare il percorso di lunghezza minima dal nodo  $a$  verso tutti gli altri.

Nelle righe grigio chiare della tabella indichiamo le distanze determinate a quel passo dell'algoritmo

Nelle righe grigio scure indichiamo la distanza che viene ritenuta la migliore determinando il nodo a cui fare riferimento per i calcoli al prossimo passo dell'algoritmo

Non scelgo B perché ho già trovato il suo percorso minimo da A (grigio scuro). Scelgo, tra quelli disponibili nelle caselle bianche, C.

## Esempio



- Passo 1: il nodo a minima distanza da a è b
- Passo 2: il nodo a minima distanza da a avente b come predecessore è e
- Passo 3: il nodo a minima distanza da a avente e come predecessore è c

.....

È dimostrato che, procedendo in questo modo, le distanze determinate sono minime e non possono essere migliorate nei passi successivi dell'algoritmo

Nella riga gialla viene riassunta la soluzione dell'algoritmo. Essa riporta la distanza minima verso X ed il nodo “predecessore” di X sul relativo percorso

Nella riga arancio viene indicata la distanza da A al nodo X e il gateway da A verso X (ossia il nodo a cui A deve inviare i dati per raggiungere X)

Determinare il percorso di lunghezza minima dal nodo  $a$  verso tutti gli altri.

Nelle righe grigio chiare della tabella indichiamo le distanze determinate a quel passo dell'algoritmo

Nelle righe grigio scure indichiamo la distanza che viene ritenuta la migliore determinando il nodo a cui fare riferimento per i calcoli al prossimo passo dell'algoritmo

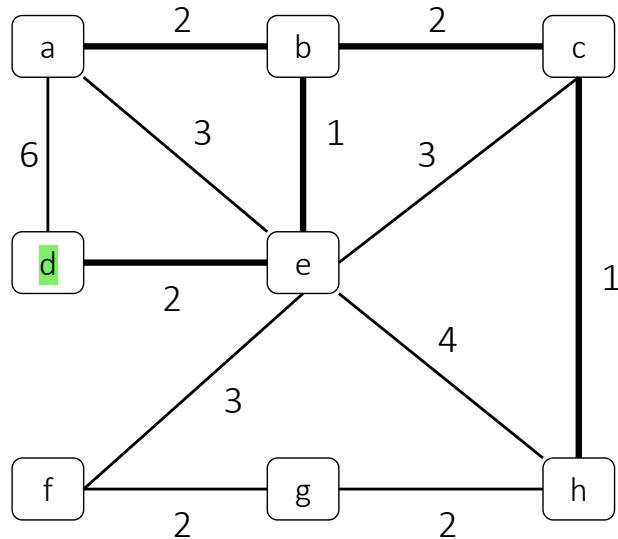
Non scelgo H perché non è il percorso minimo disponibile (ce ne sono altri più piccoli). Allora scelgo D

scelgo D al posto di H

Spesso, a parità di distanza, alcuni algoritmi di Dijkstra scelgono il nodo in ordine alfabetico o nell'ordine in cui i nodi sono processati.



## Esempio



Passo 1: il nodo a minima distanza da a è b

Passo 2: il nodo a minima distanza da a avente b come predecessore è e

Passo 3: il nodo a minima distanza da a avente e come predecessore è c

1

È dimostrato che, procedendo in questo modo, le distanze determinate sono minime e non possono essere migliorate nei passi successivi dell'algoritmo

Nella riga gialla viene riassunta la soluzione dell'algoritmo. Essa riporta la distanza minima verso X ed il nodo “predecessore” di X sul relativo percorso

Nella riga arancio viene indicata la distanza da A al nodo X e il gateway da A verso X (ossia il nodo a cui A deve inviare i dati per raggiungere X)

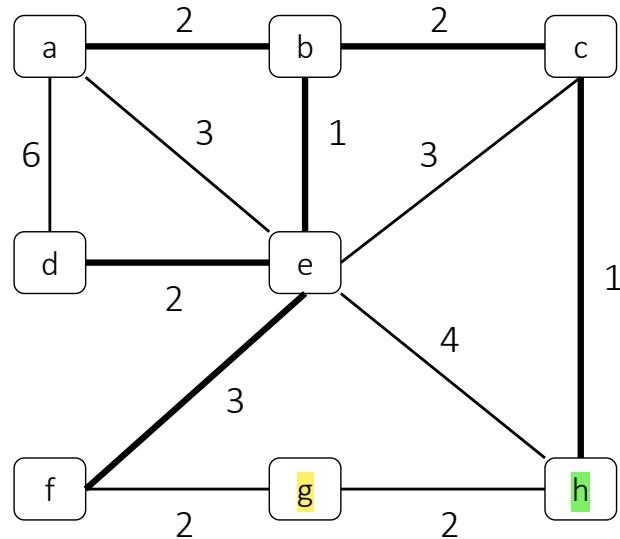
Determinare il percorso di lunghezza minima dal nodo a verso tutti gli altri.

Nelle righe grigio chiare della tabella indichiamo le distanze determinate a quel passo dell'algoritmo

Nelle righe grigio scure indichiamo la distanza che viene ritenuta la migliore determinando il nodo a cui fare riferimento per i calcoli al prossimo passo dell'algoritmo

Scelgo quello con percorso minimo disponibile: scelgo H. A questo passo, non ho aggiunto nessun percorso nella riga bianca, perché tutti i nodi vicini hanno già il percorso minimo scelto

# Esempio



Passo 1: il nodo a minima distanza da a è b  
 Passo 2: il nodo a minima distanza da a avente b come predecessore è e  
 Passo 3: il nodo a minima distanza da a avente e come predecessore è c  
 ....

È dimostrato che, procedendo in questo modo, le distanze determinate sono minime e non possono essere migliorate nei passi successivi dell'algoritmo

Nella riga gialla viene riassunta la soluzione dell'algoritmo. Essa riporta la distanza minima verso X ed il nodo "predecessore" di X sul relativo percorso

Nella riga arancio viene indicata la distanza da A al nodo X e il gateway da A verso X (ossia il nodo a cui A deve inviare i dati per raggiungere X)

Determinare il percorso di lunghezza minima dal nodo a verso tutti gli altri.

Nelle righe grigio chiare della tabella indichiamo le distanze determinate a quel passo dell'algoritmo

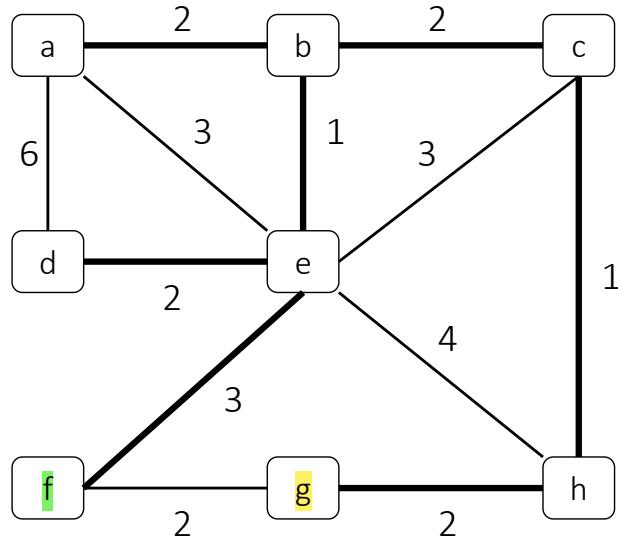
Nelle righe grigio scure indichiamo la distanza che viene ritenuta la migliore determinando il nodo a cui fare riferimento per i calcoli al prossimo passo dell'algoritmo

Scelgo F perché ha il percorso minimo più basso rispetto a G



A	B	C	D	E	F	G	H
	A 2			A 6	A 3		
	<b>A 2</b>						
		B 4			B 3		
	<b>A 2</b>				<b>B 3</b>		
		E 6	E 5			E 6	
	<b>A 2</b>	<b>B 4</b>			<b>B 3</b>		
							E 7
	<b>A 2</b>	<b>B 4</b>	<b>E 5</b>		<b>B 3</b>		
							C 5
	<b>A 2</b>	<b>B 4</b>	<b>E 5</b>	<b>B 3</b>			
							<b>C 5</b>
	<b>A 2</b>	<b>B 4</b>	<b>E 5</b>	<b>B 3</b>	<b>E 6</b>		
							H 7
	<b>A 2</b>	<b>B 4</b>	<b>E 5</b>	<b>B 3</b>	<b>E 6</b>		
							<b>C 5</b>

# Esempio



Determinare il percorso di lunghezza minima dal nodo a verso tutti gli altri.

Nelle righe grigio chiare della tabella indichiamo le distanze determinate a quel passo dell'algoritmo

Nelle righe grigio scure indichiamo la distanza che viene ritenuta la migliore determinando il nodo a cui fare riferimento per i calcoli al prossimo passo dell'algoritmo

Scelgo G con distanza 7 da H perché è il percorso minimo più piccolo rispetto a quella da F

Passo 1: il nodo a minima distanza da a è b  
 Passo 2: il nodo a minima distanza da a avente b come predecessore è e  
 Passo 3: il nodo a minima distanza da a avente e come predecessore è c  
 .....  
 È dimostrato che, procedendo in questo modo, le distanze determinate sono minime e non possono essere migliorate nei passi successivi dell'algoritmo

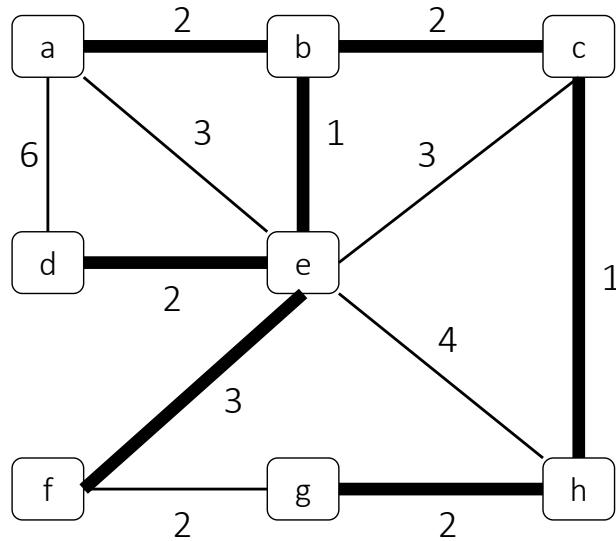
Nella riga gialla viene riassunta la soluzione dell'algoritmo. Essa riporta la distanza minima verso X ed il nodo "predecessore" di X sul relativo percorso

Nella riga arancio viene indicata la distanza da A al nodo X e il gateway da A verso X (ossia il nodo a cui A deve inviare i dati per raggiungere X)

↓

A	B	C	D	E	F	G	H
	A 2		A 6	A 3			
	<b>A 2</b>						
		B 4		B 3			
	<b>A 2</b>	<b>B 4</b>		<b>B 3</b>			
			E 6	E 5	E 6		E 7
	<b>A 2</b>	<b>B 4</b>	<b>E 5</b>	<b>B 3</b>			
							C 5
	<b>A 2</b>	<b>B 4</b>	<b>E 5</b>	<b>B 3</b>			<b>C 5</b>
							H 7
	<b>A 2</b>	<b>B 4</b>	<b>E 5</b>	<b>B 3</b>	<b>E 6</b>		<b>C 5</b>
							F 9
	<b>A 2</b>	<b>B 4</b>	<b>E 5</b>	<b>B 3</b>	<b>E 6</b>	<b>H 7</b>	<b>C 5</b>

# Esempio



Passo 1: il nodo a minima distanza da a è b  
 Passo 2: il nodo a minima distanza da a avente b come predecessore è e  
 Passo 3: il nodo a minima distanza da a avente e come predecessore è c  
 ....  
 È dimostrato che, procedendo in questo modo, le distanze determinate sono minime e non possono essere migliorate nei passi successivi dell'algoritmo

Nella riga gialla viene riassunta la soluzione dell'algoritmo. Essa riporta la distanza minima verso X ed il nodo "predecessore" di X sul relativo percorso

Nella riga arancio viene indicata la distanza da A al nodo X e il gateway da A verso X (ossia il nodo a cui A deve inviare i dati per raggiungere X)

Riga Gialla: Utilizzata da A per calcolare e memorizzare le distanze minime e i predecessori alla destinazione, ma non viene inclusa nel pacchetto LSP.  
 Riga Arancione: Utilizzata da A per determinare il next hop per ciascuna destinazione, ma non viene inclusa nel pacchetto LSP

Determinare il percorso di lunghezza minima dal nodo a verso tutti gli altri.

Nelle righe grigio chiare della tabella indichiamo le distanze determinate a quel passo dell'algoritmo

Nelle righe grigio scure indichiamo la distanza che viene ritenuta la migliore determinando il nodo a cui fare riferimento per i calcoli al prossimo passo dell'algoritmo

In pratica, nella riga gialla, viene copiata l'ultima riga grigia scura, mentre nella riga arancio, viene messa la distanza minima con il primo nodo "vicino" per il nodo A, che usa per arrivare alla destinazione (Ad esempio, per andare a D, uso B come "Gateway" per arrivare ad E e poi a D)

A	B	C	D	E	F	G	H
	A 2		A 6	A 3			
	A 2		B 4	B 3			
	A 2		E 6	E 5	B 3	E 6	E 7
	A 2	B 4					
	A 2	B 4	E 5	B 3			C 5
	A 2	B 4	E 5	B 3			C 5
	A 2	B 4	E 5	B 3			H 7
	A 2	B 4	E 5	B 3	E 6		C 5
	A 2	B 4	E 5	B 3	E 6	H 7	C 5
	A 2	B 4	E 5	B 3	E 6	F 9	
	A 2	B 4	E 5	B 3	E 6	H 7	C 5
	A 2	B 4	E 5	B 3	E 6	H 7	C 5
	A 2	B 4	E 5	B 3	E 6	B 7	B 5

H precede G nel percorso ottimo da A a G

B è il primo nodo che A incontra sul percorso ottimo verso G



# Il router IP

- Il nodo di commutazione nelle reti IP viene detto **router**
- Il router è un nodo di commutazione a pacchetto specializzato per l'utilizzo del protocollo IP
- Nonostante siano tutti identificati con il termine **router** i nodi di commutazione della rete Internet possono essere fra loro molto diversi



# Classificazione dei router

Simile a quello che ho in casa

## • SOHO (Small Office and HOrne) router

- Utilizzo domestico o piccoli uffici
- Interfaccia sulla LAN (switch con poche porte Fast Ethernet 100Mbit/s e wi-fi)  
Dotato di una semplice interfaccia LAN con switch integrato, poche porte Ethernet (generalmente Fast Ethernet a 100 Mbit/s) e connettività Wi-Fi. Ideale per reti con pochi dispositivi e bassa complessità.

## • Router di accesso

- used by ISPs to provide access service Utilizzato dai provider di servizi Internet (ISP) per garantire l'accesso a Internet agli utenti finali.
- large number of medium-low speed ports (50 kbps ÷ 10 Mbps)
- capabile of several protocols and access technologies (PPP, SLIP, ADSL, FTTx, ...)

## • Enterprise/campus router (devono fare multiplexing di porte a basse velocità in alta velocità)

- Interconnessione fra LAN per organizzazioni di medie dimensioni
- Poche porte ad elevata velocità (Fast o Gigabit Ethernet)

## • Backbone router (si occupano di tratti a lunghe distanze)

- Per reti di trasporto e connessioni inter-domain
- Piccolo numero di porte ad elevata velocità (>= 1Gbps)
- Equipaggiato con sistemi di garanzia dell'affidabilità (ridondanza, monitoraggio remoto, ecc.)

Dispone di un numero ridotto di porte ad alta velocità per gestire la connettività interna e l'intercomunicazione delle sottoreti.



# Le 4 funzioni dei Router

- **Routing** (Completare la Tabella di Instradamento per eseguire Forwarding)
  - Scambio di informazioni con altri router (IGP/EGP)
  - Elaborazione locale (routing algorithm) (Esecuzione di Algoritmi per Calcolare Percorsi Minimi)
  - Popolazione delle tabelle di routing
- **Forwarding**
  - IP
    - Table lookup → Quando un pacchetto arriva, il router esegue una ricerca nella tabella di routing per determinare il prossimo salto verso la destinazione.
    - Header update → Il router può modificare l'intestazione del pacchetto
- **Switching**
  - Trasferimento del datagramma da interfaccia di input a interfaccia di output
- **Trasmissione** (Implementata da singola Interfaccia di Rete)
  - Trasmissione del datagramma sul mezzo fisico (utilizzando l'interfaccia di rete di output) Una volta che il pacchetto è stato inoltrato all'interfaccia di output, il router si occupa di trasmettere il datagramma attraverso il mezzo fisico

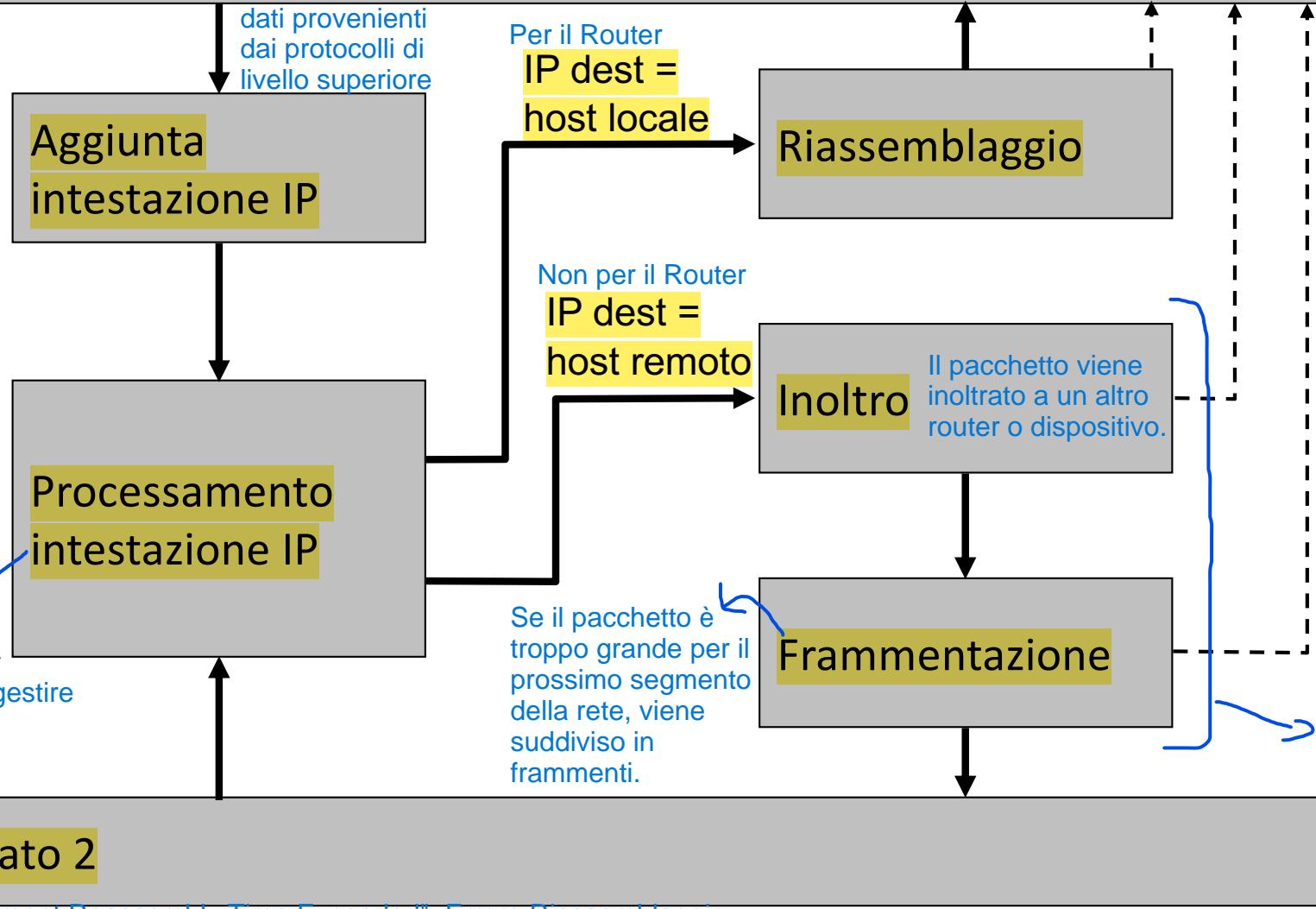
I Higher Layer Protocols sono protocolli che operano sopra il livello IP e si occupano di gestire aspetti specifici della trasmissione e del controllo dei dati all'interno della rete.



# Schema funzionale di un router

Higher layer protocols (TCP, UDP, ICMP, OSPF, ...)

Nel router, esiste un Livello di Protocolli Applicativi usati dal Piano di Controllo: cui non solo coordina la configurazione e l'aggiornamento delle tabelle di routing, ma supervisiona anche la comunicazione con altri router e gestisce gli errori, fornendo così un controllo complessivo della rete.



"Fragment Reassembly Time Exceeded": Errore Riassemblaggio

"Time Exceeded" e "Destination Unreachable": Errore Inoltro

"Don't Fragment" e "Fragmentation Needed and DF Set": Errore Frammentazione



# Tabella di Routing e di forwarding

## • Routing table

- result of the routing protocols and algorithms
- each entry includes route prefix, next hop and metric
- also called Routing Information Base (RIB)

In sintesi, la tabella di routing aiuta a costruire la mappa della rete, ma non si occupa di inviare fisicamente i dati: è come un "elenco di strade" per sapere quali percorsi esistono e quanto sono "costosi".

La metrica nella tabella di routing rappresenta il "costo" del percorso verso una destinazione ed è il parametro su cui si basano i protocolli di routing per calcolare i percorsi minimi o ottimali.

La tabella di routing è il risultato delle operazioni di protocollo di routing e degli algoritmi che calcolano i percorsi migliori per raggiungere ciascuna destinazione nella rete.

## • Forwarding table

- built upon routing table content (complete or partial)
- each entry includes also the output interface
- used to actually forward datagram
- optimized for fast table lookup
- also called Forward Information Base (FIB)

La tabella di forwarding è progettata per essere rapida nell'accesso e consente di inoltrare pacchetti a velocità elevate, ottimizzando l'efficienza della rete.

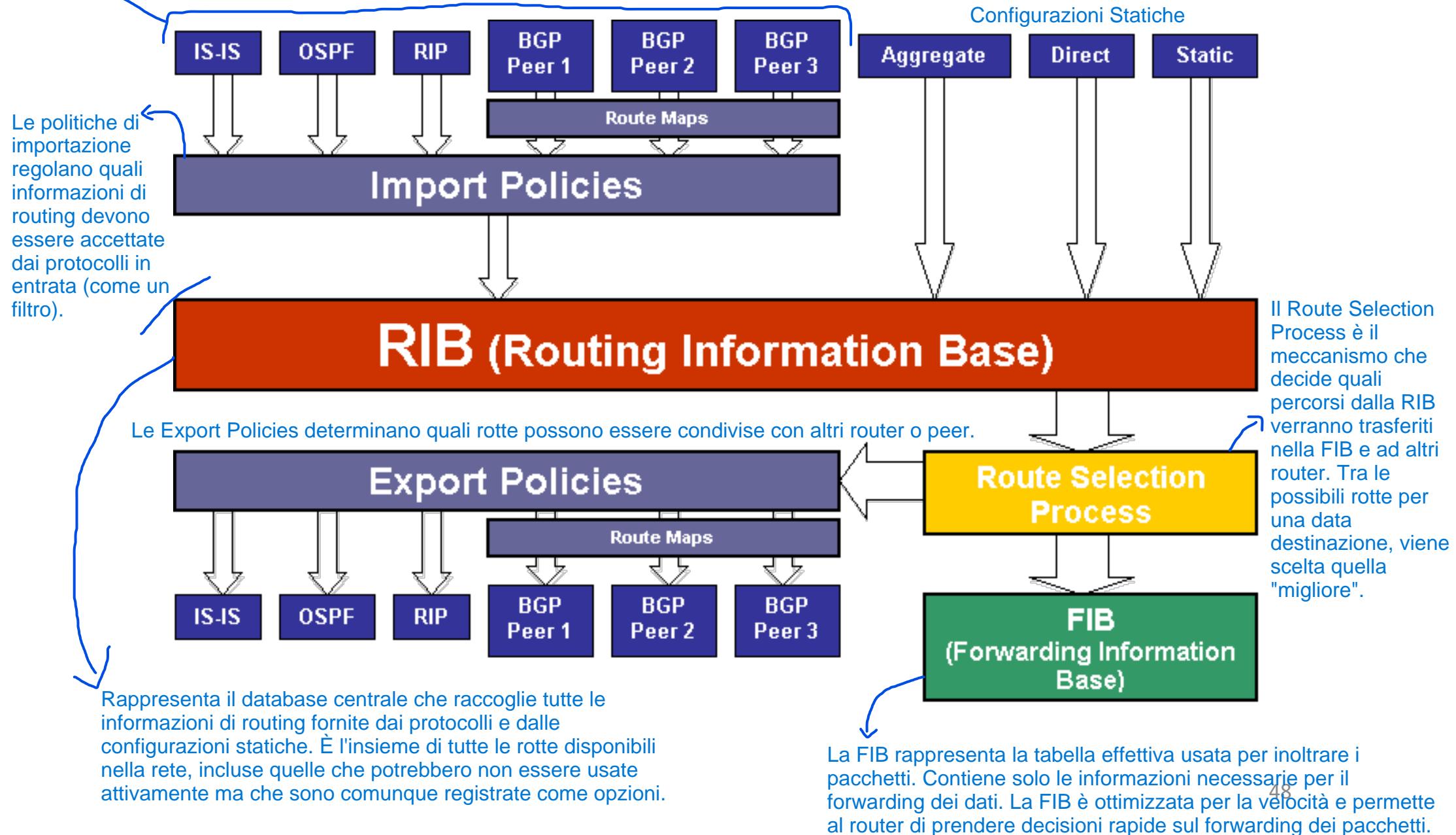
Ogni entry nella FIB include router prefix, next hop, Metric e Output Interface

La tabella di forwarding viene costruita a partire dalla tabella di routing, ma contiene solo l'informazione necessaria per inoltrare effettivamente i pacchetti. In altre parole, la FIB ottimizza e riduce le informazioni della tabella di routing per agevolare un accesso rapido e diretto.



Forniscono informazioni di routing diverse a seconda della topologia e dei percorsi di rete. Ognuno di questi protocolli ha il suo metodo di calcolo delle rotte ottimali e delle metriche, che vengono poi processate e aggregate nella Routing Information Base (RIB).

# Routing vs. forwarding table





# Arrivare alla FIB

- La RIB è una base dati che viene compilata con
  - il concorso di numerosi protocolli
  - diverse strategie di sintesi delle informazioni note
- La FIB si ottiene a partire dalle informazioni della RIB
  - Vengono utilizzati opportuni algoritmi
- Nel complesso queste operazioni determinano la strategia di instradamento utilizzata dai nodi della rete



ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

# Instradamento nell'Internet globale

Franco CALLEGATI

Dipartimento di Informatica - Scienza e Ingegneria

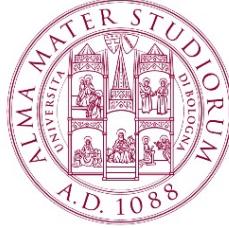


# Routing gerarchico

In Internet, si utilizza una struttura di routing gerarchico per gestire l'enorme numero di reti globali in modo organizzato.

- In Internet si usa il routing gerarchico e le aree di routing sono chiamate **Autonomous System** (AS)
  - un AS può essere ulteriormente suddiviso in porzioni dette **Routing Area** (RA) interconnesse da un **backbone** (dorsale)
    - tradizionalmente secondo la classe, oggi secondo il CIDR
  - gli AS decidono autonomamente i protocolli e le politiche di routing che intendono adottare al loro interno
  - i vari enti di gestione si devono accordare su quali protocolli utilizzare per il dialogo tra i router che interconnettono AS diversi (È necessario che i vari enti di gestione dei diversi AS concordino i protocolli da utilizzare per garantire un dialogo efficace tra i router che collegano AS differenti)
- I protocolli di routing all' interno di un AS sono detti **Interior Gateway Protocol** (IGP)
- I protocolli di routing fra AS sono detti **Exterior Gateway Protocol** (EGP)

Ogni rete IP appartiene a un singolo AS o RA, garantendo una gestione coerente del routing per quella rete.

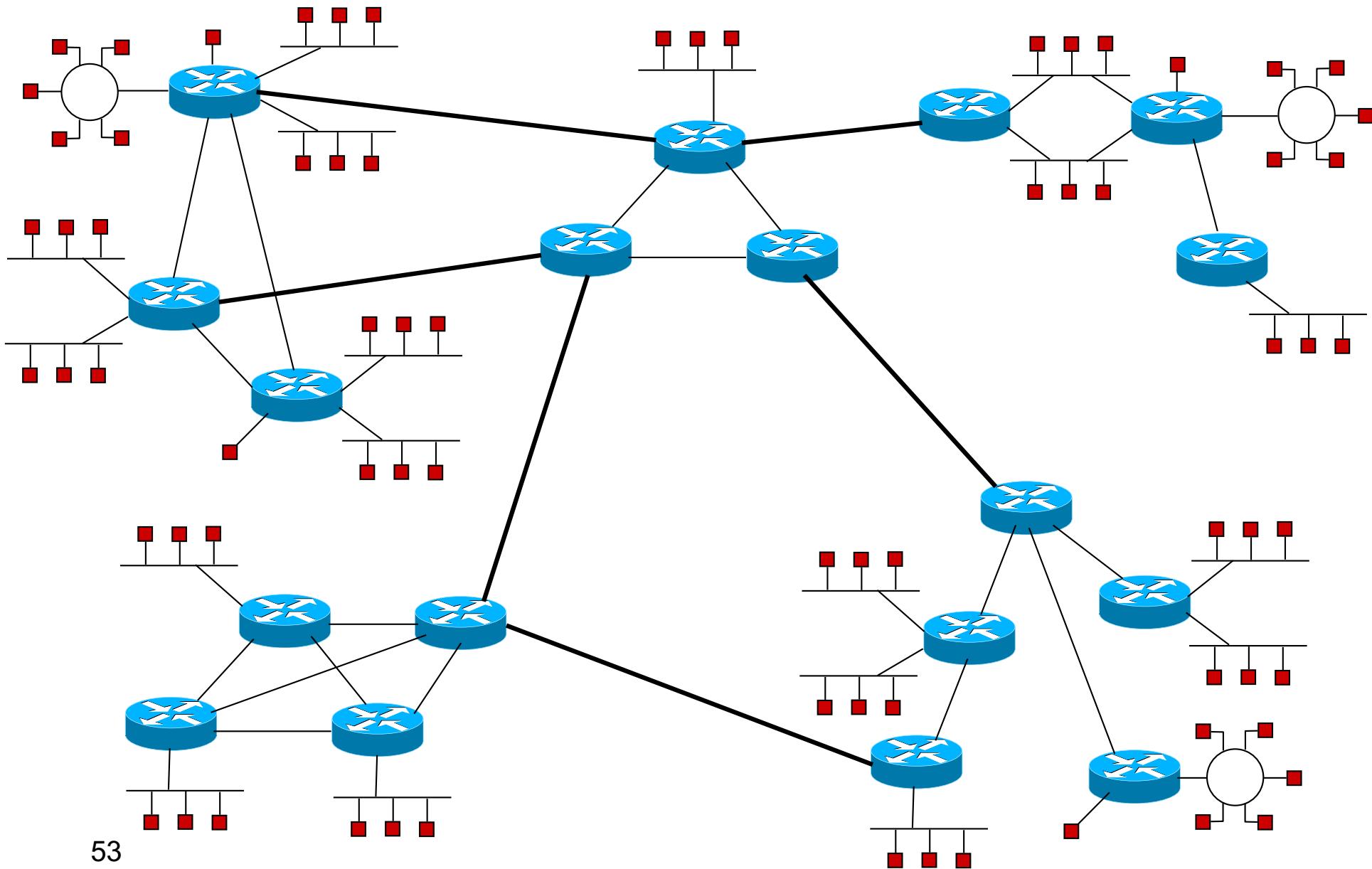


ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

# Authonomous Systems and peering

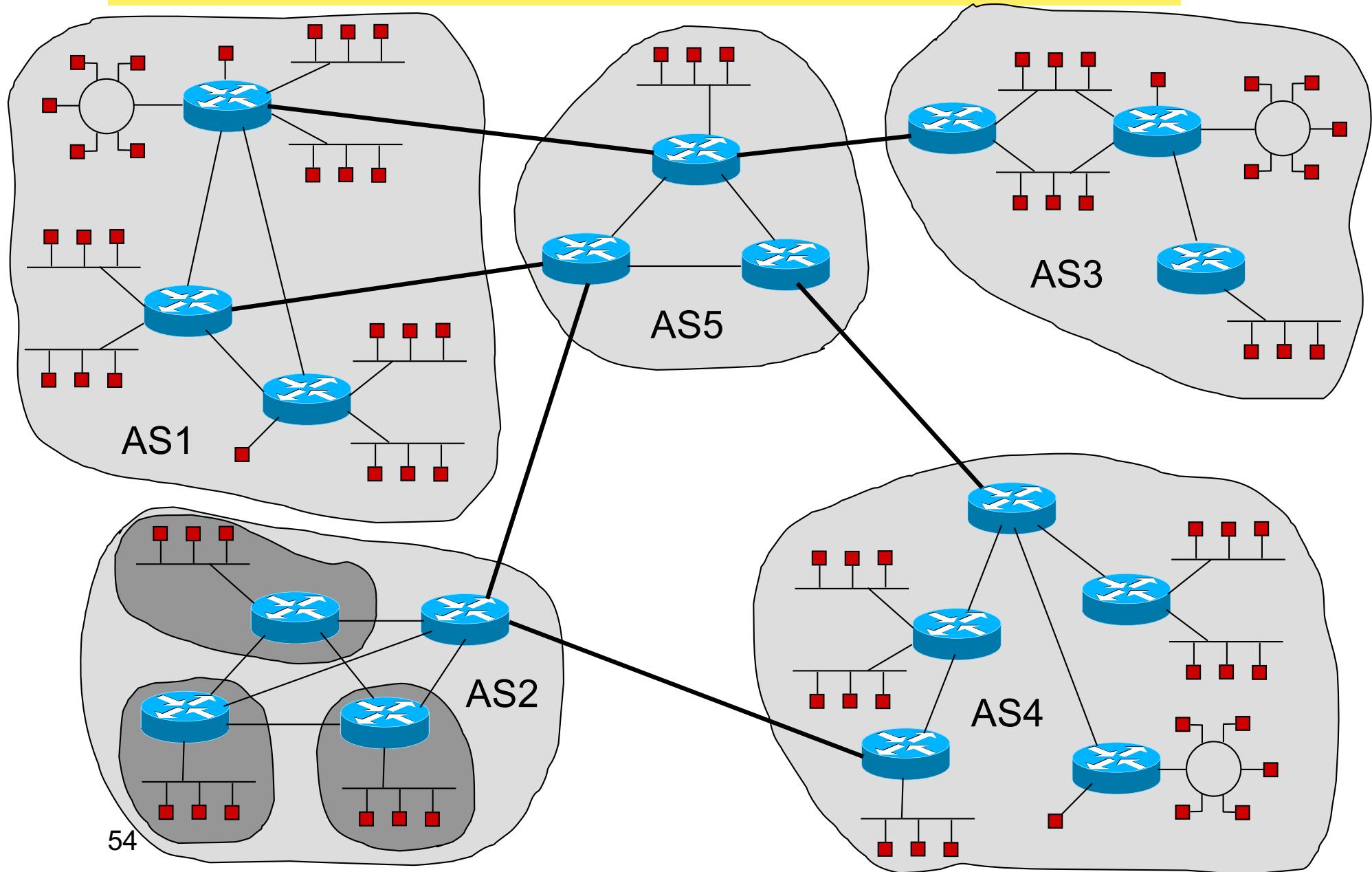


# Internet = rete di reti



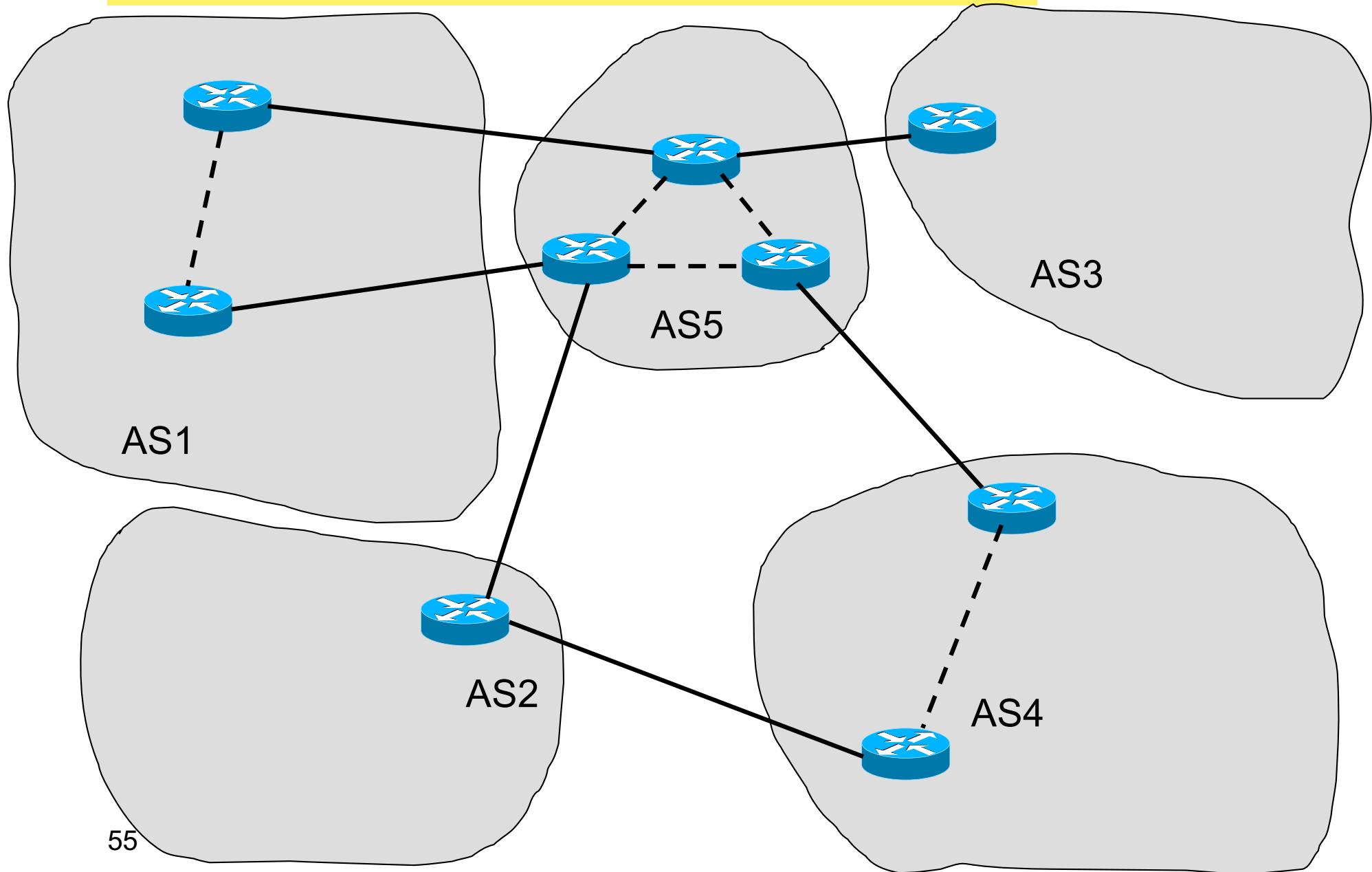


# Internet = sistemi interconnessi





# Internet: grafo semplificato





Questo approccio a "compartimenti separati" consente di limitare il carico computazionale e la complessità di gestione, mantenendo ogni AS responsabile del proprio traffico e delle proprie politiche di instradamento, semplificando il routing globale tra milioni di reti.

# Il routing a livello globale

Internet è suddiviso in una serie di sottoinsiemi autonomi di rete, ciascuno responsabile dell'instradamento all'interno della propria area.

- **Routing gerarchico:**

- Identificazione di sottoinsiemi di rete autonomi per quanto riguarda l'instradamento
- Identificazione di punti di contatto fra i sottoinsiemi

- **Due tipi di grafo**

La topologia della rete gerarchica è rappresentata attraverso due tipi di grafi:

- **Topologia dei sottoinsiemi della rete**

- Grafi di dettaglio

- **Topologie dei sottoinsiemi interconnessi**

- Grafo semplificato

- I sottoinsiemi sono i nodi
- I collegamenti fra i sottoinsiemi sono gli archi

- **A ciascun livello non si ha conoscenza dell'altro**

Questo grafo rappresenta la topologia dettagliata all'interno di ciascun AS o sottoinsieme, mostrando tutte le connessioni tra i nodi (router e switch) all'interno dell'AS.

Rappresenta la topologia tra i vari sottoinsiemi interconnessi (ovvero, tra gli AS). A differenza del grafo di dettaglio, qui i nodi sono rappresentati in modo aggregato come singoli blocchi (gli AS) e gli archi (i collegamenti tra AS) rappresentano solo le connessioni principali tra questi blocchi, omettendo i dettagli del routing interno.

Gli AS agiscono come entità indipendenti, con ciascuno che gestisce il proprio routing interno.

Tuttavia, per garantire la comunicazione tra AS diversi, esistono punti di connessione o punti di contatto. Questi punti permettono il passaggio del traffico da un AS all'altro.

A livello interno (grafo di dettaglio), i router all'interno di un AS non hanno una visione diretta delle connessioni tra gli AS, vedendo solo la propria topologia interna.  
A livello globale (grafo semplificato), gli AS vedono solo le connessioni principali tra i vari domini, senza conoscere la topologia dettagliata all'interno di ognis AS.



# Authonomous Systems

- I sottoinsiemi in cui viene suddivisa logicamente la rete Internet sono detti

## Authonomous Systems (AS)

(identificati da numero progressivo)

- Cosa è un AS?
- La definizione classica di AS è
  - un insieme di router gestiti da un'unica amministrazione
  - che utilizza
    - un solo protocollo di routing
    - un'unica logica per definire le metriche
- Questa definizione *era applicabile* nella prima fase di sviluppo di Internet ma è diventata *troppo limitata* con l'evolversi della rete



cioè che l'AS utilizza un criterio uniforme per valutare e determinare i migliori percorsi di instradamento all'interno della propria rete.

Diversità dei Protocolli e delle Politiche: Un singolo AS può oggi contenere reti diverse, ognuna con i propri requisiti e protocolli. Ad esempio, alcune reti interne potrebbero usare OSPF, altre IS-IS, pur appartenendo allo stesso AS.

Multipli Domini di Routing Interno: Un AS può essere suddiviso in più aree o domini di routing, ciascuno con una propria logica di routing interno, ma con una visione unificata verso l'esterno (es: UniBo fa parte di una AS più grande che contiene molte Università diverse).

Politiche di Routing Complesse: L'esigenza di implementare politiche diverse per la gestione del traffico in entrata e in uscita, e per i collegamenti con AS esterni, ha portato a una gestione più complessa del routing.



# I protocolli di routing

Un Autonomous System (AS) deve gestire il traffico al suo interno, quindi è necessario un protocollo per instradare i pacchetti tra i vari router all'interno dell'AS.

- Un AS deve implementare il routing al suo interno
  - Lo fa utilizzando uno o più protocolli di routing detti Interior Gateway protocols (IGP)
- Un AS deve comunicare con gli altri AS per implementare il routing fra AS
  - Lo fa utilizzando un protocollo di routing pensato appositamente detto Exterior Gateway Protocol (EGP)
- Interior Gateway Protocol
  - RIP: Routing Information Protocol (Implementa Distance Vector)
  - OSPF: Open Shortest Path First (Implementa il Link State)
- Exterior Gateway Protocol
  - EGP: Exterior Gateway Protocol
  - BGP: Border Gateway Protocol

Un Autonomous System (AS) è un insieme di network IP che rispondono a criteri comuni di gestione e routing. All'interno di questo AS sono definite le politiche di instradamento che regolano il flusso dei pacchetti tra le network IP al suo interno e tra le network IP di un AS e quelle di altri AS.



# RFC 1930

- L'evoluzione di Internet e l'introduzione del CIDR richiedono una definizione più estensiva dell'AS
  - Oggi un AS è
    - Un insieme di prefissi di rete IP (network IP definite secondo la logica CIDR)
    - Gestito in modo unitario e con una ben definita politica di routing
      - Questo significa che chi gestisce l'AS ha definito in modo chiaro al suo interno come raggiungere le network IP
  - Quindi l'AS
    - Può
      - Avere uno o più enti gestori
      - Utilizzare una o più tecnologie
    - Ma deve
      - Avere un'unica logica che garantisce la connettività con il resto del mondo
- Ogni AS gestisce un gruppo di indirizzi IP organizzati in prefissi CIDR, che rendono la struttura degli indirizzi più flessibile e permettono di aggregare più reti in blocchi contigui.
- Gestione unitaria: Tutte le reti IP all'interno di un AS sono gestite come una singola entità amministrativa. Questo significa che l'AS opera come una struttura coerente con regole comuni per l'instradamento, anche se può includere molte reti separate (o sottoreti).
- Politica di routing definita: L'ente che gestisce l'AS decide internamente come raggiungere ciascuna delle proprie reti IP. Questo implica che esiste una politica di routing ben stabilita, con un protocollo o una combinazione di protocolli (come OSPF o IS-IS) che dirigono il traffico verso le destinazioni corrette all'interno dell'AS.
- Nonostante la flessibilità nella gestione e nelle tecnologie usate, l'AS deve mantenere una logica di instradamento unificata verso l'esterno. Questo significa che, pur gestendo internamente le rotte in vari modi, l'AS deve apparire come un'unica entità agli altri AS.



# Esempio

- Università di Bologna -> 137.204.0.0/16
- Politecnico di Torino -> 130.192.0.0/16
- Entrambi
  - sono connessi al GARR, la rete italiana degli enti di ricerca
  - comunicano con il resto del mondo tramite il GARR
- Non c'è bisogno di avere un AS per ogni ateneo e infatti il GARR (e tutte le reti connesse ad esso) costituiscono un unico AS (AS137)



# Internet Routing Registries

- Database contenenti le politiche di routing degli AS

Query the RADb:   Advanced Options

Register Now   Features   Support   FAQ   Contact Us   Log In

Advanced Query

Query the RADb: AS137  Advanced Options

```
aut-num: AS137
as-name: ASGARR
descr: Consortium GARR
org: ORG-GIRal-RIPE
import: from AS20965 action pref=300; accept ANY
import: from AS1299 action pref=100; accept ANY
mp-import: afi ipv4.multicast from AS20965 action pref=100; accept ANY
mp-import: afi ipv6.unicast from AS20965 action pref=100; accept ANY
mp-import: afi ipv6.multicast from AS20965 action pref=100; accept ANY
export: to AS20965 announce AS-GARRTOGEANT
export: to AS1299 announce AS-GARR
mp-export: afi ipv4.multicast to AS20965 announce AS-GARRTOGEANT;
mp-export: afi ipv6.unicast to AS20965 announce AS-GARRTOGEANT;
mp-export: afi ipv6.multicast to AS20965 announce AS-GARRTOGEANT;
admin-c: DUMMY-RIPE
tech-c: DUMMY-RIPE
status: LEGACY
mnt-by: RIPE-NCC-LEGACY-MNT
mnt-by: GARR-LIR
created: 2002-08-21T13:03:42Z
last-modified: 2018-06-25T06:43:36Z
source: RIPE
remarks: *****
```



# AS 137

```
aut-num: AS137
as-name: ASGARR
descr: Consortium GARR
org: ORG-GIRaI-RIPE
import: from AS20965 action pref=300; accept ANY
import: from AS1299 action pref=100; accept ANY
mp-import: afi ipv4.multicast from AS20965 action pref=100; accept ANY
mp-import: afi ipv6.unicast from AS20965 action pref=100; accept ANY
mp-import: afi ipv6.multicast from AS20965 action pref=100; accept ANY
export: to AS20965 announce AS-GARRTOGEANT
export: to AS1299 announce AS-GARR
mp-export: afi ipv4.multicast to AS20965 announce AS-GARRTOGEANT;
mp-export: afi ipv6.unicast to AS20965 announce AS-GARRTOGEANT;
mp-export: afi ipv6.multicast to AS20965 announce AS-GARRTOGEANT;
admin-c: DUMY-RIPE
tech-c: DUMY-RIPE
status: LEGACY
mnt-by: RIPE-NCC-LEGACY-MNT
mnt-by: GARR-LIR
created: 2002-08-21T13:03:42Z
last-modified: 2018-06-25T06:43:36Z
source: RIPE
remarks: ****
remarks: * THIS OBJECT IS MODIFIED
remarks: * Please note that all data that is generally regarded as personal
remarks: * data has been removed from this object.
remarks: * To view the original object, please query the RIPE Database at:
remarks: * http://www.ripe.net/whois
remarks: ****
```

## Regole di Import:

Da quali AS posso ricevere informazioni di routing  
(con scambio di path vector BGP ad esempio)

## Regole di Export:

A quali AS comunico informazioni di routing  
(inviando path vector BGP ad esempio)



# AS20965 Regole di Import

```
import:      from AS137 accept AS-GARRTOGEANT
import:      from AS378 accept AS-MACHBA
import:      from AS559 accept AS-SWITCH and AS-CERNEXT
import:      from AS680 accept AS-DFNTOWINISP
import:      from AS766 accept AS-REDIRIS {192.243.16.0/22, 192.171.2.0/24}
import:      from AS786 accept AS-JANETEURO
import:      from AS1103 accept AS-SURFNET
import:      from AS1213 accept AS-HEANET
import:      from AS1853 accept AS-ACONET and AS-ACOSERV and AS-ACONET-STH
import:      from AS1930 accept AS-RCCN
import:      from AS1955 accept AS-HBONE
import:      from AS2107 accept AS-ARNES
import:      from AS2108 accept AS-CARNet
remarks:    AS7500 (DNS root name-server) is behind RENATER
import:      from AS2200 accept AS-RENATER AS7500
import:      from AS2602 accept AS-RESTENA
import:      from AS2603 accept AS-NORDUNET
import:      from AS2607 accept AS-SANET2
import:      from AS2611 accept AS-BELNET
import:      from AS2614 accept AS-ROEDUNET AS9199
import:      from AS2847 accept AS-LITNET
import:      from AS2852 accept AS2852 {130.129.0.0/16}
import:      from AS3208 accept AS3208
import:      from AS3221 accept AS3221
import:      from AS3268 accept AS3268 AS198336
import:      from AS5379 accept AS5379
import:      from AS5408 accept AS5408:AS-TO-GEANT
import:      from AS5538 accept AS-SigmaNet-Geant
import:      from AS6802 accept AS-ISTF
import:      from AS6879 accept AS6879
import:      from AS8501 accept AS-PLNET
import:      from AS8517 accept AS-ULAKNET
import:      from AS12046 accept AS-RICERKANET
import:      from AS12687 accept AS-URAN-GEANT
import:      from AS13092 accept AS13092
import:      from AS35385 accept AS35385
import:      from AS35656 accept AS35656
import:      from AS21274 accept AS-BASNET
import:      from AS40981 accept AS40981
import:      from AS57965 accept AS57965 and AS-PALNREN
import:      from AS202993 accept AS202993
```

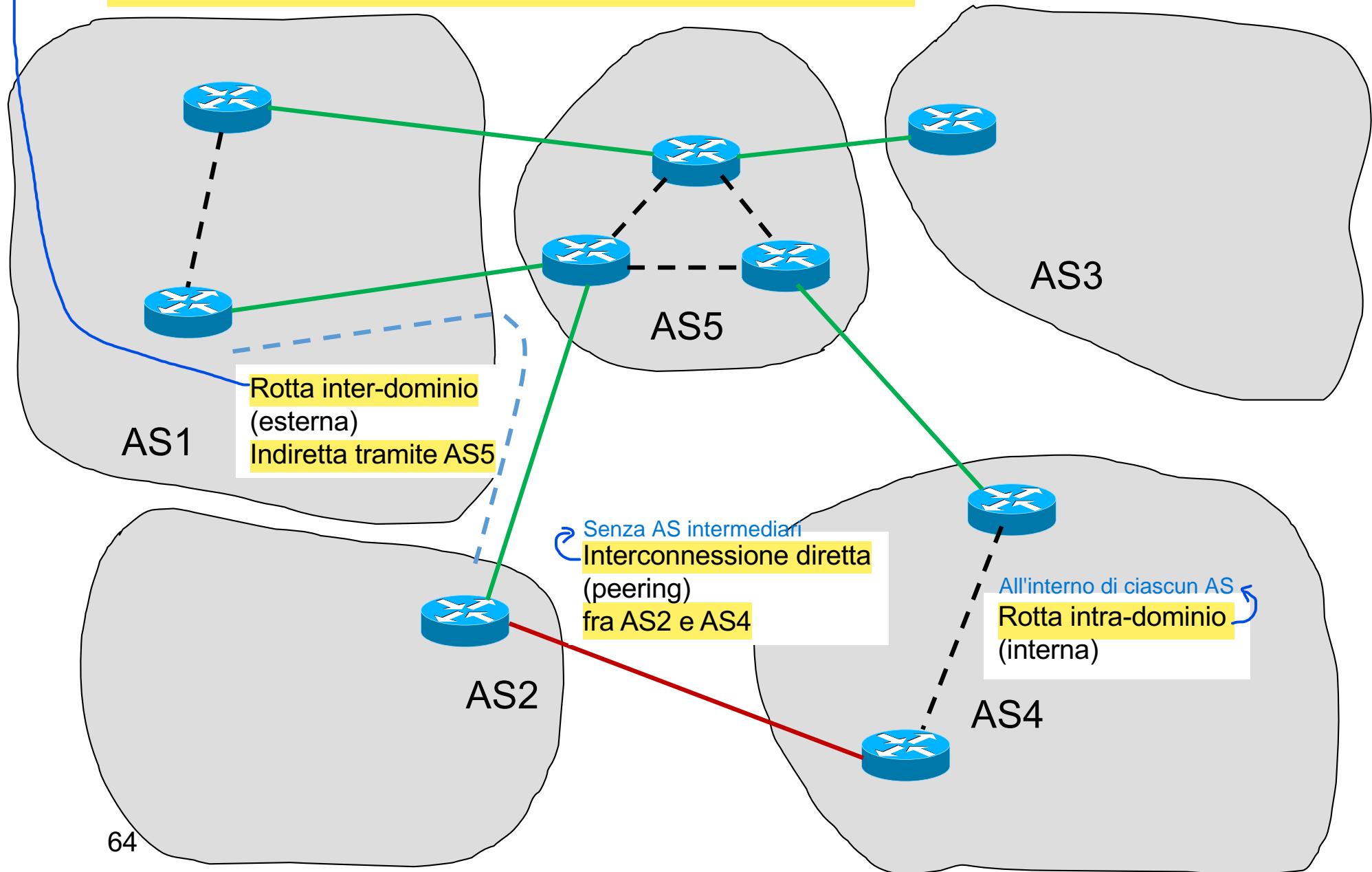
GEANT è la rete degli enti di ricerca  
Europea  
Ha interconnessioni con le principali  
reti mondiali  
Importa ed esporta informazioni di  
routing verso numerosi AS

1. Le network IP di GARR sono inviate a GEANT
2. GEANT le invia alle altre reti di trasporto mondiali

Rotta inter-dominio (esterna): Questa rottura collega reti situate in AS diversi. Poiché ogni AS rappresenta un'entità amministrativa indipendente, la rottura inter-dominio permette la comunicazione tra AS passando tramite un AS intermedio



# Interconnessione fra AS





# Internet Service Provider

- Un Internet Service Provider (ISP) è un'organizzazione che fornisce servizi per l'utilizzo di Internet
  - Servizi:
    - Connettività
    - Web, mail hosting
    - Registrazione e noleggio di numeri IP e nomi di dominio
    - ...
  - Dal punto di vista giuridico un ISP può essere:
    - Privato con finalità di lucro
    - Privato senza finalità di lucro
    - In forma cooperativa
      - enti pubblici
      - ...
  - Tipicamente un ISP si registra come AS
- Questo tipo di ISP è un'azienda privata che opera con l'obiettivo di generare profitto.  
ES: Telecom Italia
- Gli ISP senza scopo di lucro forniscono servizi Internet senza l'obiettivo di generare profitti per gli investitori.  
ES: NREN
- Gli ISP in forma cooperativa sono posseduti e gestiti dai loro membri, che possono essere individui, famiglie, aziende o enti locali.  
ES: Guifi.net



# Internet region

- Gli AS non sono necessariamente vincolate ad aree geografiche e/o confini nazionali
- Internet region
  - Una porzione di Internet contenuta in una specifica area geografica
    - Tipicamente una nazione o un insieme di nazioni
- Relazione fra Internet Region e ISP
  - Un'Internet Region è solitamente servita da più ISP
  - Uno stesso ISP può servire più Internet Region



# Classificazione degli ISP

- **Tier 1 ISP** Questi ISP non hanno bisogno di acquistare transito da altri provider poiché possiedono percorsi diretti verso tutte le destinazioni.
  - Un ISP che all'interno di una “Internet Region” raggiunge tutte le reti senza accedere a servizi a pagamento di altri
  - In breve un soggetto che possiede un'infrastruttura di rete che copre tutta una nazione
    - Tipicamente il gestore “incumbent”
  - Gli ISP Tier 1 possono essere
    - Nazionali quando servono una sola Internet regione
    - Globali quando hanno punti di accesso in paesi e continenti diversi

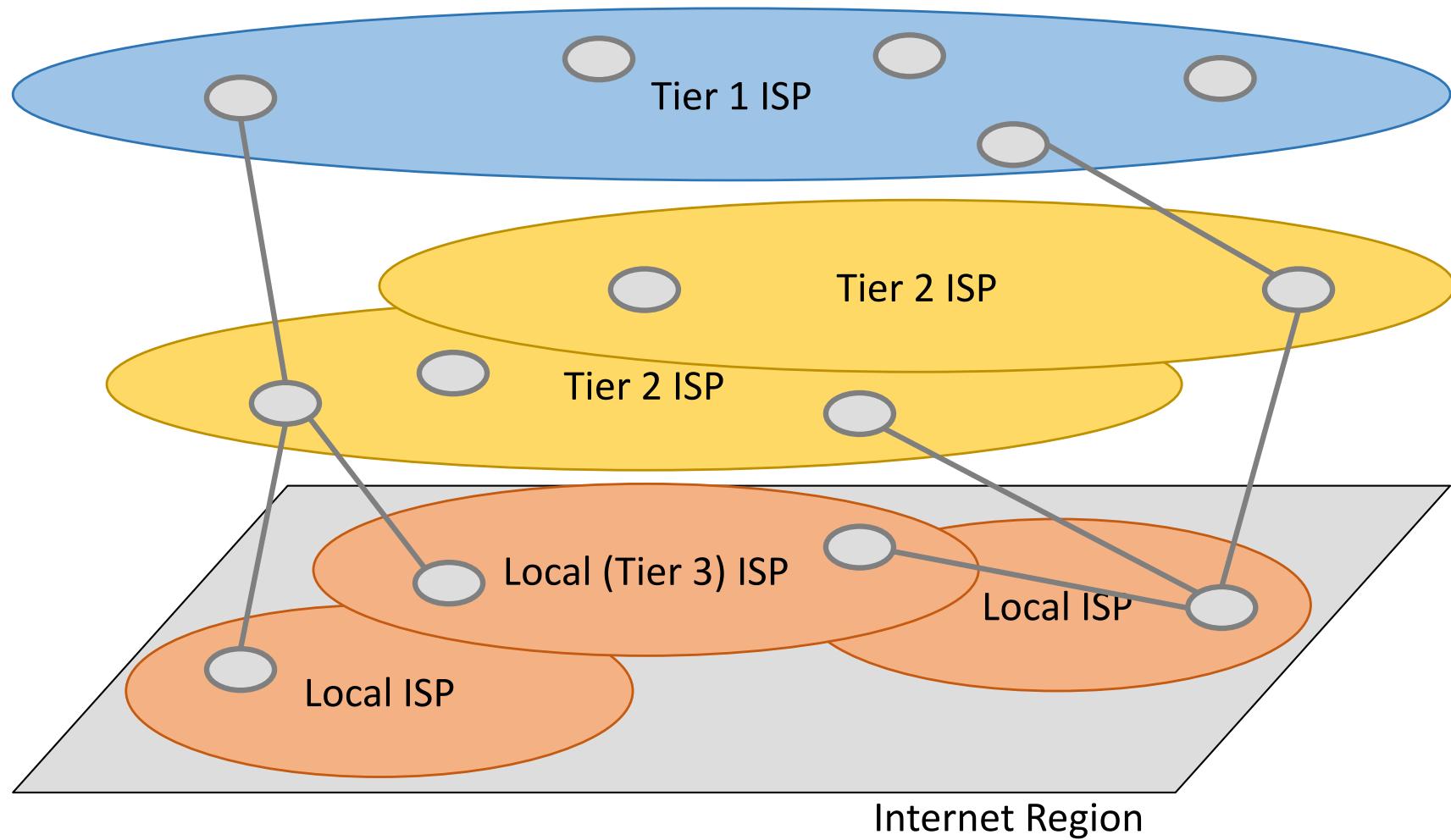


# Classificazione degli ISP

- Tier 2
  - Un ISP che raggiunge l'Internet globale acquistando servizi di interconnessione da un Tier 1 ISP
  - Un ISP Tier 2 può avere interconnessioni anche con più di un ISP Tier 1 nella stessa o in diverse Internet Region
- Tier 3
  - Un ISP che serve un'area abbastanza delimitata
    - ISP locali o regionali
  - Per raggiungere l'Internet globale acquista servizi di interconnessione da un ISP Tier 2
  - Può avere interconnessioni dirette (peering) con altri ISP Tier 3 che servono la stessa zona o zone limitrofe



# In sintesi





# In Italia

- Il principale ISP Tier 1 è Telecom Italia Sparkle
  - Da RadB

aut-num: AS6762

as-name: SEABONE-NET

descr: TELECOM ITALIA SPARKLE S.p.A.

remarks: International Internet Backbone



# Il Peering

Il peering è un'interconnessione diretta tra due AS, stabilita per consentire il passaggio diretto del traffico tra le reti degli AS coinvolti. Questo accordo è comune tra ISP dello stesso livello (come Tier 2 con Tier 2), ma anche tra ISP e operatori di contenuti (ad esempio, Amazon, Netflix, e Aruba). Il peering diretto permette un traffico più rapido ed efficiente, poiché evita di passare per un ISP Tier 1 (che di solito implica costi di transito). IL PEERING NON È LIMITATO SOLO AGLI ISP DELLO STESSO LIVELLO. Gli accordi di peering possono coinvolgere ISP di livelli differenti (Tier 1, Tier 2 e Tier 3).

- Relazione di peering
  - Interconnessione fra due AS stabilita al fine di scambiarsi traffico
    - Con operatore di contenuti (Amazon, Aruba, Netflix)
- La relazione di peering non ha carattere economico
  - Gli AS non devono pagarsi reciprocamente per lo scambio di traffico
  - I loro introiti rimangono limitati alla tariffazione dei rispettivi utenti
- Tipicamente il peering avviene fra ISP del medesimo livello

Poiché il peering è privo di costi tra i due AS, né l'uno né l'altro deve pagare per inviare o ricevere traffico attraverso questa connessione diretta. Quindi i loro introiti derivano unicamente dalle tariffe applicate ai clienti finali.



# Peering policy

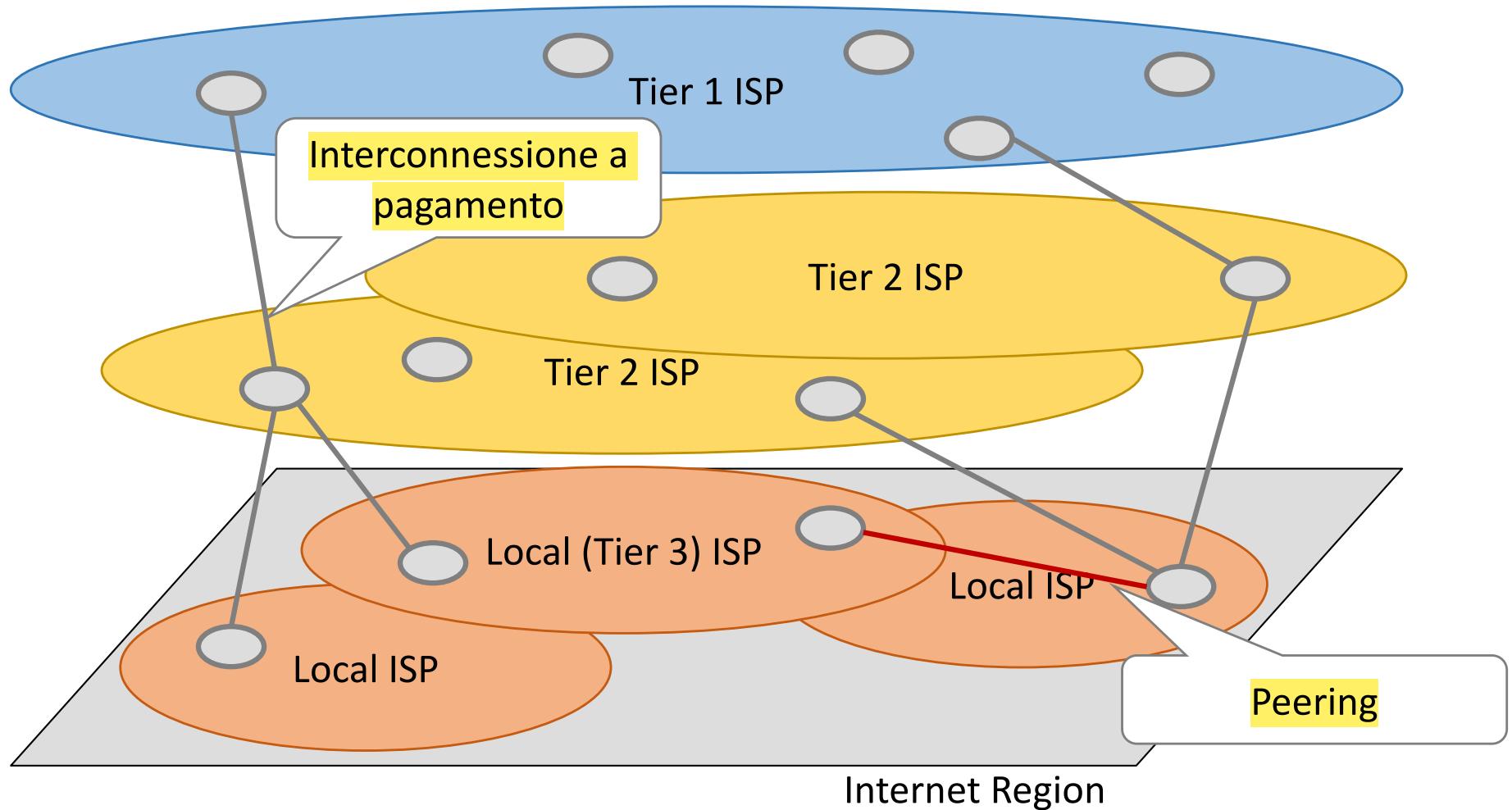
- Ristretta
  - Devi chiedere di fare peering e la richiesta va approvata
- Aperta
  - Approvato di default

# In sintesi



Interconnessione a pagamento: ISP di Tier inferiore connettono ISP di Tier superiore.

Peering gratuito: Avviene tipicamente tra ISP dello stesso livello o con grandi provider di contenuti per ottimizzare i costi e migliorare l'efficienza del traffico.





# ISP locali e POP

Il POP è il punto di accesso fisico in cui l'ISP locale ospita router, switch e altre apparecchiature di rete per fornire connessione ai clienti in quella zona. Dal POP, l'ISP gestisce e instrada il traffico di tutti gli utenti connessi. È un nodo di rete che collega l'infrastruttura locale con l'Internet globale, tramite interconnessioni con altri ISP di livello superiore (ad esempio, ISP Tier 2 o Tier 1).

cioé concentrati  
in una  
determinata area  
geografica

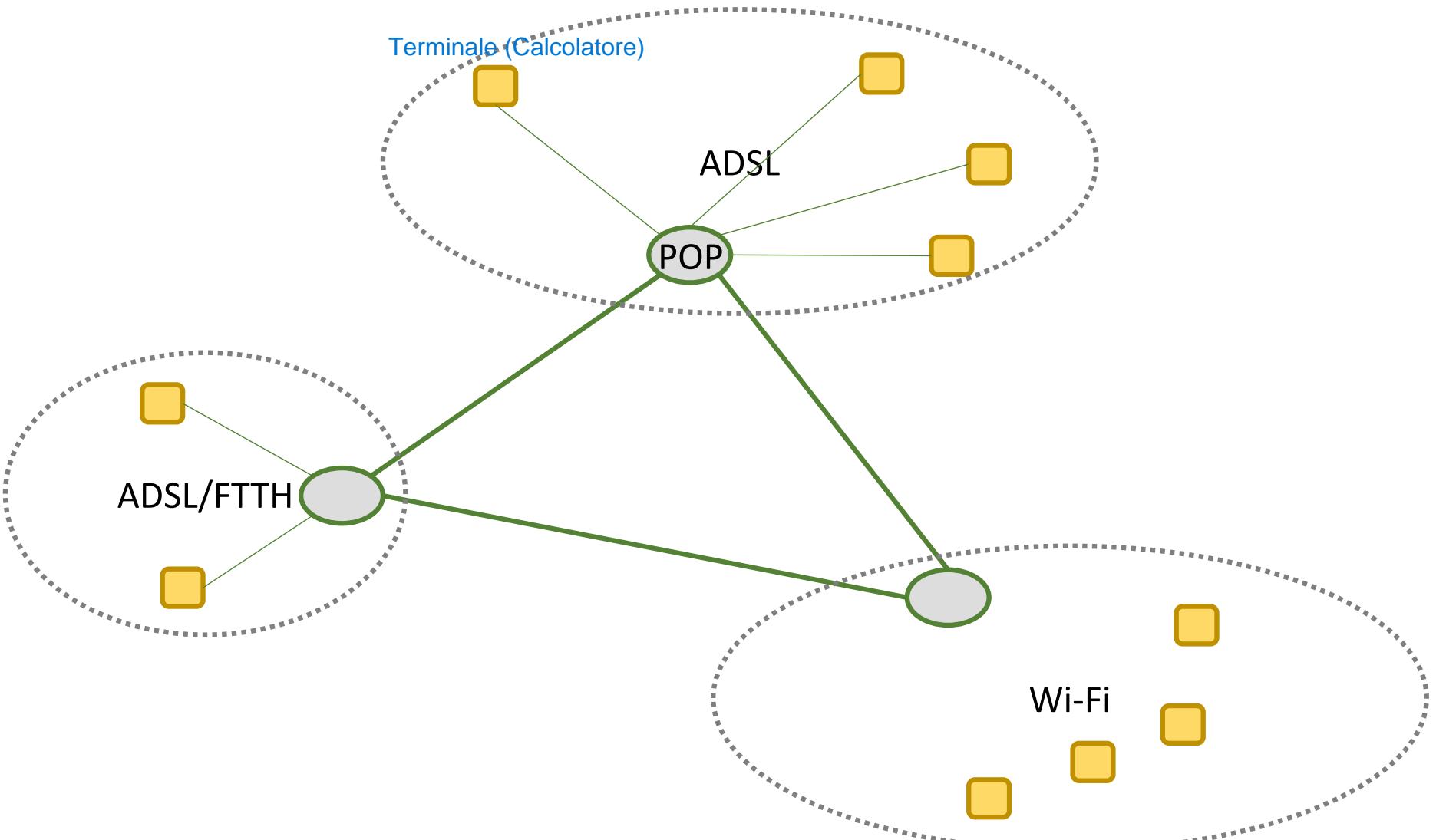
- Un ISP locale fornisce il servizio a gruppi di utenti co-localizzati (singola città, area industriale ecc.)

- Per raggiungere questi utenti, l'ISP locale realizza un'infrastruttura con router e switch in un punto della zona detto Point of Presence o POP

- Come collega gli utenti a quella infrastruttura?
  - Riutilizzo del vecchio collegamento telefonico in rame (ADSL e simili)
  - Fibra ottica (FTTH)
  - Collegamento radio (Wi-Fi e simili)
  - Soluzioni miste (rame+fibra ad esempio nel FTTC)



# Esempio





# Indirizzamento

- Un ISP dispone di un sottoinsieme di numeri IP da utilizzare per i suoi clienti
  - Se sono consecutivi possono avere lo stesso prefisso quindi unico Network ID
  - Se non sono consecutivi deve gestire più prefissi quindi più Network ID
- In funzione della dimensione (numero di utenti e distanze geografiche) la rete dell'ISP può essere composta da una o più LAN



# Interconnessione

- Come scambiano traffico ISP che coprono la medesima zona geografica

In questo caso, ogni POP di ciascun ISP è connesso direttamente agli altri POP della stessa area geografica.

- Interconnettere fra loro tutti i POP?

- Numerosi collegamenti

Ogni POP deve mantenere connessioni con altri POP, e questo implica un numero molto elevato di collegamenti fisici e virtuali da gestire.

- Complessità di gestione del routing

- Rotte specifiche per ogni POP in funzione dei numeri a loro connessi

- Percorsi di lunghezza minima

Ogni POP ha una connessione diretta con altri POP, consentendo spesso percorsi di lunghezza minima per il traffico locale.

- Interconnettere uno o pochi POP?

- Minor numero di collegamenti

- Routing semplificato

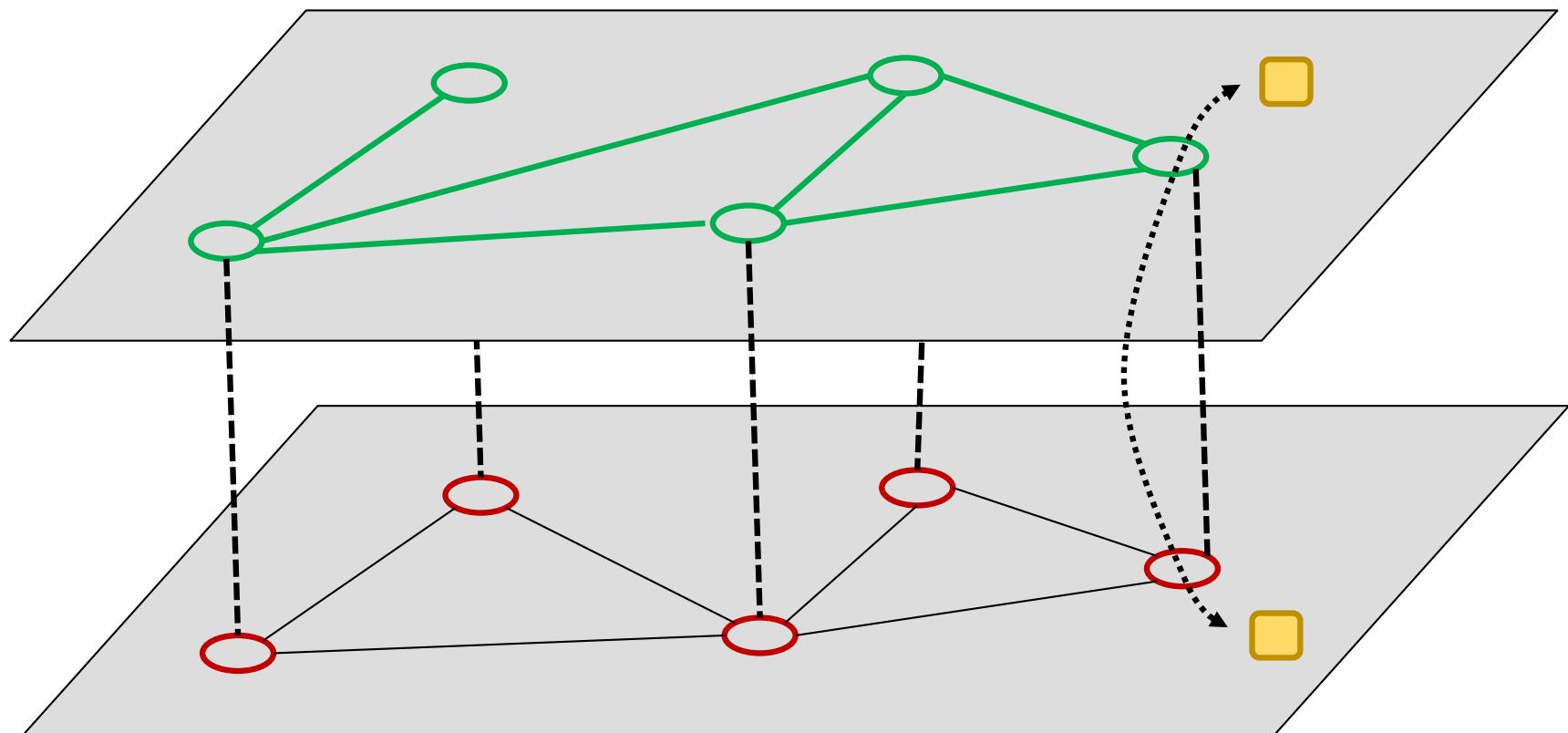
- Percorsi potenzialmente più lunghi

In questa strategia, solo uno o pochi POP fungono da nodi di interconnessione tra i vari ISP. Gli altri POP si collegano ai loro rispettivi nodi di interconnessione per accedere ad altre reti.

Con numerosi collegamenti, il sistema di routing diventa complesso, poiché ogni POP deve conoscere le rotte di tutti gli altri POP e aggiornare costantemente le informazioni sui percorsi più efficienti.

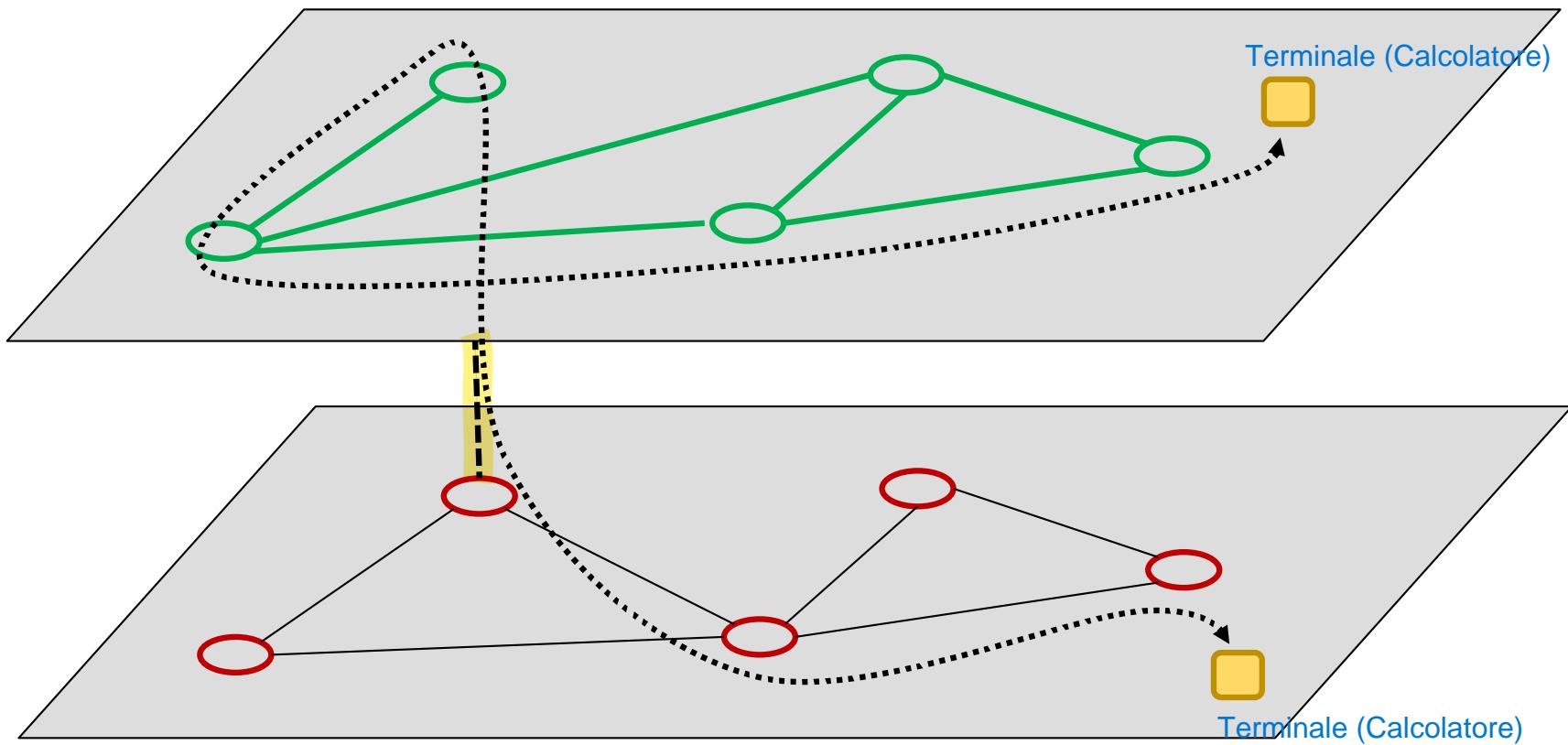


# Non utilizzata





# Peering diretto tramite due POP





# Da Tier 3 a Tier 1

- Teoricamente ogni ISP dovrebbe fare peering con ogni altro ISP con cui vuole scambiare traffico
  - Ogni AS dovrebbe essere connesso con ogni altro AS
    - Gran numero di collegamenti dedicati fra POP con un conseguente aumento della complessità e dei costi di gestione.
- Alcuni ISP svolgono la funzione di AS di transito per interconnettere con una topologia “a stella” gli ISP
  - Gli ISP specializzati nel fornire servizi di transito sono anche detti Network Service provider (NSP)
- Talvolta gli NSP coincidono con ISP Tier 1

In questo modello, gli ISP Tier 3 e Tier 2 possono connettersi a NSP (o ISP Tier 1) per raggiungere destinazioni globali senza dover mantenere una propria rete globale o numerose connessioni dirette.



# Internet Exchange

Il peering può avvenire in punti di interscambio Internet (IXP) pubblici o privati, dove ISP di vari livelli possono scambiare traffico. Gli IXP pubblici permettono a ISP di tutti i livelli di interconnettersi.

- Per favorire l'interconnessione fra ISP e NSP (ssia fra i loro AS) esistono gli IXP
- Internet Exchange Point (IX o IXP)
  - Infrastrutture attraverso le quali gli ISP possono stabilire relazioni di peering ovvero accordi per lo scambio diretto di traffico dati tra le rispettive reti.
  - L'IXP è costruito per permettere l'interconnessione diretta degli AS senza utilizzare reti di terze parti
  - L'IXP fornisce soluzioni di connettività con specifiche garanzie di qualità (disponibilità elevata, sicurezza fisica, banda garantita ecc.)
- Un elenco degli IXP nel mondo si può trovare alla pagina [www.internetexchangemap.com](http://www.internetexchangemap.com)

Gli AS appartenenti sia agli ISP sia agli NSP possono connettersi direttamente tra loro.





# IXP in Italia

il piú importante

- MIX (Milan Internet eXchange)
  - Milano, Palermo, Catania
- NaMeX (Nautilus Mediterranean eXchange point)
  - Roma
- TOP-IX (Torino Piemonte Internet Exchange)
  - Torino
- Tuscany Internet eXchange
  - Firenze
- PCIX
  - Piacenza



ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

# Interior Gateway Protocol

IGP

# Routing Information Protocol (RIP)

RIP si basa su un algoritmo di tipo "distance vector", in cui ogni router conosce solo la distanza (in termini di "hop", ossia salti da un router all'altro) e la direzione (vettore) per raggiungere una rete di destinazione.

- Protocollo **distance vector**, di implementazione vecchia (RFC 1058, Giugno 1988), discende dal protocollo di routing realizzato per la rete XNS di Xerox
- Ne esiste una **versione 2** più recente (RFC 2453)
- Molto diffuso in passato perché il codice di implementazione è liberamente disponibile
- Utilizzato praticamente solo su reti TCP/IP
- Utilizza due tipi di messaggi:
  - **REQUEST** serve per chiedere esplicitamente informazioni ai nodi vicini (ad es. all'avvio del nodo) Viene utilizzato dai router per richiedere informazioni di routing dai nodi vicini
  - **RESPONSE** serve in generale per inviare informazioni di routing (cioè i distance vector) Viene utilizzato per fornire le informazioni di routing (distance vector) ai nodi richiedenti o per aggiornare i vicini sui cambiamenti delle rotte.
- I messaggi RIP sono trasportati da UDP ed usano la porta 520 sia in trasmissione che in ricezione

 RIP è progettato per reti TCP/IP e utilizza l'User Datagram Protocol (UDP) per il trasporto dei suoi messaggi.

# Response

Se un router rileva un cambiamento in una delle rotte, invia immediatamente un RESPONSE per notificare i cambiamenti ai suoi vicini.

- Un **RESPONSE** con nuove informazioni di routing viene inviato:
  - periodicamente
  - come risposta ad una richiesta esplicita
  - quando una informazione di routing cambia (triggered update)
- Le informazioni periodiche sono inviate ogni 30 secondi, con uno scarto da 1 a 5 secondi, per evitare “tempeste” di aggiornamenti Per evitare che questi aggiornamenti simultanei causino un sovraccarico di traffico (fenomeno noto come "tempeste di aggiornamenti"), ogni router applica un intervallo casuale di 1-5 secondi rispetto ai 30 secondi standard, in modo che non tutti i router trasmettano esattamente nello stesso istante.
- Response contiene il distance vector del router che lo invia
  - Destinazione
  - Distanza (hop count) per raggiungere ciascuna destinazione.

# RIP: formato dei pacchetti

(rappresenta l'intero pacchetto RIP)

- La struttura del pacchetto è basata su parole di 32 bit
- Il pacchetto può avere lunghezza variabile fino a 512 byte (max 25 entry)  
Questo limite di 512 byte deriva dalla massima dimensione di un datagramma UDP standard.

ripetuto	command	version	must be zero
	address family identifier		must be zero
	address		
	must be zero		
	must be zero		
	metric		
	address family identifier		must be zero
	address		
	must be zero		
	must be zero		
	metric		
	32 bit		

# RIP: significato dei campi

- I bit del pacchetto sono molto ridondanti rispetto alla quantità di informazioni da inviare (molti campi fissi con i bit tutti a zero) (molti campi del pacchetto RIP sono ridondanti e impostati a zero)
  - inizialmente pensati per adattarsi ad altri protocolli

imposto 1 per  
REQUEST  
imposto 2 per  
RESPONSE



**command**: distingue tra REQUEST (1) e RESPONSE (2)

distingue il tipo  
di messaggio  
RIP

- version**: versione del RIP



**address family identifier**: indica il tipo di indirizzo di rete utilizzato, vale 2 per IP

Questo campo permette teoricamente a RIP di supportare anche altri protocolli di rete, sebbene nella pratica sia utilizzato quasi esclusivamente con IP.

- address**: identifica la destinazione per la quale viene data la distanza Campo di 32 bit che contiene l'indirizzo della rete di destinazione per cui vengono fornite le informazioni di routing.
- metrica**: è la distanza dalla destinazione indicata

Campo di 32 bit che rappresenta la "distanza" in termini di hop verso la destinazione specificata.

# RIP: la tabella di routing

- Ogni riga nella tabella contiene:

- indirizzo di **destinazione**: è un indirizzo IP a 32 bit
- **distanza** dalla destinazione (metrica)

- in termini di hop-count (ogni link ha peso = 1) In RIP, ogni hop è considerato con un peso di 1.

- la distanza massima ( $\infty$ ) per RIP è pari a **16**, al fine di limitare il conteggio all'infinito → adatto per reti relativamente piccole

- **next-hop** sul percorso verso la destinazione
  - router vicino a cui inviare i datagrammi per la destinazione
- due contatori

- **Timeout**: se una route non viene aggiornata dopo TO secondi, la sua distanza è posta all'infinito (si ipotizza una perdita di connettività)

- **Garbage-collection timer**: se dopo ulteriori GC secondi la route viene eliminata del tutto dalla tabella

- I valori di default sono TO = 180 s e GC = 120 s

## Timeout:

Se una rotta non viene aggiornata (cioè non più presente nei DV inviati dai vicini) entro un certo intervallo di tempo (default: 180 secondi), la metrica della rotta viene impostata a "infinito" (16), segnalandola come non più valida. Questo timeout indica una possibile perdita di connessione o una modifica nella topologia della rete.

## Garbage-Collection Timer:

Dopo che una rotta è stata marcata come non valida (distanza infinita), viene avviato un secondo timer, il Garbage-Collection Timer. Se trascorrono ulteriori 120 secondi (default) senza che la rotta venga aggiornata, questa viene eliminata completamente dalla tabella. Questa procedura evita che rotte obsolete rimangano inutilmente nella tabella, liberando spazio per nuove informazioni di routing.

RIP ha una distanza massima di 16, che viene interpretata come "infinito", indicando che la destinazione non è raggiungibile. Questo limite rende RIP adatto solo per reti relativamente piccole, perché non può supportare percorsi con più di 15 router intermedi. Questa limitazione serve a prevenire i loop di routing, poiché un valore di 16 indica automaticamente una rotta non valida.

Ogni riga contiene un indirizzo di destinazione a 32 bit, che rappresenta la rete o il nodo finale che si vuole raggiungere.

Il next-hop è il router vicino a cui inviare i pacchetti per raggiungere la destinazione. Il next-hop è importante per determinare il percorso effettivo dei dati

# RIP: aggiornamento della tabella di routing

- A riceve un RESPONSE da B
  - Si controlla la correttezza dei dati (indirizzi IP e metriche validi)
  - Si considerano solo le voci  $i$  con distanze  $d_i < \infty$  (ovvero, voci che rappresentano percorsi raggiungibili)
  - Si calcola  $d_i = d_i + 1$
- Esiste già una entry per la destinazione  $i$  ?
  - NO
    - Si crea una nuova entry
      - la distanza è  $d_i$
      - il next-hop è B (mittente del RESPONSE)
      - si fa partire il timeout
  - SI
    - $d_i$  è minore di quella presente in tabella
      - la entry viene aggiornata con next hop = B e distanza =  $d_i$
      - si fa ripartire il timeout
    - Next hop = B
      - Si aggiorna a distanza
      - Si fa ripartire il timeout

A calcola la nuova distanza verso la destinazione  $i$  attraverso il router B come  $D_i = D_i + 1$ , aggiungendo un hop alla metrica ricevuta da B (poiché la distanza include un ulteriore hop per passare da A a B).

Questo assicura che la rotta resti valida e aggiornata anche se il costo rimane invariato.

# RIP: problematiche

Split Horizon è una tecnica utilizzata per prevenire i loop di routing, che consiste nel non inviare informazioni su una rotta attraverso l'interfaccia da cui la rotta è stata appresa. In RIP, questo significa che il router non reinvia su una determinata interfaccia le informazioni di routing ricevute da quella stessa interfaccia. Questo può causare comportamenti differenti tra le interfacce, creando situazioni in cui i RESPONSE delle diverse interfacce contengono informazioni diverse.

- Fa uso di split horizon
  - RESPONSE di interfacce diverse possono essere diverse
- Fa uso di triggered update
  - non è necessario indicare nella RESPONSE tutte le entry della tabella ma solamente quelle appena modificate
- Non supporta il CIDR
- È un protocollo insicuro
  - Chiunque trasmetta datagrammi dalla porta UDP 520 viene considerato come un router autorizzato
  - Esempio di malfunzionamento indotto:
    - un router non autorizzato trasmette messaggi contenenti indicazione di una distanza 0 tra se stesso e tutti gli altri della rete
    - dopo qualche tempo tutti i percorsi ottimi convergono su questo router

I triggered update consentono a RIP di inviare un aggiornamento immediato solo quando c'è un cambiamento di rotta, senza aspettare l'intervallo periodico di aggiornamento. Questo significa che non è necessario inviare l'intera tabella di routing, ma solo le voci che sono cambiate.

Ogni interfaccia di un router è collegata a una rete o a uno o più router vicini. Poiché lo split horizon impedisce a un router di annunciare una rotta su un'interfaccia specifica (quella da cui l'ha appresa), le informazioni inviate su ciascuna interfaccia possono essere diverse.

RIP non supporta CIDR (Classless Inter-Domain Routing), ovvero il sistema di indirizzamento che permette di specificare maschere di sottorete di lunghezza variabile. Questo limita RIP, poiché può lavorare solo con classi di indirizzi standard (A, B e C), rendendolo inefficiente in reti moderne.

RIP è considerato insicuro perché non ha meccanismi di autenticazione robusti (soprattutto nelle versioni precedenti a RIP v2). Chiunque trasmetta datagrammi sulla porta UDP 520 può essere interpretato come un router autorizzato, il che espone RIP a potenziali attacchi di rete.



# La mancanza di CIDR

- Si supponga di voler utilizzare la rete 10.0.0.0 suddivisa in sottoreti /24
- Si realizzi la seguente architettura di rete



- RouterB riceve distance vector contenenti la rete 10.2.2.0 da RouterC e la rete 10.1.1.0 da RouterA
- In assenza di CIDR 10.1.1.0 e 10.2.2.0 sono indirizzi appartenti alla stessa rete di classe A 10.0.0.0/8
  - Per RouterB 10.0.0.0/8 deve essere un'unica destinazione
  - RouterB è confuso perché vede la stessa rete in due diverse direzioni

Router B si trova "confuso" perché riceve informazioni contrastanti sulla rete 10.0.0.0/8 da due direzioni diverse. Non potendo distinguere le sottoreti 10.1.1.0/24 e 10.2.2.0/24 come entità separate, Router B non è in grado di instradare correttamente il traffico verso queste sottoreti e può causare problemi di instradamento, instradando i pacchetti verso il next-hop sbagliato.

# RIP versione 2

RIP v2 supporta CIDR (Classless Inter-Domain Routing), permettendo l'uso di maschere di sottorete di lunghezza variabile. Questo consente una maggiore flessibilità nell'organizzazione degli indirizzi IP, superando la limitazione della classe A, B, C e rendendo possibile l'uso di sottoreti con dimensioni diverse all'interno di una stessa rete principale. L'introduzione del campo subnet mask permette a RIP v2 di specificare esattamente la maschera associata a ogni rotta, consentendo una gestione più precisa e dettagliata delle rotte.

- I miglioramenti introdotti riguardano soprattutto:

- subnetting e CIDR
- autenticazione

RIP v2 introduce un meccanismo di autenticazione per migliorare la sicurezza del protocollo. Il campo authentication type indica il tipo di autenticazione utilizzata. Nella maggior parte delle implementazioni, viene usata un'autenticazione semplice basata su password. I campi di authentication data contengono effettivamente le informazioni di autenticazione (come una password), che i router possono utilizzare per verificare la legittimità delle informazioni di routing ricevute. Questo aiuta a prevenire attacchi in cui un router non autorizzato trasmette aggiornamenti di routing falsi, un problema comune in RIP v1, che non prevedeva autenticazione.

command	version	routing domain
11111111	11111111	authentication type
authentication data		
address family identifier		route tag
address		
subnet mask		
next hop		
metric		

ripetuto

Zona di Autenticazione

# RIP versione 2

RIP v2 mantiene la compatibilità con RIP v1. Questo consente un'integrazione graduale di RIP v2 in reti già esistenti con RIP v1, anche se i miglioramenti di RIP v2 non sono completamente utilizzabili da router con RIP v1: Un router configurato con RIP versione 1 (che non conosce i campi aggiuntivi di RIP v2) ignorerà tutte le informazioni di routing presenti nel messaggio RIP v2 che contengono valori in campi che RIP v1 non riconosce (cioè nei campi di RIP v1 con "must be zero").

- Compatibilità verso il basso
  - RIP-1 ignora le entry con i campi riservati diversi da zero
- Possibilità di indicare sottoreti o indirizzamento CIDR
  - tramite il campo **subnet mask**
- Possibilità di **autenticare** chi invia i messaggi
- Possibilità di indicare il proprio AS e di scambiare informazioni con protocolli EGP
  - tramite i campi **route tag** e **routing domain**
- Possibilità di specificare un **next hop** più appropriato

RIP v2 permette di specificare un next hop per le rotte in modo più preciso, ottimizzando il percorso per raggiungere la destinazione.

- Comunque non adatto ad AS grandi
- Comunque ha problemi di convergenza
  - è pur sempre un distance vector

RIP v2 introduce un meccanismo di autenticazione, il che migliora la sicurezza del protocollo. L'autenticazione consente di verificare l'origine dei messaggi, prevenendo potenziali attacchi in cui un dispositivo non autorizzato invia aggiornamenti di routing falsi.

Il Route Tag è un campo che può essere utilizzato per marcare le rotte con un'etichetta specifica. Questo tag consente di distinguere tra rotte interne (apprese all'interno dello stesso AS) e rotte esterne (apprese da un altro AS tramite un protocollo esterno EGP).

Il campo Routing Domain indica il dominio di routing a cui appartiene il pacchetto RIP. Questo campo è utile quando si hanno reti grandi e segmentate in più domini di routing all'interno dello stesso AS. Il routing domain permette di mantenere le rotte separate anche all'interno dello stesso AS. Ad esempio, se un'organizzazione ha più divisioni con reti interne separate, il routing domain permette di indicare a quale di queste divisioni appartiene una rota.

# Open Shortest Path First (OSPF)

- Divenuto standard nella versione 2 (RFC 2328)
- Oggi è il più diffuso IGP
- Protocollo di tipo **link state** che utilizza le informazioni sullo stato delle connessioni tra router per costruire una mappa completa della rete.
  - invio di **Link State Advertisement** (LSA) a tutti gli altri router
- Incapsulato direttamente in IP OSPF è incapsulato direttamente in IP, senza l'utilizzo di un protocollo di trasporto come TCP o UDP
  - il valore del campo protocol dell'intestazione IP (89 per OSPF) serve a distinguere questi pacchetti da altri
- OSPF è stato progettato specificamente per:
  - semplificare il routing in reti grandi tramite la suddivisione in aree
  - gestire intrinsecamente reti punto-punto e punto-multipunto
  - separare logicamente gli host dai router

OSPF è progettato per gestire solamente il routing tra i router e non si occupa direttamente degli host

Gestire intrinsecamente reti punto-punto e punto-multipunto significa che OSPF è progettato per riconoscere e adattarsi automaticamente a ciascuna di queste configurazioni tra router, ottimizzando il comportamento del protocollo di routing per ridurre il traffico di aggiornamento, semplificare la configurazione e migliorare l'efficienza complessiva della rete. Questa capacità di adattamento automatico rende OSPF versatile e adatto per reti complesse e di grandi dimensioni, senza necessità di configurazioni manuali per ogni specifica topologia.

# OSPF: aree di routing

L'Area 0 è conosciuta come backbone della rete OSPF ed è fondamentale per connettere tutte le altre aree. Ogni altra area deve essere connessa all'Area 0.

- Un AS può essere suddiviso in porzioni dette **Routing Area** (RA) interconnesse da un **backbone** (Area 0)

Ogni area gestisce solo le informazioni di routing necessarie per le destinazioni all'interno di quell'area. Anche se ogni area gestisce solo le informazioni di routing per le destinazioni interne a quell'area, esiste un meccanismo per permettere la comunicazione tra aree diverse: attraverso l'Area 0 (backbone) e gli Area Border Router (ABR)

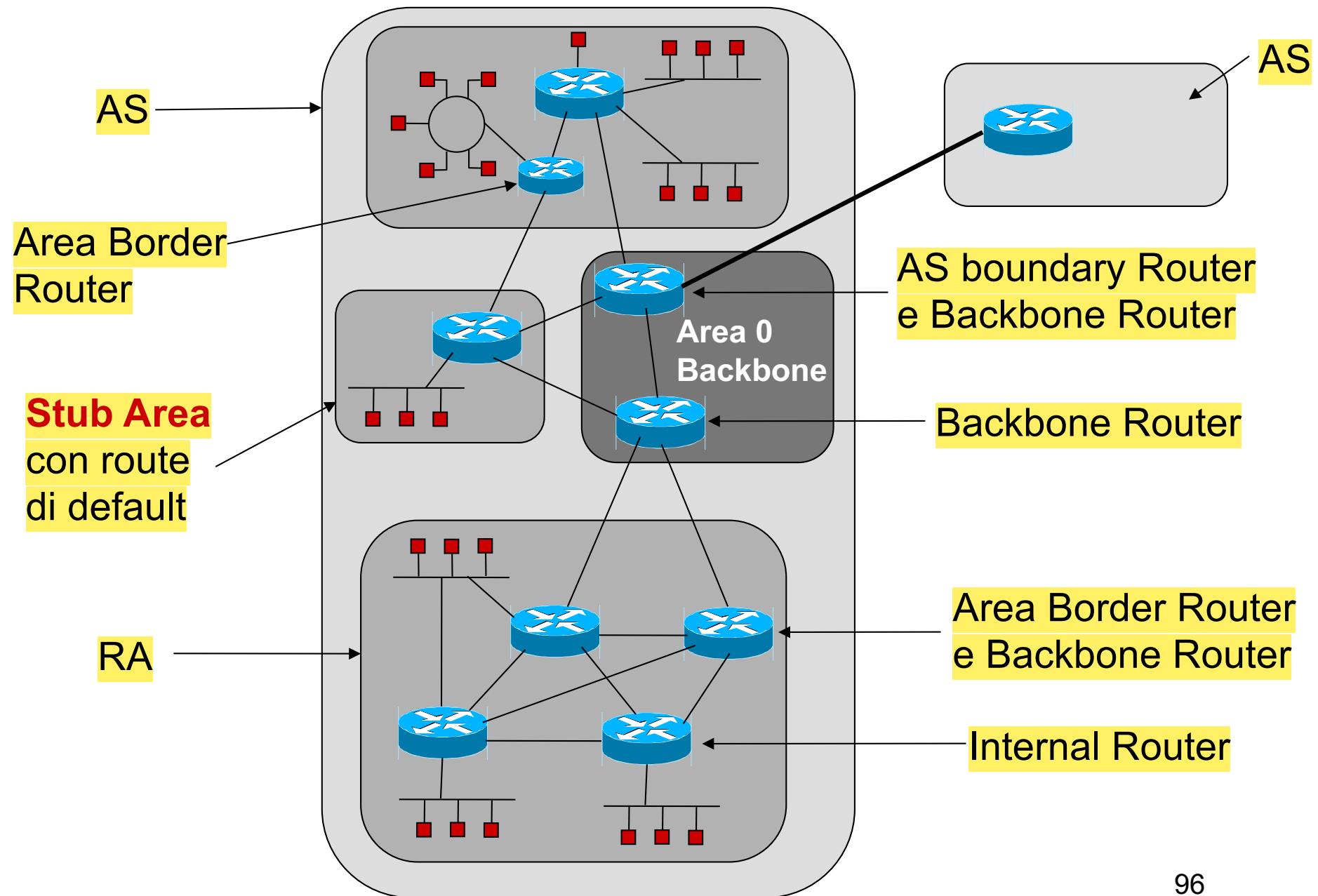
- ciascuna area risulta separata dalle altre per quanto riguarda lo scambio delle informazioni di routing e si comporta come un'entità indipendente (3° livello gerarchico di routing)
- per interconnettere le aree vi devono essere router connessi a più aree e/o al backbone (almeno un router per area)

## Classificazione dei router secondo OSPF:

- **Internal Router**: router interni a ciascuna area
- **Area Border Router**: router che scambiano informazioni con altre aree
- **Backbone Router**: router che si interfacciano con il backbone
- **AS Boundary Router**: router che scambiano informazioni con altri AS usando un protocollo EGP

i Backbone Router sono i router che fanno parte della Area 0

# OSPF: aree di routing e tipologie di router





# Tipi di route

- **Route intra-area** (sono rotte che collegano destinazioni all'interno della stessa area)
  - aggiornamento delle informazioni di routing pertinenti all'area

Questi aggiornamenti di routing sono limitati all'interno dell'area stessa e non vengono condivisi con altre aree.
- **Route inter-area** (sono rotte che collegano destinazioni situate in aree diverse)
  - Aggiornamento delle informazioni di routing pertinenti ad aree diverse da quella considerata
- **Route esterni** (sono rotte che puntano a destinazioni al di fuori del dominio OSPF)
  - Aggiornamenti delle informazioni di route provenienti da altri protocolli al di fuori del dominio OSPF
  - Inoltrati nel dominio OSPF dal ASBR

Gli aggiornamenti di routing per le rotte inter-area vengono scambiati tra le aree

L'ASBR prende le informazioni di routing apprese da un protocollo esterno e le inietta nel dominio OSPF, permettendo agli altri router di conoscere rotte esterne.



# Tipi di aree

- **Area normale** → Un'area normale accetta tutti i tipi di route, inclusi route intra-area, inter-area e route esterne (apprese da altri AS tramite ASBR). È l'area standard di OSPF, dove tutte le rotte, comprese quelle esterne all'AS, vengono propagate all'interno dell'area.
    - Accetta tutti i tipi di route
  - **Stub area**
    - Accetta route intra e inter area
    - Tutti i router della stub area usano un “default route” verso destinazioni al di fuori dell'AS
      - Comunicato dall'Area Border Router (ABR)
    - I requisiti di memoria dei router sono ridotti
- Totally stub area**
- Vengono propagati solamente route intra-area ed il route di default
    - Il default route viene propagato dal ABR
    - Tutti i router dell'area usano il default route per destinazioni esterne all'area
- Not so stubby area**
- Stub area che importa alcuni route esterni
  - Uno dei router dell'area è connesso a un AS diverso e diventa un ASBR

Una NSSA è una variante della Stub Area che permette l'importazione di alcune route esterne pur bloccando la maggior parte delle altre rotte esterne. In una NSSA, uno dei router può essere connesso a un AS esterno e diventare un ASBR. Questo ASBR può importare alcune rotte esterne e renderle disponibili all'interno dell'NSA. Le route esterne importate non sono propagate in tutta l'area OSPF come rotte esterne standard, ma solo all'interno dell'NSA.

Una Totally Stub Area è una variante più restrittiva della Stub Area, che accetta solo route intra-area e una default route fornita dall'ABR. Oltre a bloccare le route esterne, una Totally Stub Area blocca anche le route inter-area, propagando solo le rotte interne all'area e una default route per tutte le destinazioni esterne. Adatta per aree che hanno solo bisogno di raggiungere l'Area 0 senza dettagli sui percorsi inter-area, tipicamente piccole reti periferiche o aree con risorse limitate.

# OSPF: ulteriori caratteristiche

Questo approccio riduce il sovraccarico su un singolo percorso e utilizza in modo efficiente tutte le risorse di rete disponibili. Inoltre, bilanciando il carico, si evita che un percorso sia congestionato mentre altri rimangono inutilizzati.

- **Bilanciamento del carico:** se un router ha più percorsi di uguale lunghezza verso una certa destinazione, il carico viene ripartito equamente su di essi

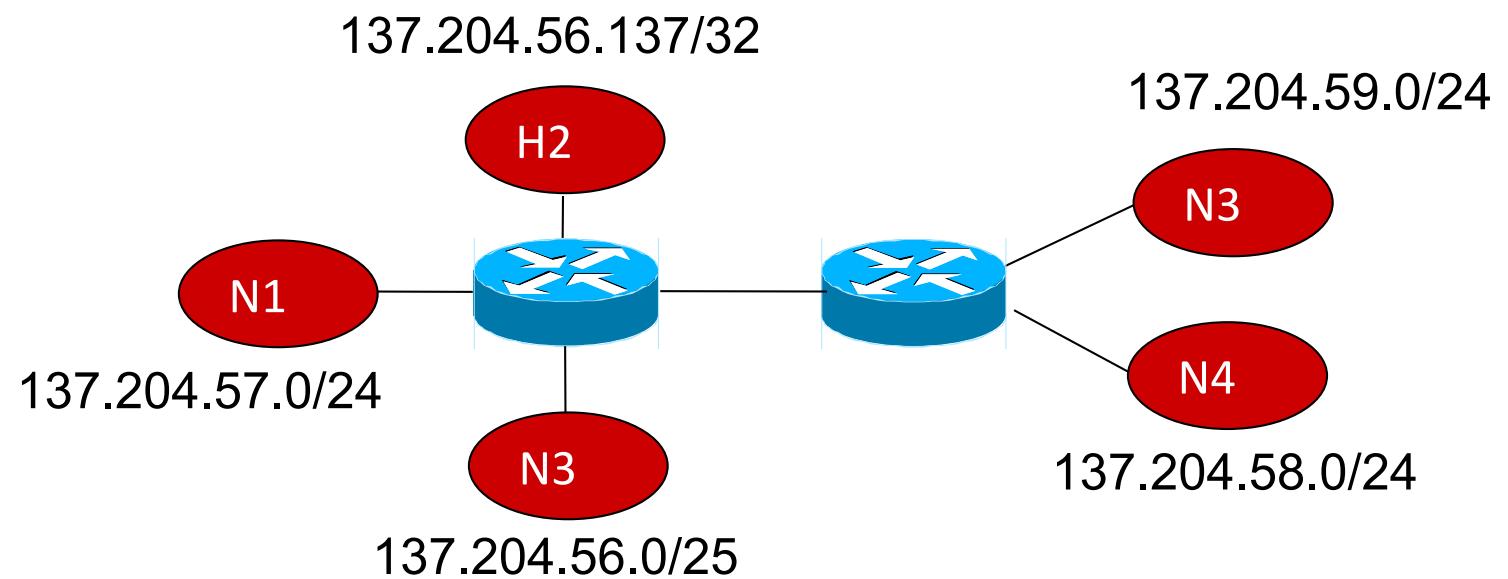
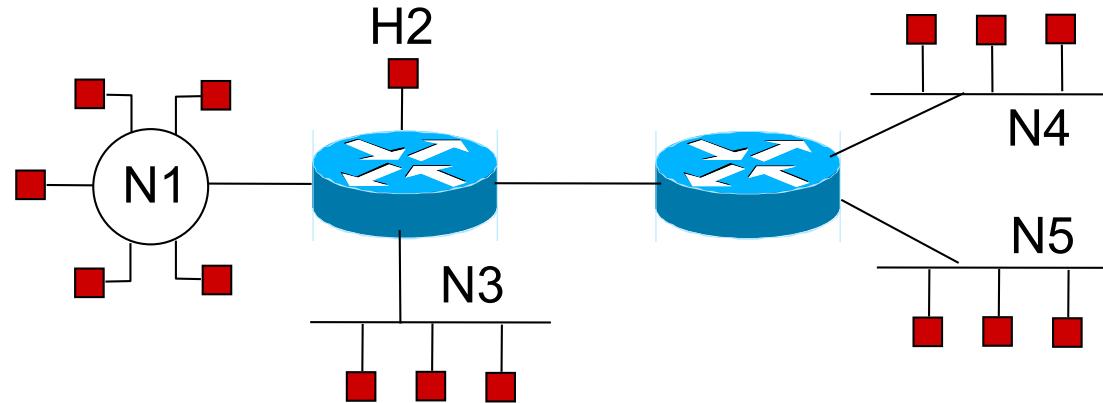
→ L'autenticazione aumenta la sicurezza, impedendo a router non autorizzati di unirsi al dominio OSPF e manipolare le informazioni di routing.

- **Autenticazione:** per garantire maggiore sicurezza nello scambio delle informazioni di routing è prevista autenticazione con password ed uso di crittografia

→ Permette di instradare i pacchetti in base ai requisiti di qualità del servizio, migliorando l'efficienza della rete per traffico sensibile alla latenza o con esigenze di larghezza di banda.

- **Routing dipendente dal grado di servizio:** i router scelgono il percorso sul quale instradare un pacchetto sulla base dell'indirizzo e del campo Type of Service dell'intestazione IP, tenendo conto che percorsi diversi possono offrire diversi gradi di servizio

# OSPF: rappresentazione di host e router



# OSPF: tipologie di rete

In una rete BMA, più router condividono lo stesso segmento di rete e possono comunicare tra loro direttamente tramite broadcast. In questo tipo di rete, per evitare un eccessivo traffico di aggiornamento tra router, OSPF utilizza un meccanismo di elezione per scegliere un Designated Router (DR) e un Backup Designated Router (BDR). Il DR gestisce la distribuzione degli aggiornamenti di routing a tutti i router sulla rete.

- OSPF è progettato per operare correttamente con reti:

- **Point-to-Point** Questa tipologia rappresenta una connessione diretta tra due router, dove non ci sono altri dispositivi o router sulla linea.
- **Broadcast Multi-Access** (BMA: LAN 802)
- **Non-Broadcast Multi-Access** (NBMA: X.25, ATM, Frame Relay)

Le reti NBMA supportano più connessioni tra router, ma non permettono la trasmissione in broadcast. OSPF deve gestire le connessioni in modo diverso, poiché non è possibile inviare pacchetti broadcast per comunicare con tutti i router.

## In una **rete ad accesso multiplo** tutti gli N router

connessi alla rete sono di fatto connessi con tutti gli altri

- il numero di archi bidirezionali da inserire nel grafo è  $N(N-1)/2+N$ 
  - Sono inclusi gli archi per collegare i router alle network
- il numero totale di LSA da trasmettere è  $N(N-1)$
- conviene adottare una **topologia a stella equivalente**, inserendo un nodo virtuale che rappresenta la rete
  - solo N archi bidirezionali

Questo calcolo considera sia le connessioni dirette tra router ( $N(N-1)/2$ ) sia gli archi per collegare i router alla rete stessa ( $+N$ ).

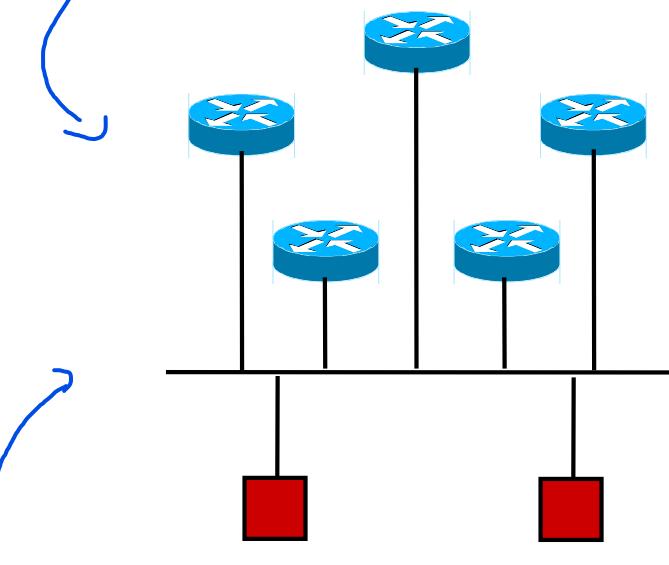
Per ridurre la complessità e migliorare l'efficienza in reti ad accesso multiplo, OSPF adotta una topologia a stella equivalente: Viene introdotto un nodo virtuale che rappresenta la rete. Questo nodo virtuale è rappresentato dal Designated Router (DR).

Il numero di archi bidirezionali si riduce da  $N(N-1)/2$  a N, poiché ogni router ha solo una connessione al DR. Riduzione del numero di LSA necessari, poiché il DR gestisce la distribuzione degli aggiornamenti per tutti i router, evitando un eccessivo traffico di aggiornamento.

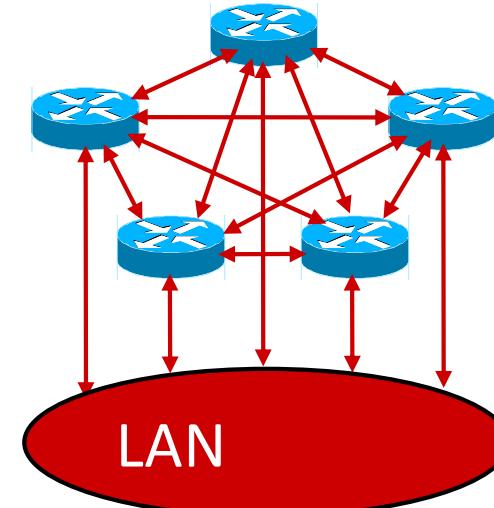
In una rete ad accesso multiplo (sia Broadcast che Non-Broadcast), tutti i router collegati alla rete possono comunicare tra loro. Tuttavia, questo porta a un problema di scalabilità, poiché il numero di connessioni (o archi) tra i router cresce rapidamente all'aumentare del numero di router.

# OSPF: rappresentazione di reti multi-accesso

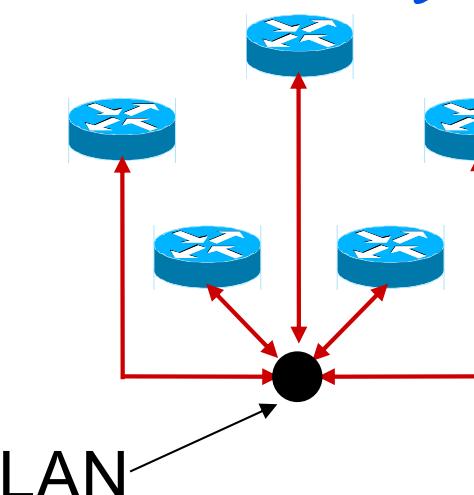
I router in blu sono tutti collegati a una rete comune (rappresentata dalla linea orizzontale nera)



Illustra una situazione in cui tutti i router su una rete multi-accesso hanno connessioni bidirezionali tra di loro. Ciascun router avrebbe bisogno di mantenere una connessione diretta con ogni altro router, portando a un numero elevato di connessioni e aggiornamenti di stato dei link (LSA) da trasmettere. Questo può diventare inefficiente e pesante per la rete.



Quando si dice che più router "condividono lo stesso segmento di rete", significa che essi sono connessi alla stessa rete fisica o logica, spesso rappresentata da una singola subnet o network. In altre parole, questi router possono comunicare direttamente tra loro tramite una connessione comune, come una LAN (ad esempio Ethernet) o un segmento broadcast.



OSPF ottimizza la gestione delle reti multi-accesso introducendo un Designated Router (DR) e una topologia a stella equivalente. In questa rappresentazione, i router sono collegati a un nodo virtuale (indicato dal punto nero), che rappresenta il Designated Router (DR). Invece di avere connessioni dirette con ogni altro router, ogni router stabilisce una connessione bidirezionale solo con il DR. Il DR agisce come punto centrale per la trasmissione delle informazioni di routing e distribuisce gli aggiornamenti a tutti i router collegati, riducendo il traffico di rete e semplificando la topologia.

rappresentazione  
secondo OSPF

# OSPF: vicinanza e adiacenza tra router

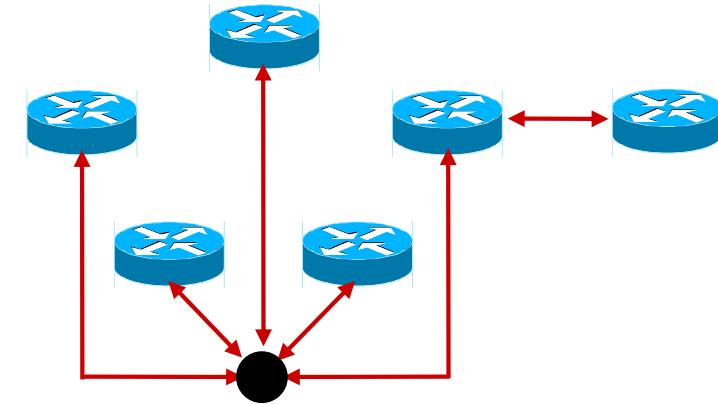
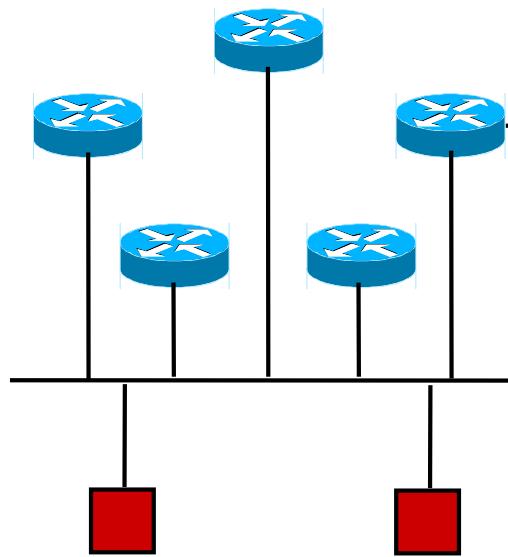
Due router sono considerati vicini se sono connessi alla stessa rete e possono comunicare direttamente tra loro. La vicinanza indica semplicemente la possibilità fisica di comunicazione tra due router, ma non implica che stiano scambiando attivamente informazioni di routing.

- **Vicini:** due router che sono connessi alla medesima rete e possono comunicare direttamente
  - punto-punto o punto-multipunto
- **Adiacenti:** due router che si scambiano informazioni di routing
  - Due router sono considerati adiacenti se hanno stabilito una connessione logica per lo scambio di informazioni di routing. Non tutti i router vicini diventano necessariamente adiacenti. Solo i router che si scambiano attivamente informazioni di routing sono considerati adiacenti.
- In una rete ad accesso multiplo risulta molto più efficiente eleggere un **Designated Router** (DR) fra gli N vicini
  - ogni router della LAN è adiacente solo al DR
  - lo scambio di informazioni di routing avviene solo tra router adiacenti (cioè DR fa da tramite)
  - inoltre il DR è l'unico a comunicare la raggiungibilità di router e host della LAN al mondo esterno
    - Il DR è l'unico router che comunica la raggiungibilità di router e host della LAN verso l'esterno, agendo come un rappresentante della LAN per la rete esterna.
  - Per ragioni di affidabilità occorre avere anche un **Backup Designated Router** (BDR) adiacente a tutti i router locali

Tutti i router della LAN diventano adiacenti solo al DR, invece di diventare adiacenti tra loro. In questo modo, il DR raccoglie le informazioni di routing da ciascun router e le distribuisce agli altri router, riducendo il numero di connessioni e messaggi necessari.

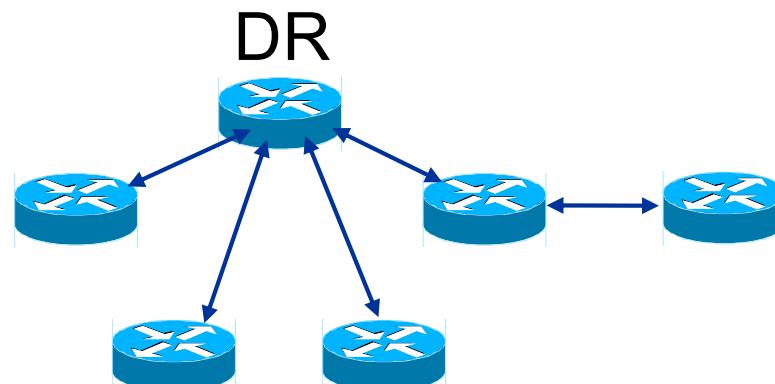
Per garantire la continuità e l'affidabilità in caso di malfunzionamento del DR, viene eletto un BDR. Il BDR mantiene una copia delle informazioni di routing e può assumere immediatamente il ruolo di DR se il DR principale diventa inaccessibile o non funziona correttamente. Il BDR è adiacente a tutti i router della LAN, preparandosi a prendere il controllo se necessario.

# OSPF: vicinanza e adiacenza tra router



router vicini

Tutti i router sono vicini, cioè fanno parte della stessa rete e possono comunicare tra loro



router adiacenti

Tutti i router sono adiacenti SOLO con il DR e BDR

# OSPF: identificazione di router e priorità

Ogni router deve avere un identificativo univoco all'interno dell'Autonomous System (AS) chiamato Router ID

- Ogni router di un AS utilizzante OSPF deve avere un identificativo univoco (**router ID**):

Metodo Automatico

- di default si prende l'indirizzo IP più alto fra quelli assegnati alle interfacce <sup>attive</sup> del router

Metodo Manuale

- si può assegnare manualmente un router ID ad ogni router configurando opportunamente l'interfaccia di loop-back
- configurare l'interfaccia di loop-back è un modo più stabile e sicuro di assegnare il router ID perché questa interfaccia non viene mai disabilitata

Poiché l'interfaccia di loopback è virtuale e non dipende dalla connessione fisica, il Router ID assegnato tramite questa interfaccia non viene mai disabilitato (a meno che non venga fatto esplicitamente)

- Ai singoli router di un'area possono essere associate delle priorità

- utilizzate nell'elezione del DR e del BDR
- valore di priorità compreso tra 0 e 255 (8 bit)
- di default tutti i router hanno priorità 0 (più bassa)

Durante l'elezione del DR, il router con la priorità più alta viene selezionato come DR. Se c'è un pareggio (stessa priorità tra più router), viene scelto il router con il Router ID più alto. Il BDR viene scelto in modo simile, come secondo router con priorità più alta.

# OSPF: elezione di DR e BDR

- Ciascun router nella rete ad accesso multiplo:
  - esamina la lista dei suoi vicini
  - elimina dalla lista tutti i router non eleggibili (ad esempio tutti quelli che hanno priorità nulla)
  - fra quelli rimasti seleziona il router avente la priorità maggiore
    - il più alto router ID in caso di uguale priorità
  - elegge il router selezionato a DR
  - rivede la tabella dei vicini e riselecta gli eleggibili (a questo punto il router che è stato eletto DR non è più eleggibile)
  - seleziona ed elegge il BDR secondo le regole già adottate per il DR
  - termina la procedura una volta eletti DR e BDR

# OSPF: Link State Database

Il Link State Database è una rappresentazione della topologia di rete condivisa tra tutti i router OSPF all'interno di un'area. Ogni router mantiene una copia del LSDB che contiene informazioni dettagliate sui link (collegamenti) nella rete.

- Il grafo orientato della rete sul quale ciascun router calcola lo **shortest path tree** è rappresentato dal **Link State Database** presente in ogni router
  - Nodi:

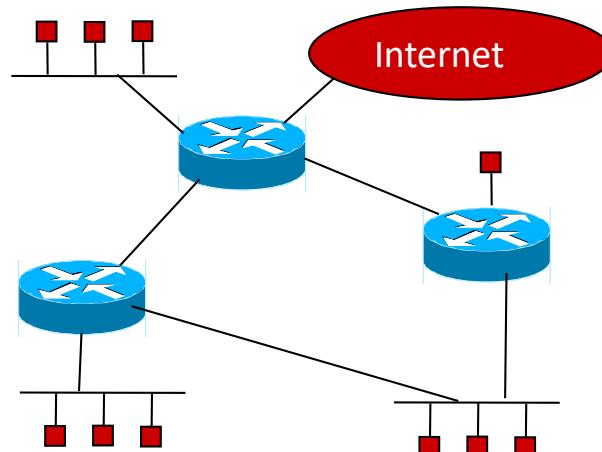
Le destinazioni che si trovano al di fuori dell'Autonomous System (AS).

- router
- reti o host singoli
- nodi virtuali delle topologie a stella equivalenti

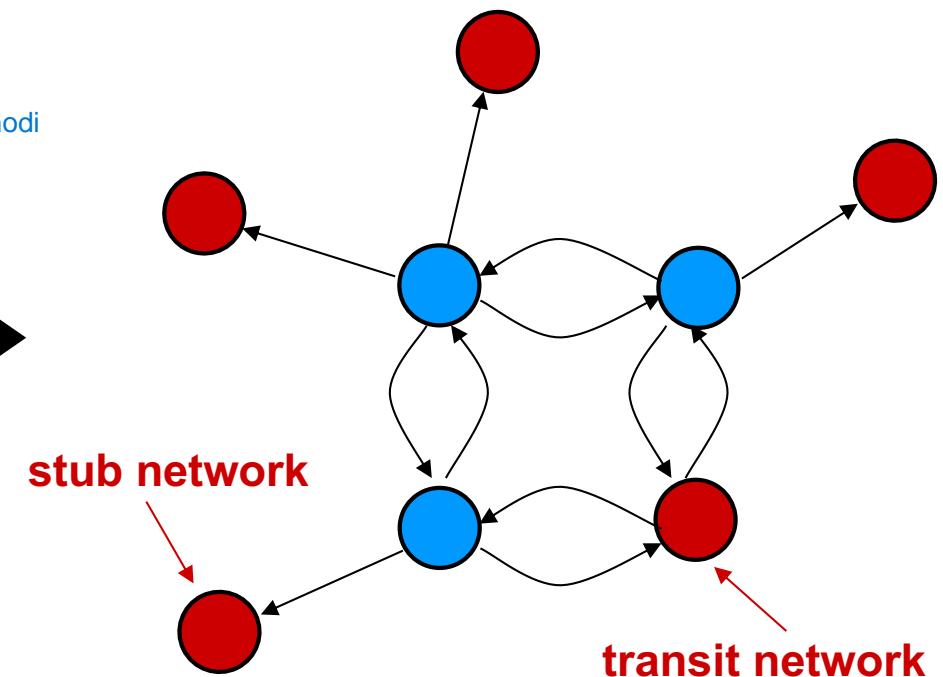
destinazioni esterne

– Archi:

- collegamenti fisici o virtuali tra i nodi



Per reti multi-accesso (come LAN), OSPF utilizza un nodo virtuale che rappresenta il Designated Router (DR) e BDR per ridurre la complessità della topologia.

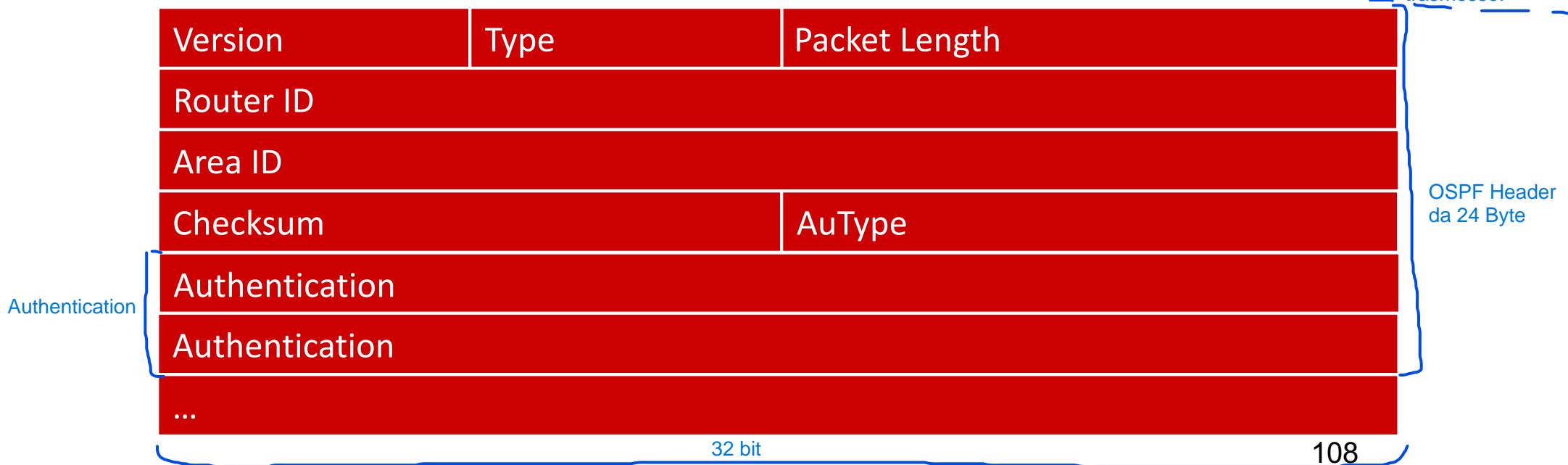


# OSPF: i protocolli

- Hello: Utilizzato per scoprire e mantenere i vicini OSPF, verificando che i router siano attivi e operativi. È fondamentale per stabilire le adiacenze tra router.
- Exchange: Durante questa fase, i router adiacenti scambiano informazioni dettagliate sulla topologia della rete, aggiornando le proprie tabelle di stato dei link (Link State Database).
- Flooding: Consente la propagazione di aggiornamenti di stato dei link a tutti i router della rete, in modo che ognuno possa mantenere una mappa aggiornata della topologia.

- OSPF invia messaggi utilizzando direttamente il protocollo IP (campo protocol = 89)
- Si compone di tre sottoprotocolli:
  - hello, exchange, flooding
- Tutti i messaggi hanno una intestazione comune
  - vengono aggiunte informazioni per il particolare scopo a cui il messaggio è destinato (tipo di pacchetto)

Tutti i messaggi OSPF hanno una struttura di intestazione comune che contiene i campi essenziali per identificare il tipo di pacchetto e garantire l'integrità e la sicurezza delle informazioni trasmesse.



# OSPF: intestazione comune

- **Version** indica la versione di OSPF (versione 2)  
la versione attuale è OSPFv2.
- **Type** indica il tipo di pacchetto
- **Packet Length** numero di byte del pacchetto  
Rappresenta la lunghezza totale del pacchetto, in byte, inclusi intestazione e dati.
- **Router ID** indirizzo IP che identifica il router mittente
- **Area ID** identifica l'area di appartenenza
  - il numero 0.0.0.0 è l'area di backboneIdentifica l'area OSPF a cui appartiene il pacchetto.  
Ogni area ha un identificatore univoco
- **Checksum** calcolata su tutto il pacchetto OSPF escludendo gli 8 byte del campo authentication
  - si utilizza l'algoritmo classico di IPUtilizza l'algoritmo classico di checksum di IP per verificare l'integrità del pacchetto e rilevare eventuali errori durante la trasmissione.
- **AuType** indica il tipo di autenticazione:
  - 0 nessuna autenticazione
  - 1 autenticazione semplice (password nel campo **authentication**)
  - 2 autenticazione crittografica (dati nel campo **authentication**)



# Type

- **Type 1** Utilizzato per la scoperta dei vicini e per mantenere la connettività tra router.
  - Hello (Hello protocol, neighbour discovery)
- **Type 2** Inizia il processo di scambio di informazioni sulla topologia tra router adiacenti.
  - Database description (exchange protocol)
- **Type 3** Pacchetto che richiede informazioni specifiche non presenti o aggiornate nel database del router richiedente.
  - Link state request
- **Type 4** Propaga le informazioni di stato dei link attraverso la rete.
  - Link state update
- **Type 5** Conferma la ricezione dei pacchetti di aggiornamento.
  - Link state acknowledge

Il pacchetto Hello permette ai router di scoprire altri router vicini e di definire DR e BDR. È essenziale per iniziare e mantenere le connessioni tra router, verificando la presenza e l'attività dei vicini.

Questo pacchetto fornisce una panoramica delle informazioni contenute nel Link State Database (LSDB) di ciascun router. I router utilizzano il pacchetto Database Description per descrivere i contenuti del proprio database di stato dei link agli altri router, aiutando a sincronizzare le informazioni.

Se un router rileva che gli mancano alcuni dati di stato dei link che un router adiacente possiede, invia un pacchetto di Link State Request per ottenere queste informazioni specifiche.

Questo pacchetto contiene le informazioni aggiornate sui link e viene utilizzato per diffondere queste informazioni a tutti i router della rete, assicurando che ogni router mantenga una mappa aggiornata della topologia. Il pacchetto Link State Update trasmette esclusivamente le informazioni aggiornate, evitando la trasmissione dell'intero LSDB.

Quando un router riceve un pacchetto di Link State Update, invia un pacchetto di Link State Acknowledgment come conferma. Questo meccanismo evita che vengano inviati duplicati e assicura che tutte le informazioni siano ricevute correttamente.

# OSPF: Hello protocol

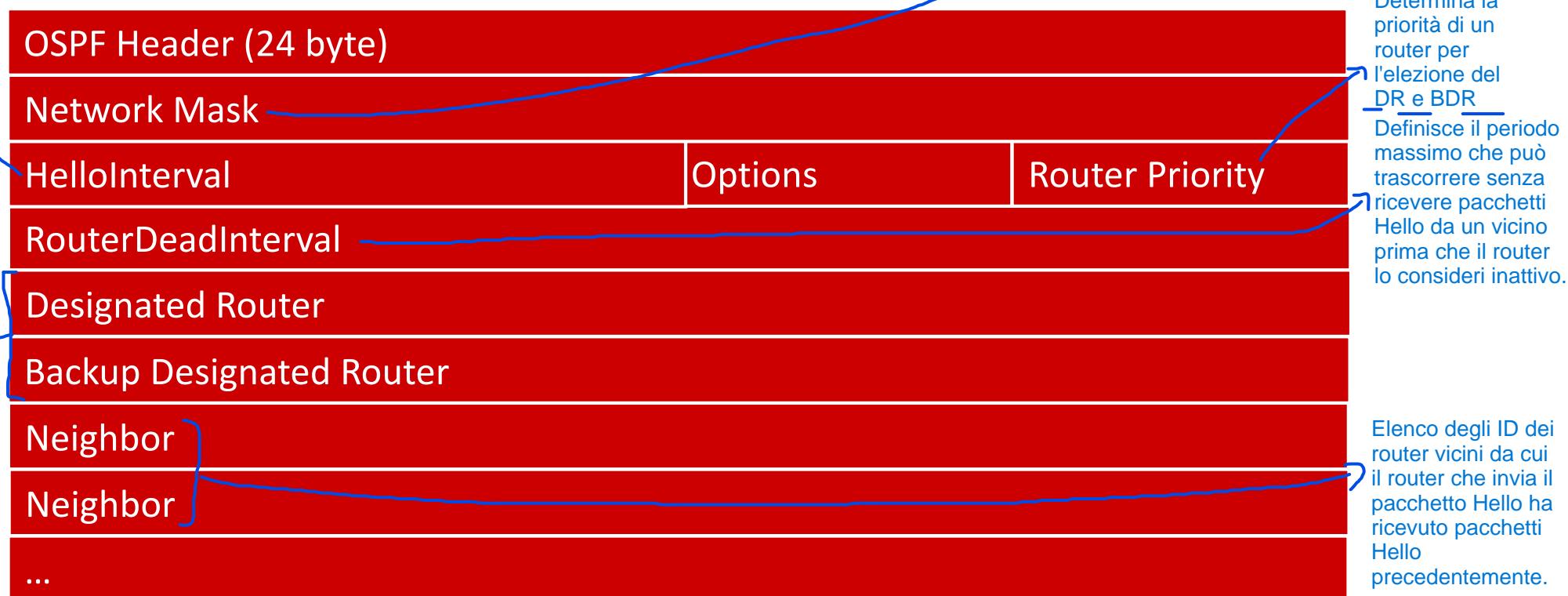
→ Assicura che i router rilevino e monitorino i loro vicini e stabiliscano relazioni robuste e affidabili.

- Unico tipo di pacchetto: **Hello** (Type = 1)

- Utilizzato per:

- controllare l'operatività dei link
- scoprire e mantenere relazioni fra vicini
- eleggere DR e BDR

Indica l'intervallo di tempo (in secondi) tra l'invio di pacchetti Hello. I router devono utilizzare lo stesso intervallo per formare una relazione di vicinato.



Specificano l'ID dei router eletti come DR e BDR per la rete multi-accesso. Questi campi aiutano i router a sincronizzarsi e a scegliere il router responsabile per la propagazione delle informazioni di stato dei link nella rete.

# OSPF: Hello protocol

I router inviano periodicamente pacchetti Hello attraverso le loro interfacce abilitate per il protocollo (continuano a scambiarsi pacchetti Hello per tutta la durata della loro operatività)

- I pacchetti HELLO sono inviati sulle interfacce periodicamente secondo quanto specificato dal parametro **HelloInterval**
  - si riescono così a scoprire i propri vicini
- Includono una lista di tutti i vicini (**Neighbor**) dai quali è stato ricevuto un pacchetto HELLO recente (cioè non più vecchio di **RouterDeadInterval**)
  - si riesce così a conoscere se per ciascun vicino è presente un collegamento bidirezionale e se esso è ancora attivo
- I campi **Router Priority**, **Designated Router** e **Backup Designated Router** sono utilizzati per l'elezione di DR e BDR
- **Network Mask** indica la maschera relativa all'interfaccia del router (l'indirizzo è nell'header IP)
- **Options** indica se si supportano funzionalità opzionali

# OSPF: Exchange protocol

Il protocollo di Exchange in OSPF è il processo attraverso cui due router adiacenti sincronizzano i loro Link State Databases (LSDB) per assicurarsi di avere una visione coerente della rete.

- Una volta stabilite le adiacenze, router adiacenti devono sincronizzare i rispettivi Link State Database
- La procedura di sincronizzazione è asimmetrica
  - si stabilisce chi è il master e chi lo slave All'inizio, si decide quale router sarà il "master" e quale lo "slave" per il processo di scambio.
  - il master invia una serie di pacchetti **Database Description** (Type = 2) contenenti l'elenco dei LSA del proprio database
    - nell'elenco sono indicati il tipo di LSA, l'età<sup>del LSA</sup>, il router che lo ha generato e il numero di sequenza
    - non ci sono i dati relativi al LSA
  - lo slave risponde con l'elenco dei LSA del suo database
  - durante lo scambio ciascuno dei due router confronta le informazioni ottenute con quelle in proprio possesso
  - se nel proprio database ci sono dei LSA meno recenti rispetto all'altro, questi (e solo questi) vengono richiesti con un successivo pacchetto **Link State Request** (Type = 3)

Questi pacchetti contengono un elenco degli LSA (Link State Advertisements) che il router possiede, ma non includono i dettagli completi di ciascun LSA.

Sia MASTER sia SLAVE, mandano pacchetti che contengono solo le intestazioni degli LSA presenti nei rispettivi LSDB, senza includere i dettagli completi degli LSA.

Lo slave risponde con il suo elenco di LSA per confronto.

Durante questo scambio, ciascun router confronta il proprio LSDB con quello ricevuto: se un router ha LSA più vecchi rispetto a quelli posseduti dall'altro (sia MASTER sia SLAVE), richiederà i dettagli di tali LSA. I router richiedono gli LSA aggiornati tramite pacchetti Link State Request (Type 3). Una volta ricevuto il pacchetto LSR dal master, lo slave risponde con un pacchetto Link State Update (LSU), che contiene i dettagli completi degli LSA richiesti dal master. Una volta ricevuto il pacchetto LSU con i dettagli completi degli LSA, il master invia un pacchetto di Link State Acknowledgment (LSAck) per confermare la ricezione corretta degli LSA inviati dallo slave. Il master può continuare a inviare richieste LSR se trova ulteriori LSA nello slave che necessitano di aggiornamenti.

I numeri di sequenza di un Link-State Advertisement (LSA) in OSPF sono valori utilizzati per tenere traccia della "versione" di ciascun LSA. Ogni volta che un LSA viene aggiornato, il numero di sequenza viene incrementato.

# OSPF: Flooding protocol

Il Flooding Protocol in OSPF gestisce la diffusione delle informazioni di stato di collegamento (LSA) in modo che ogni router della rete abbia una visione aggiornata della topologia.

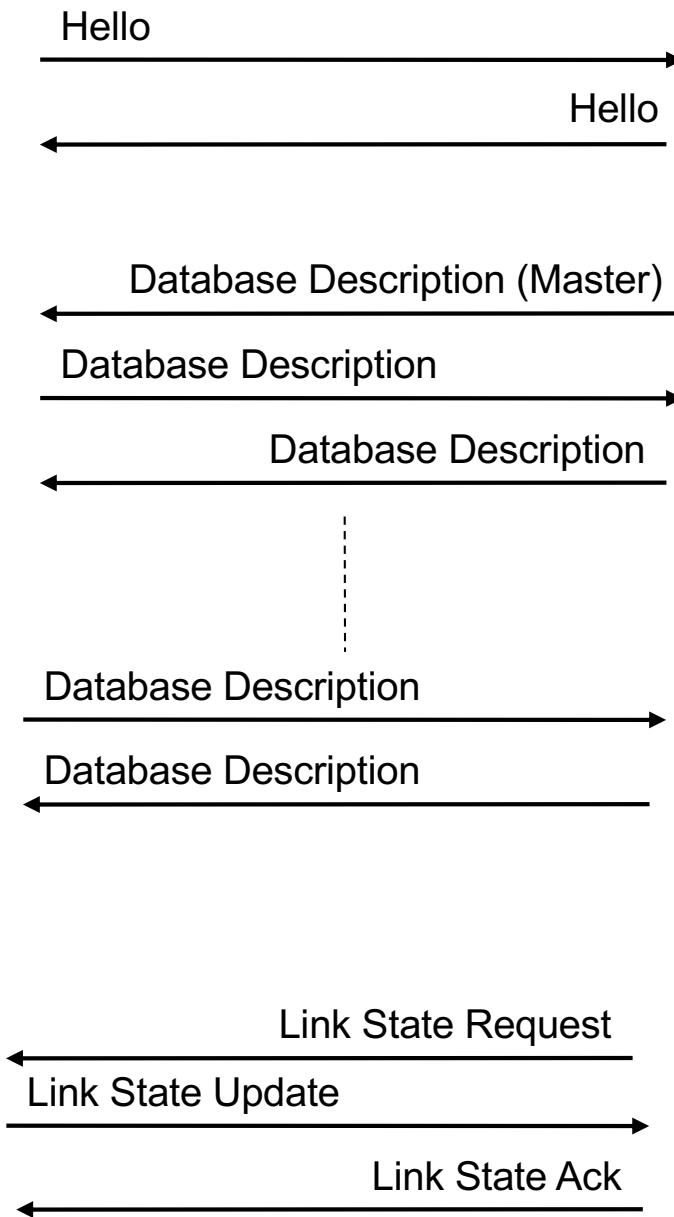
Questo pacchetto contiene gli LSA aggiornati, con informazioni come la sequenza e l'età dell'LSA per distinguere i dati recenti da quelli obsoleti.

- La diffusione dei LSA a tutti i router della rete avviene tramite l'invio di pacchetti **Link State Update** (Type = 4)
  - a fronte di un cambiamento nello stato di un collegamento
  - a fronte di una **Link State Request** (richiesta esplicita di aggiornamento)
  - periodicamente (ogni 30 minuti)
- Si esegue in modalità flooding per fare in modo che tutti i router vedano gli aggiornamenti
  - flooding efficiente: si usano i numeri di sequenza dei LSA
- Si continua ad inviare lo stesso update finché non viene confermata la sua ricezione dai nodi adiacenti tramite il pacchetto **Link State Acknowledgment** (Type = 5)
  - in questo modo si rende il flooding affidabile

Ogni router che riceve un LSU invia un Link State Acknowledgment (LSAck) (Tipo 5) al router che ha inviato l'aggiornamento, confermando di aver ricevuto correttamente l'informazione. Se un router non riceve l'ack per un update inviato, continuerà a inviarlo fino a quando la ricezione non viene confermata.

Ogni router propaga l'LSU ai propri vicini, e questi router a loro volta inoltrano gli aggiornamenti fino a coprire tutta la rete. In questo modo, tutti i router ricevono l'informazione. Il flooding usa i numeri di sequenza degli LSA per garantire che solo le informazioni più recenti vengano propagate.

# OSPF: sincronizzazione e aggiornamento



## Fase di Hello

I router scoprono l'esistenza reciproca

## Fase di Exchange

Si sceglie il master e lo slave

Si confrontano i database

## Fase di Update

Si inviano richieste di aggiornamento ai  
router adiacenti per aggiornare il database



# Stub Area

La Stub Area in OSPF è un tipo speciale di area di routing utilizzata per ottimizzare l'uso delle risorse, specialmente in aree che hanno un solo punto di uscita verso altre aree

- **Stub Area** = tipicamente un'area con uno solo punto di interconnessione con il resto della rete tramite un Area Border Router
- Instradamento verso l'esterno della stub area
  - Viene effettuato con la tecnica del “default route”
  - I percorsi verso network esterne alla stub area non vengono propagati da OSPF internamente alle stub areas.
- Vantaggi
  - Dimensioni molto ridotte della tabella di routing
  - Il router di bordo necessita di poca memoria
- Default route
  - Esiste un solo punto di uscita verso tutte le destinazioni possibili

I router all'interno della Stub Area (specialmente quelli di confine) necessitano di meno memoria, rendendo la gestione più efficiente.

La Stub Area dispone di un unico punto di uscita per tutto il traffico destinato fuori dall'area. Questo punto di uscita è generalmente l'ABR, che instrada il traffico verso il resto della rete OSPF.



ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

# Exterior Gateway Protocols EGP



# Exterior Gateway Protocols

- I protocolli di tipo EGP sono diversi da quelli di tipo IGP
- All'interno di un AS si persegue l'ottimizzazione dei percorsi
- Nel routing tra diversi AS si deve tener conto anche (e soprattutto) delle **politiche di instradamento**
  - ogni AS vuole mantenere una propria autonomia ed indipendenza dagli altri e non vuole subire decisioni prese da altri
  - alcuni AS non vogliono permettere ad altri AS di instradare il traffico attraverso le loro reti
  - in altri casi bisogna operare secondo accordi internazionali
- Per Internet sono stati definiti due protocolli di tipo EGP:
  - **Exterior Gateway Protocol** (EGP)
  - **Border Gateway Protocol** (BGP)



# EGP: Exterior Gateway Protocol

- Primo protocollo tra AS
  - risale ai primi anni ottanta (RFC 827)
- Caratterizzato da tre funzionalità principali:
  - **neighbor acquisition**
    - verificare se esiste un accordo per diventare vicini
  - **neighbor reachability**
    - monitorare le connessioni con i vicini
  - **network reachability**
    - scambiare informazioni sulle reti raggiungibili da ciascun vicino
- EGP è simile ad un protocollo distance vector
  - le informazioni inviate ai vicini sono sostanzialmente informazioni di raggiungibilità
  - non sono specificate le regole per definire le distanze
  - la distanza minima può non essere il criterio migliore da seguire



# EGP: limiti

- EGP fu progettato per una topologia assai specifica,
  - una dorsale di Internet, la rete ARPAnet
  - vari domini connessi alla dorsale attraverso un unico router
- Funziona bene per una topologia ad albero, ma non per reti a maglia complessa (presenza di cicli)
  - la convergenza del protocollo può essere molto lenta
  - si possono facilmente creare instabilità
- Non si adatta velocemente alle modifiche della topologia
- EGP non implementa alcun meccanismo di sicurezza
  - qualunque malintenzionato può annunciare quello che vuole ed essere creduto dai router
  - un router guasto può danneggiare il routing di tutta la rete

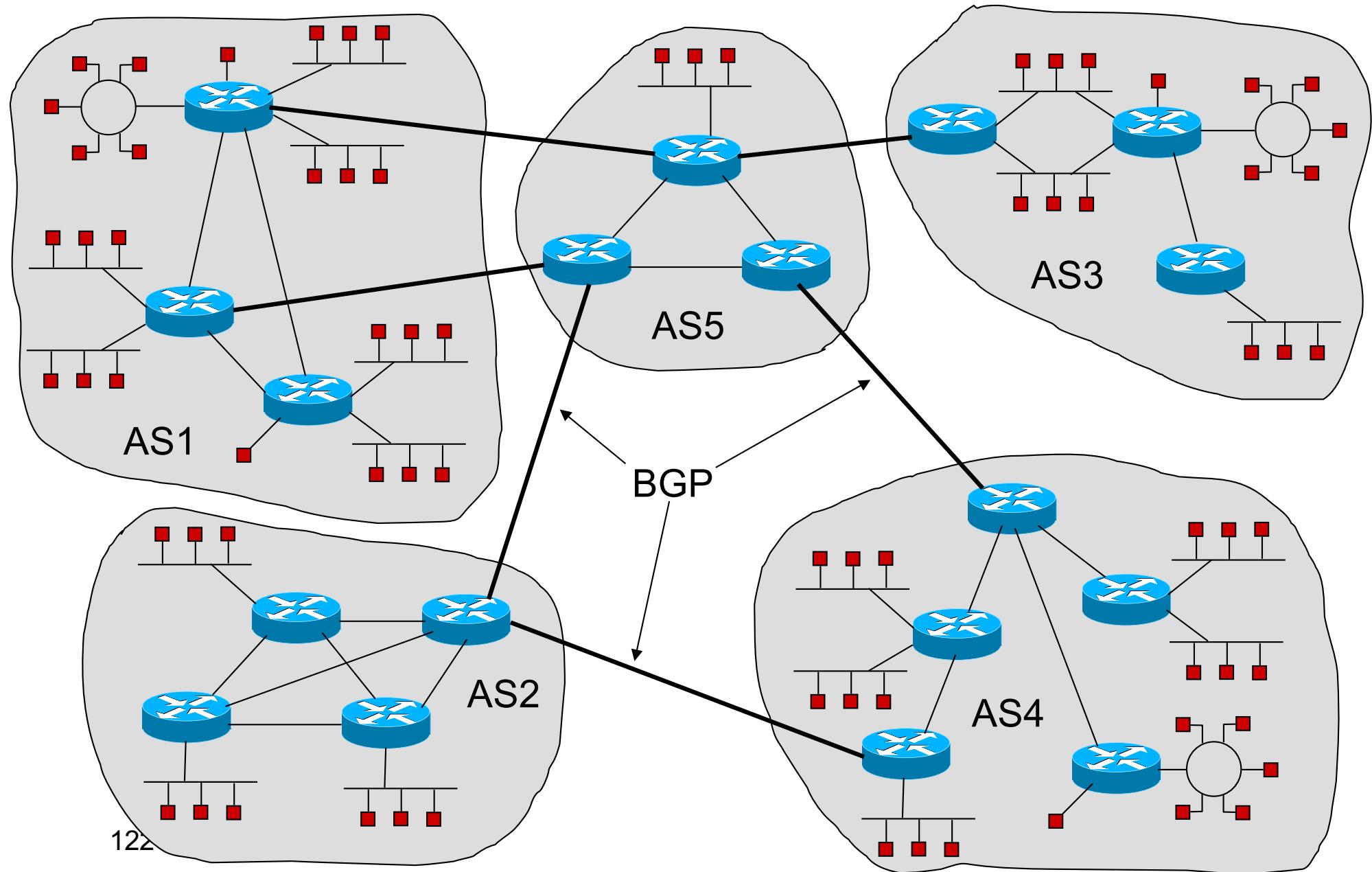


# BGP: Border Gateway Protocol

- **BGP** è stato concepito come sostituto di EGP
- Oggi è in uso la versione 4 (RFC 1771)
- I router BGP si scambiano informazioni attraverso connessioni TCP (porta 179) chiamate **sessioni BGP**
  - le comunicazioni sono affidabili
  - funzionalità di controllo degli errori demandate allo strato di trasporto  
→ BGP più semplice
- Si distinguono due tipi di sessioni BGP:
  - sessioni BGP **esterne (eBGP)** instaurate tra router BGP appartenenti ad AS diversi
  - sessioni BGP **interne (iBGP)** instaurate tra router BGP appartenenti allo stesso AS
- Le informazioni scambiate riguardano la raggiungibilità di reti IP secondo lo schema classless (CIDR)

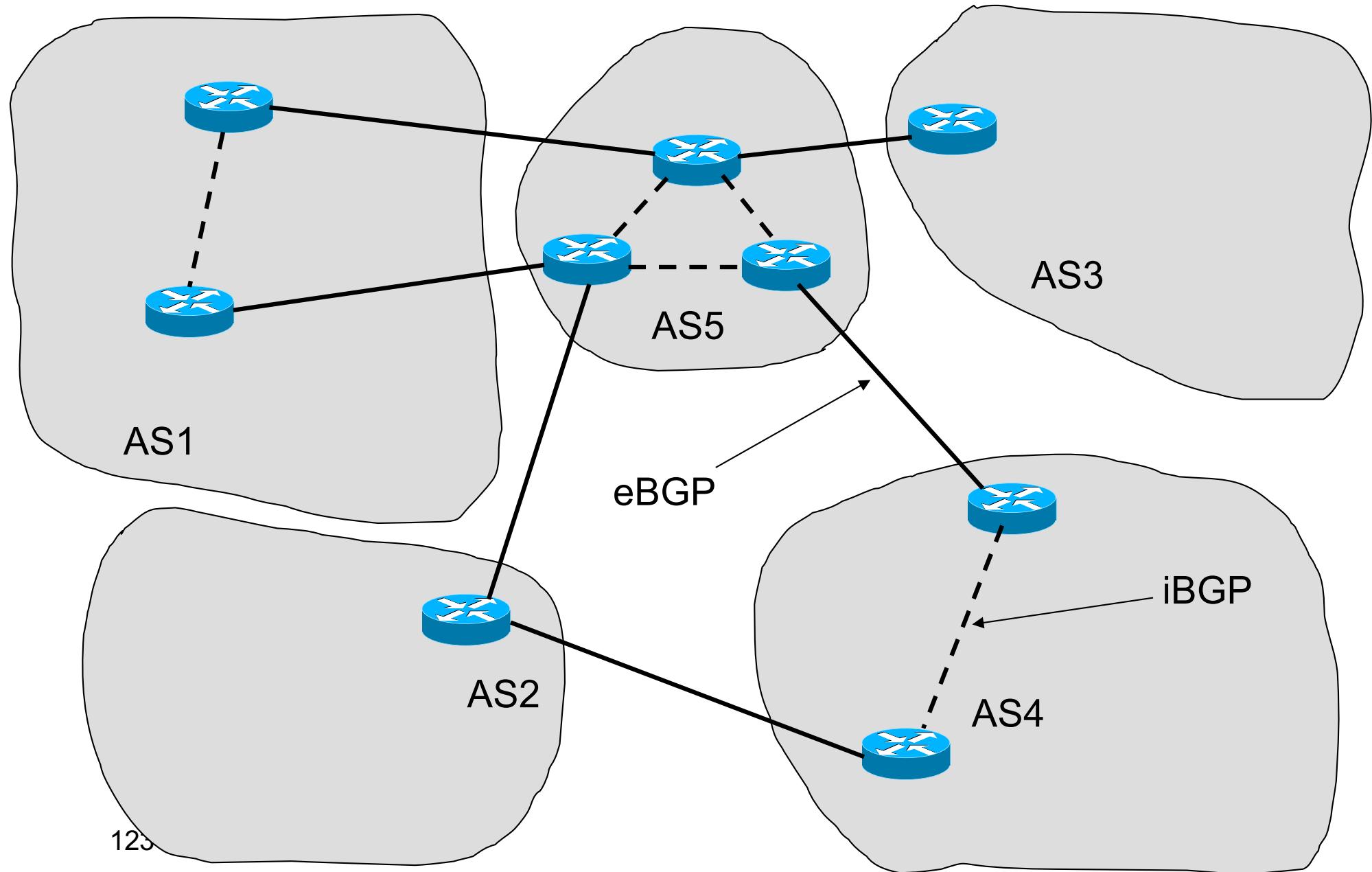


# BGP: interconnessione tra AS





# BGP: sessioni interne ed esterne





# BGP: Path Vector

- BGP è un protocollo di tipo **Path Vector**
  - evoluzione del Distance Vector
  - nel vettore dei percorsi si elencano tutti gli AS da attraversare per raggiungere una destinazione
  - risolve il problema dei percorsi ciclici
  - più consono a definire le politiche di routing tra AS rispetto alla semplice distanza
- Come si evitano i cicli:
  - quando un router di bordo di un AS riceve un path vector controlla se il suo AS è già elencato al suo interno
  - se lo è significa che esiste la possibilità di un loop e quel path vector non viene considerato
  - altrimenti il path vector viene aggiornato con l'indicazione dell'AS di appartenenza e comunicato ai vicini, in quanto considerato corretto

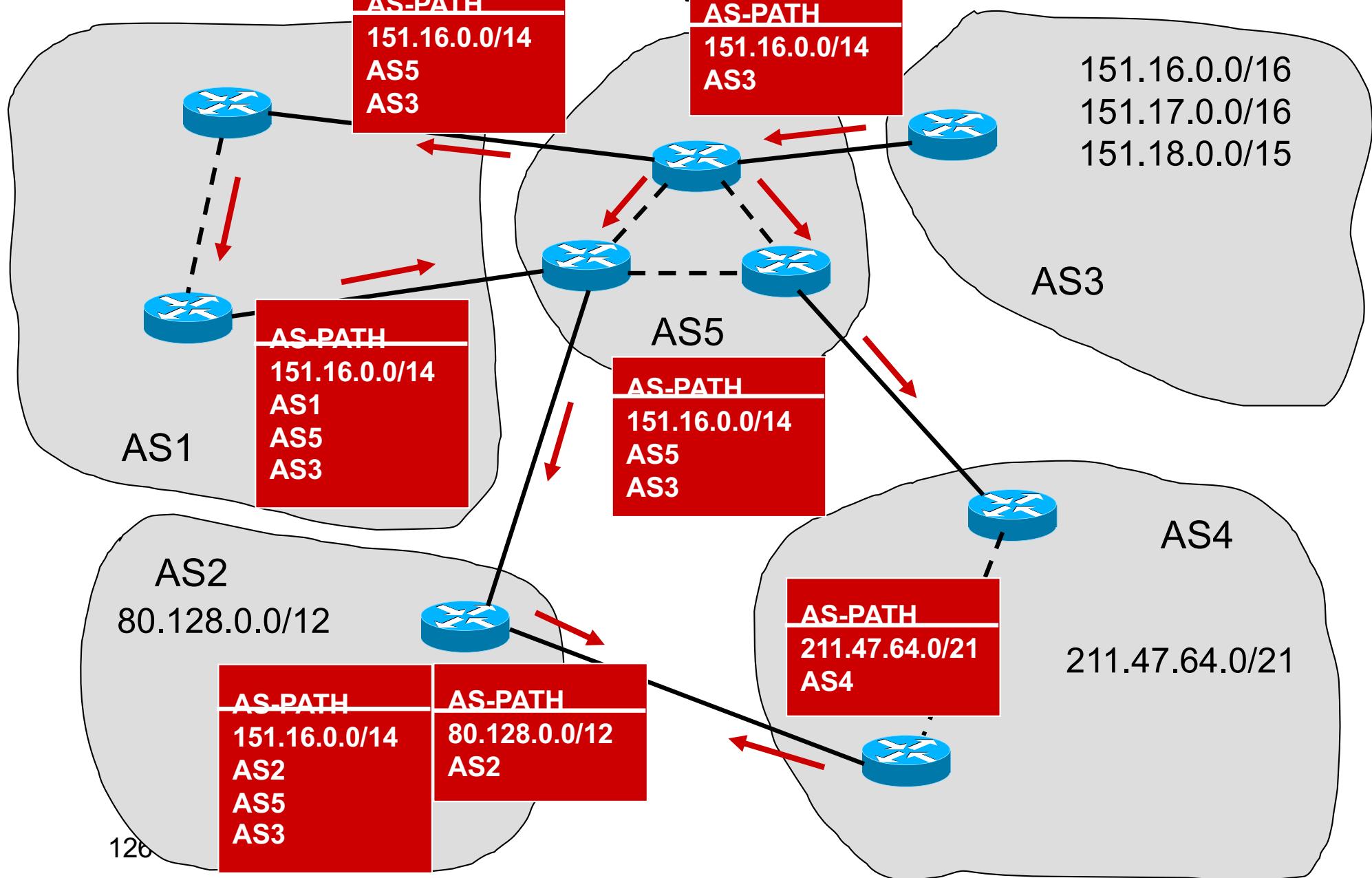


# BGP: Path Vector

- Come si applicano le politiche di routing:
  - si comunicano ai vicini solo i path vector relativi alle destinazioni verso le quali si vuole permettere il transito (**export policies**)
  - dal path vector è possibile risalire agli AS da attraversare per raggiungere una destinazione: se nel path vector ricevuto da un vicino sono presenti uno o più AS incompatibili con le politiche di routing stabilite, esso viene ignorato (**import policies**)
- L'approccio basato sul percorso invece che sulla distanza non richiede che tutti i router usino la stessa metrica → possibilità di scelte arbitrarie
- Maggior consumo di banda per le informazioni di routing
- Maggiori requisiti di memoria nei router per mantenere le tabelle



# BGP: scambio di path vector





- 000100 00 -> 16
  - 000100 01 -> 17
  - 000100 1 0 -> 18
  - 000100 1 1 -> 19
- 
- **151.000100 00**
  - 151.16.0.0/14



# BGP: attributi

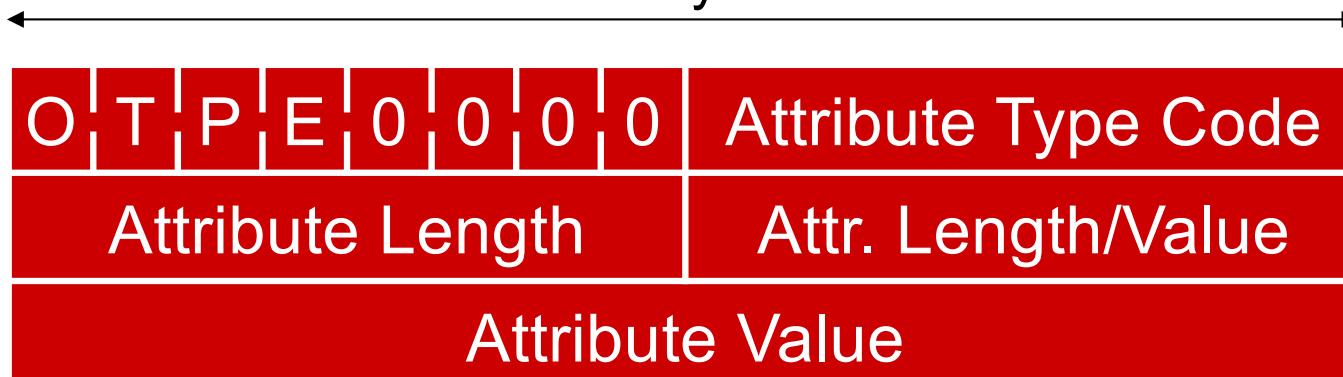
- A ciascun path vector sono associati degli **attributi** che ne specificano la natura (ad es. il “path” è un attributo)
- Un determinato attributo può essere:
  - **well-known**: riconoscibile da tutte le implementazioni BGP, deve essere inoltrato assieme al path vector (dopo un eventuale aggiornamento)
    - **mandatory**: deve essere presente nel path vector
    - **discretionary**: può anche non essere indicato
  - **optional**: può non essere riconosciuto da alcuni router
    - **transitive**: deve essere inoltrato anche se non riconosciuto
    - **non-transitive**: deve essere ignorato se non riconosciuto
  - **partial**: si tratta di un attributo optional-transitive che è stato ritrasmesso senza modifiche da un router perché non lo ha riconosciuto (indica se un determinato path vector è stato riconosciuto o meno da tutti i router attraversati)



# BGP: codifica degli attributi

- All'interno di un path vector, gli attributi sono codificati da una struttura di lunghezza variabile

2 byte



- O = 1 → optional                                    O = 0 → well-known
- T = 1 → transitive                                    T = 0 → non-transitive
- P = 1 → partial
- E = 1 → attribute length = 2 byte
- E = 0 → attribute length = 1 byte



# BGP: alcuni attributi

- **Origin** (Code = 1): è well-known mandatory e può valere:
  - **0 = IGP**: l'informazione è stata ottenuta direttamente dal protocollo di routing operante all'interno dell'AS in cui si trova la destinazione e per cui la si ritiene veritiera
  - **1 = EGP**: l'informazione è stata appresa dal protocollo EGP, che non funziona se vi sono cicli → un percorso caratterizzato da questo valore è peggiore di uno di tipo IGP
  - **2 = incomplete**: serve ad indicare che il percorso è stato determinato in altro modo (es. statico) oppure è utilizzato per marcare un percorso di AS che è stato troncato perché la destinazione è al momento non raggiungibile
- **AS path** (Code = 2): è well-known mandatory
  - consiste nell'elenco degli AS da attraversare lungo il percorso verso la destinazione
- **Next hop** (Code = 3): è well-known mandatory
  - indica l'indirizzo IP del router di bordo dell'AS che deve essere usato come next hop verso la destinazione specificata



# BGP: formato dei messaggi

# byte	HEADER COMUNE
16	Marker
2	Length
1	Type

Tutti i messaggi hanno la seguente parte comune:

- **Marker**: campo per possibile schema di autenticazione
- **Length**: numero di byte del messaggio BGP, header incluso
- **Type**: assume uno dei seguenti valori:
  - Open
  - Update
  - Notification
  - Keepalive



# BGP: tipi di messaggio

- **Open**: primo messaggio trasmesso quando viene attivata una connessione verso un router BGP vicino, contiene
  - informazioni di identificazione dell'AS di chi trasmette
  - durata del timeout per considerare un vicino non più attivo
  - dati di autenticazione
- **Update**: contiene il path vector e i relativi attributi
- **Notification**: messaggio di notifica di errori e/o di chiusura della connessione
- **Keepalive**: non contiene informazioni aggiuntive, ma è usato per comunicare ad un router BGP vicino, in assenza di nuove informazioni di routing, che il trasmettitore è comunque attivo, anche se silente