



ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

# Instradamento nelle reti a pacchetto e in Internet

Franco CALLEGATI

Walter CERRONI



# Funzioni di IP

- Indirizzamento
- Frammentazione
- 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 tabelle
- 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
  - Instradamento dinamico a distanza minima



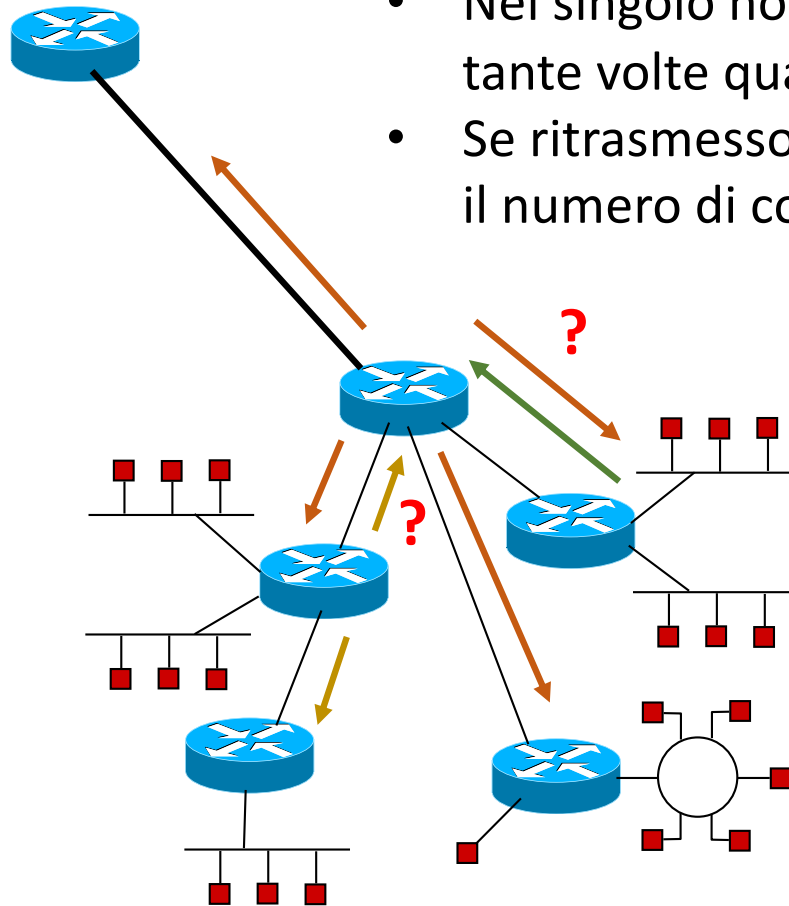
# Flooding

- 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

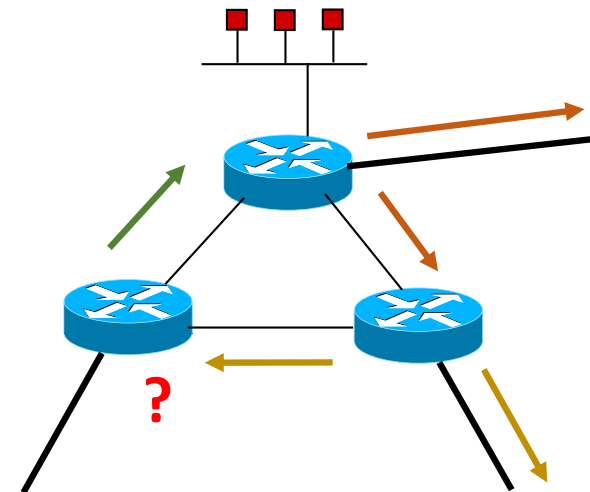
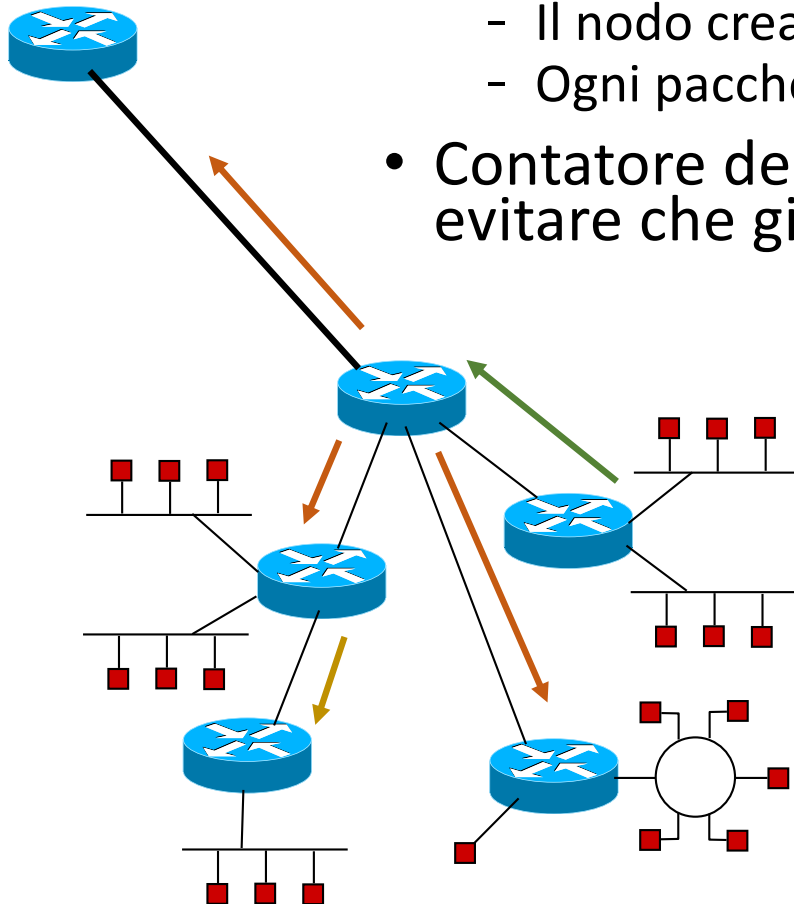
## Proliferazione dei pacchetti

- Nel singolo nodo ogni pacchetto viene copiato tante volte quante sono le interfacce
- Se ritrasmesso sull'interfaccia da cui è arrivato il numero di copie cresce esponenzialmente



# Soluzioni

- Un nodo non ritrasmette il pacchetto nella direzione dalla quale è giunto
- 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
- Contatore del tempo di vita (TTL) di un pacchetto per evitare che giri all'infinito





# Dinamicità

- 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)
  - 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



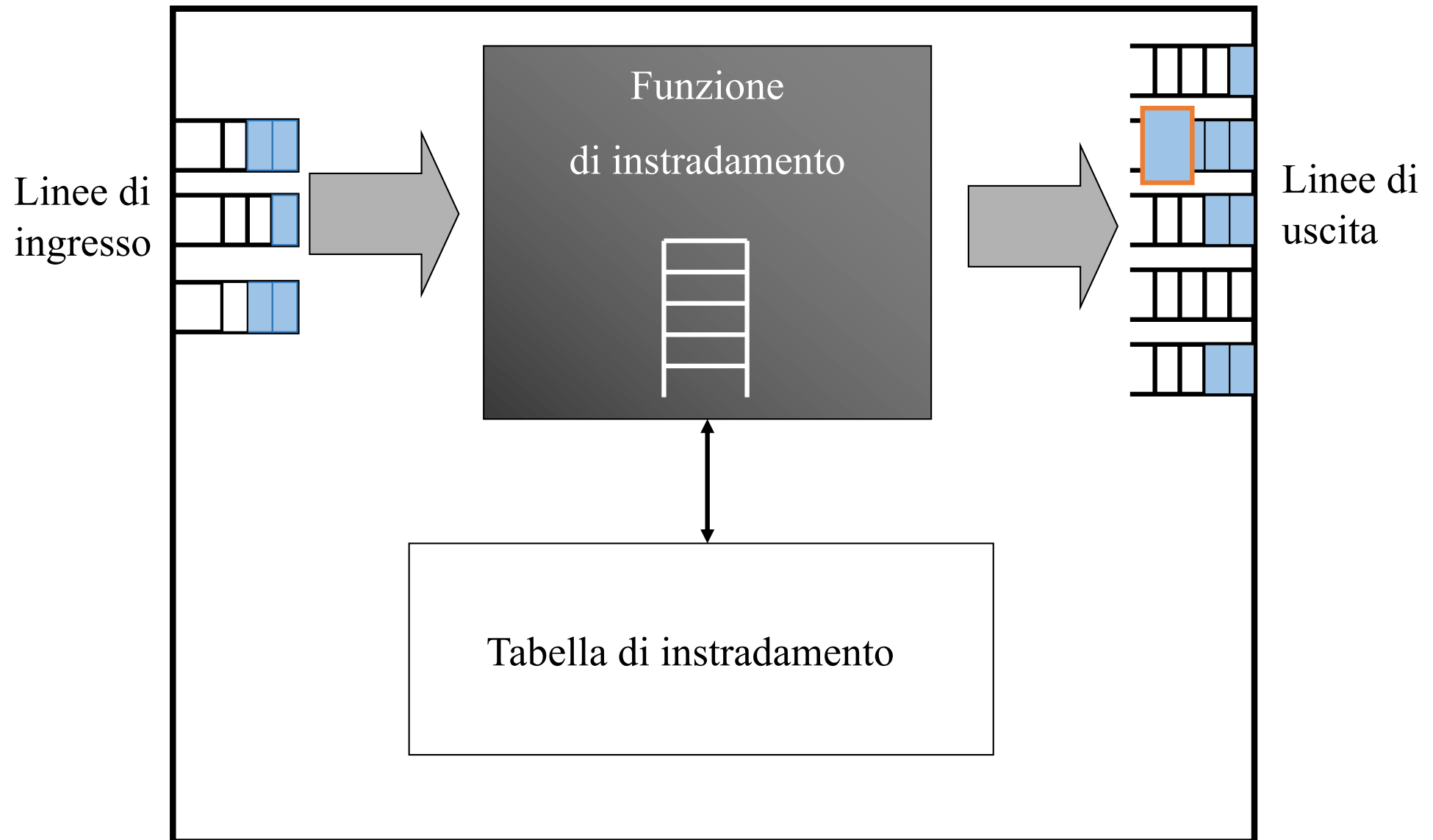
# Random

- 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
- Si deve prevedere un meccanismo per limitare il tempo di vita dei pacchetti
- Non tiene conto della destinazione finale del pacchetto

# Instradamento con tabella



# Store-and-Forward

- 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 se
    - 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



# Rappresentazione della rete

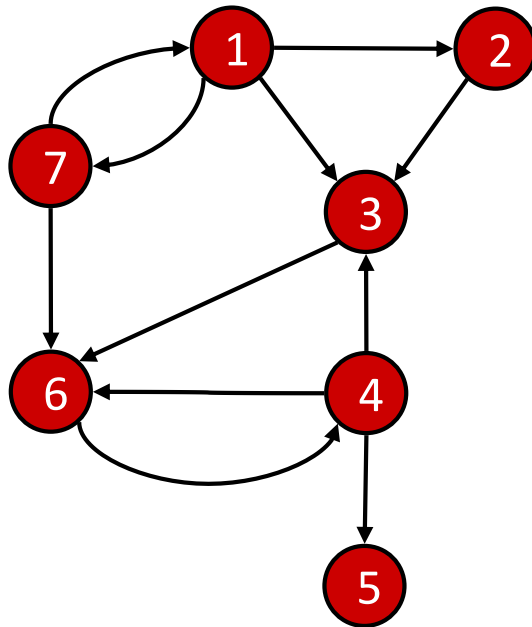
- Ad una generica rete si può facilmente associare un grafo orientato:
  - i nodi rappresentano i terminali ed i commutatori
  - gli archi rappresentano i collegamenti
  - L'orientazione degli archi rappresenta la direzione di trasmissione
  - il peso degli archi rappresenta il costo dei collegamenti, 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

# 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

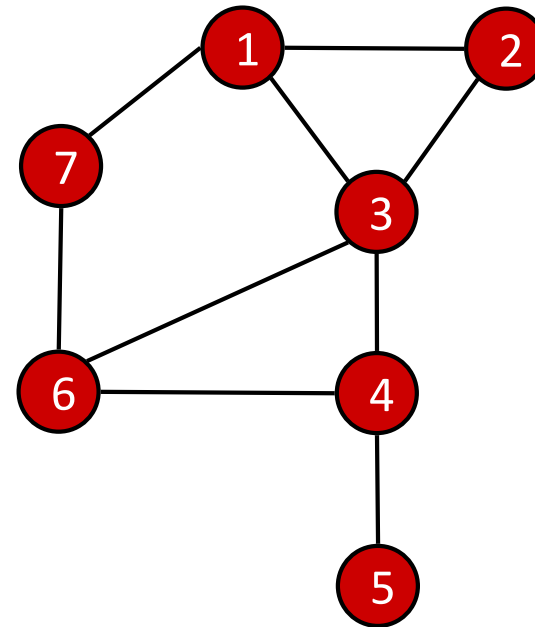


## Grafo orientato

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

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

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



## Grafo non orientato

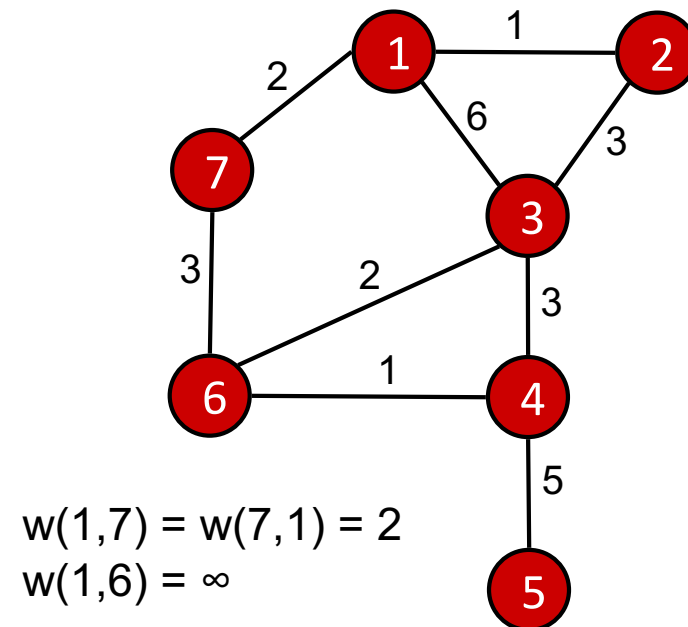
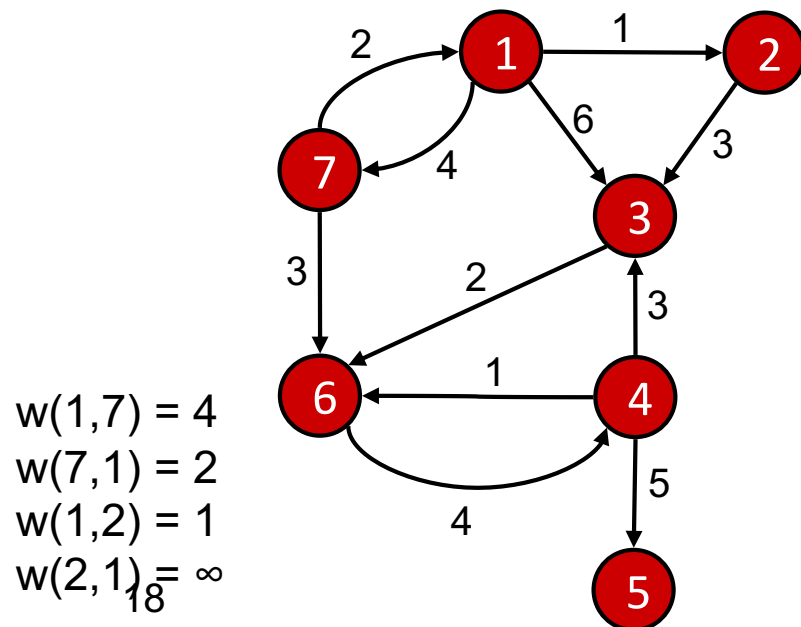
$V = \{1, 2, 3, 4, 5, 6, 7\}$

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

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  $(i,j) \in E$





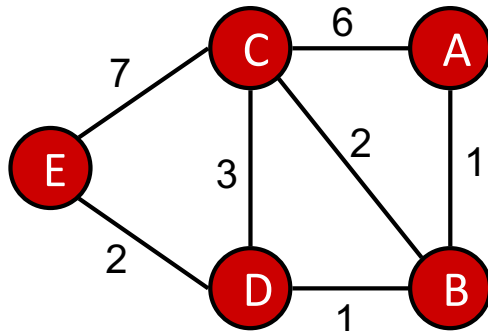
# Routing shortest path nel mondo IP

- Quando i nodi di rete vengono accesi conoscono solamente la configurazione delle loro interfacce
  - Statica
  - Dinamica con 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

# Routing Distance Vector

- 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)
- E' un protocollo semplice e richiede poche risorse
- Problemi:
  - convergenza lenta, partenza lenta (cold start)
  - problemi di stabilità: conteggio all'infinito

# 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            |

Tabella di A

2. A riceve  $DV(C)$

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

Tabella di A

3. B riceve  $DV(D)$

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

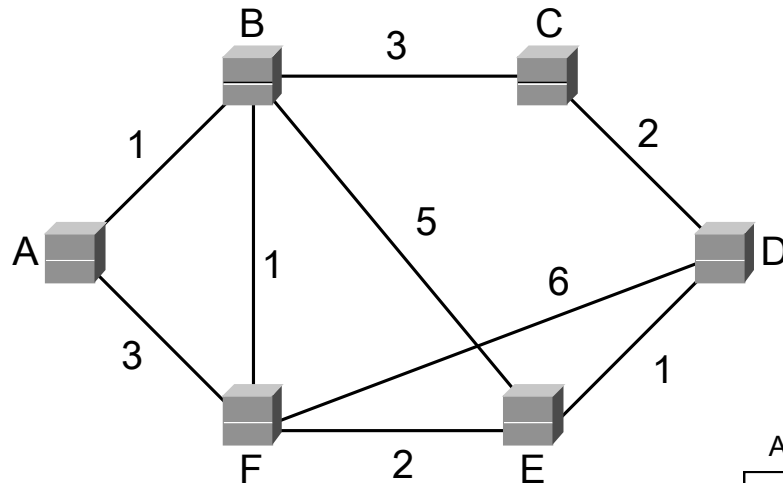
Tabella di B

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

# 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 $T$ |   |  | B - Time $2T$ |   |  | B - Time $3T$ |   |  | B - Time $4T$ |   |  |
|--------------|---|--|---------------|---|--|---------------|---|--|---------------|---|--|
| D            | C |  | D             | C |  | D             | C |  | D             | C |  |
| A            | 1 |  | A             | 1 |  | A             | 1 |  | A             | 1 |  |
| B            | - |  | B             | - |  | B             | - |  | B             | - |  |
| C            | 3 |  | C             | 3 |  | C             | 3 |  | C             | 3 |  |
| D            | 5 |  | D             | 5 |  | D             | 4 |  | D             | 5 |  |
| E            | 3 |  | E             | 3 |  | E             | 3 |  | E             | 4 |  |
| F            | 1 |  | F             | 1 |  | F             | 1 |  | F             | 1 |  |

| A - Time $T$ |   |    | A - Time $2T$ |   |    | A - Time $3T$ |   |    | A - Time $4T$ |   |    | A - Time $5T$ |   |    |
|--------------|---|----|---------------|---|----|---------------|---|----|---------------|---|----|---------------|---|----|
| D            | C | NH | D             | C | NH | D             | C | NH | D             | C | NH | D             | C | NH |
| A            | - |    | A             | - |    | A             | - |    | A             | - |    | A             | - |    |
| B            | 1 | B  | B             | 1 | B  | B             | 1 | B  | B             | 1 | B  | B             | 1 | B  |
| C            |   |    | C             | 4 | B  | C             | 4 | B  | C             | 4 | B  | C             | 4 | B  |
| D            |   |    | D             | 9 | F  | D             | 6 | F  | D             | 5 | B  | D             | 6 | B  |
| E            |   |    | E             | 5 | F  | E             | 4 | B  | E             | 4 | B  | E             | 5 | B  |
| F            | 3 | F  | F             | 2 | B  | F             | 2 | B  | F             | 2 | B  | F             | 2 | B  |

| F - Time $T$ |   | F - Time $2T$ |   | F - Time $3T$ |   | F - Time $4T$ |   |
|--------------|---|---------------|---|---------------|---|---------------|---|
| D            | C | D             | C | D             | C | D             | C |
| A            | 3 | A             | 2 | A             | 2 | A             | 2 |
| B            | 1 | B             | 1 | B             | 1 | B             | 1 |
| C            |   | C             | 4 | C             | 4 | C             | 4 |
| D            | 6 | D             | 3 | D             | 6 | D             | 6 |
| E            | 2 | E             | 2 | E             | 3 | E             | 3 |
| F            | - | F             | - | F             | - | F             | - |

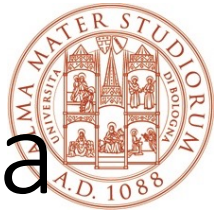
# Algoritmo

- Nodo sorgente del traffico denominato  $s$
- $D_i^h$  costo del percorso di lunghezza minima da  $s$  a  $j$  in al più  $h$  salti
- $d_{ij}$  costo del collegamento diretto fra  $i$  e  $j$
- $d_{ij} = \infty$  se  $i$  e  $j$  non sono connessi direttamente

- Per  $h=1$

$$D_j^h = d_{sj} \quad \forall j \neq s$$

- Per  $h = h+1$   $D_j^h = \min_i \{ D_i^{h-1} + d_{ij}, D_j^{h-1} \}$



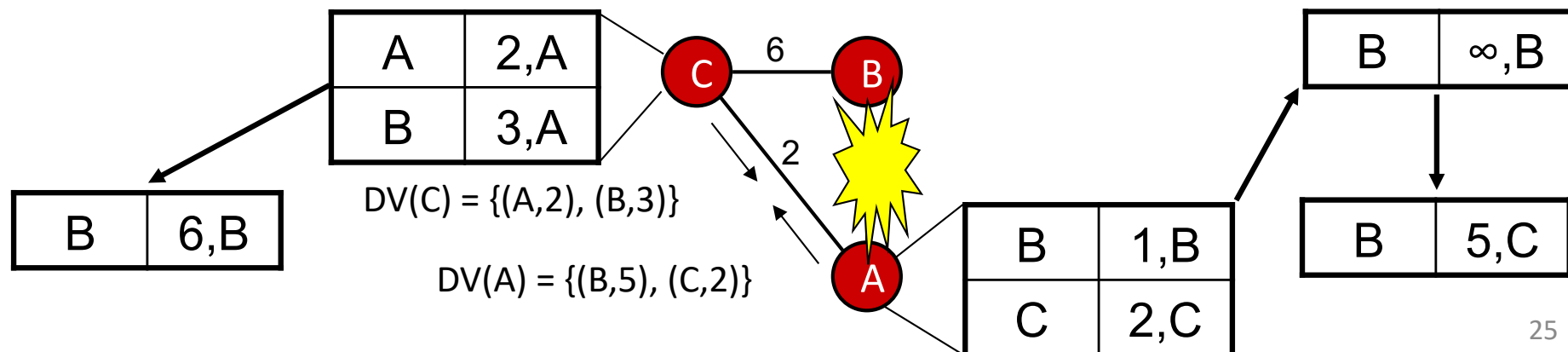
# Cold start e tempo di convergenza

- 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
- 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

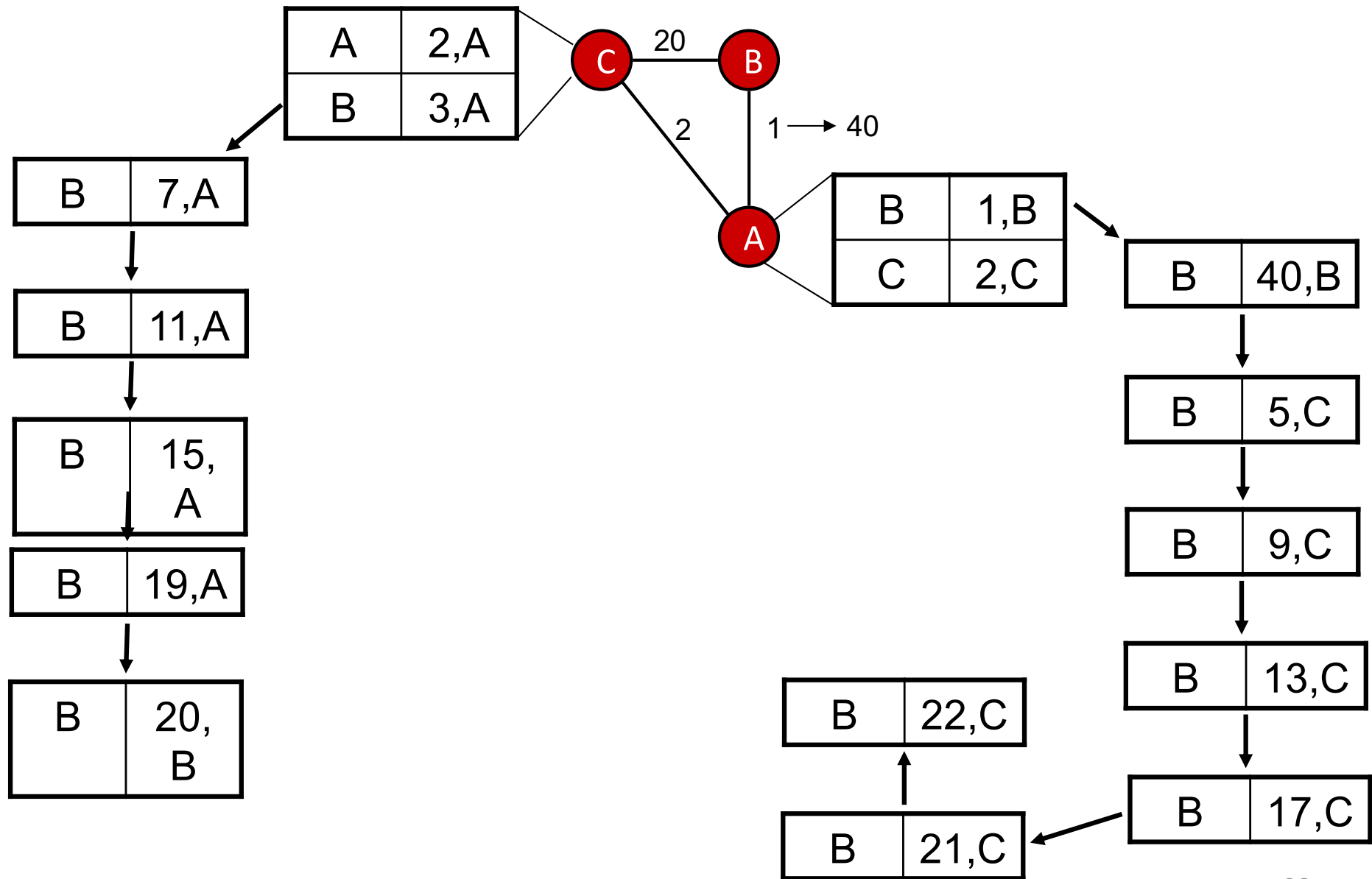


# Bouncing effect

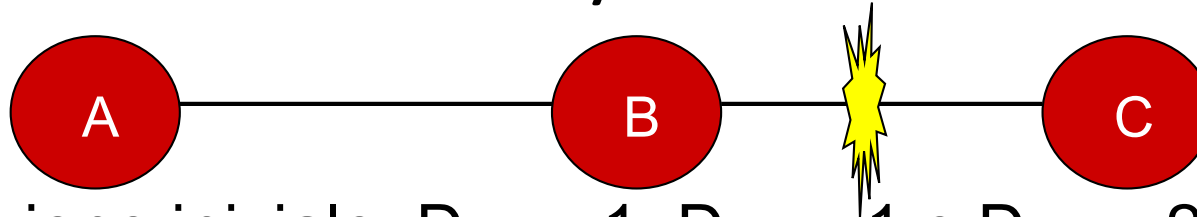
- Il link fra due nodi A e B cade
  - A e B si accorgono che il collegamento non funziona e immediatamente pongono ad infinito la sua lunghezza
  - Se altri nodi hanno nel frattempo inviato anche i loro vettori delle distanze, si possono creare delle incongruenze temporanee, di durata dipendente dalla complessità della rete
    - ad esempio A crede di poter raggiungere B tramite un altro nodo C che a sua volta passa attraverso A
  - 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



# Convergenza lenta



# Count to infinity



- Situazione iniziale:  $D_{AB} = 1$ ,  $D_{BC} = 1$  e  $D_{AC} = 2$ 
  - Link BC va fuori servizio
  - B riceve il DV di A che contiene l'informazione  $D_{AC} = 2$ , per cui esso computa una nuova  $D'_{BC} = D_{BA} + D_{AC} = 3$
  - B comunica ad A la sua nuova distanza da C
  - A calcola la nuova distanza  $D_{AC} = D_{AB} + D'_{BC} = 4$
  - ...
- 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

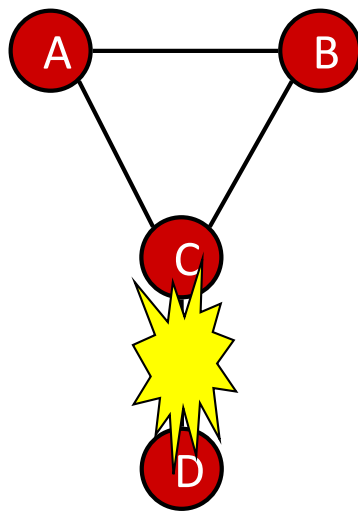
- Split horizon è una tecnica molto semplice per risolvere in parte i problemi suddetti
  - se A instrada i pacchetti verso una destinazione X tramite B, non ha senso per B cercare di raggiungere X tramite A
  - di conseguenza non ha senso che A renda nota a B la sua distanza da X
- Un algoritmo modificato 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
- Triggered update: un nodo deve inviare immediatamente le informazioni a tutti i vicini qualora si verifichi una modifica della propria tabella di instradamento

# Ma non basta...

- 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

- 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

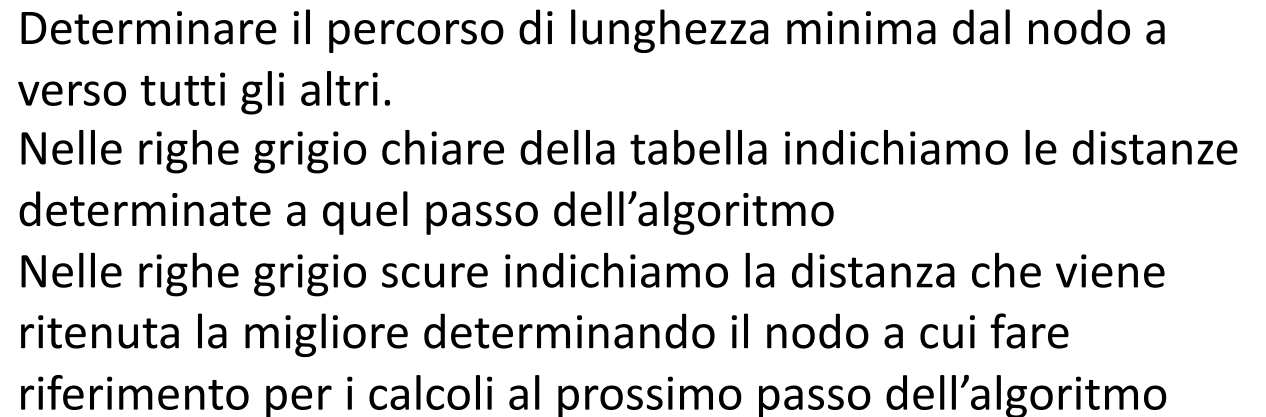
# Raccolta delle informazioni

- Ogni router deve comunicare con i propri vicini ed “imparare” i loro indirizzi
  - **Hello Packet**
- Deve poi misurare la distanza dai vicini
  - **Echo Packet**
- 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



# 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
    - 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

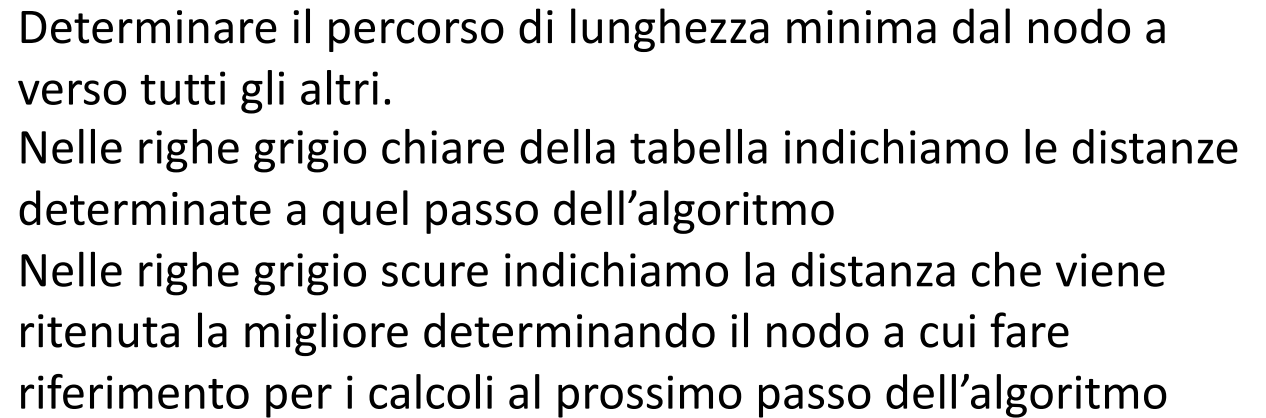


È dimostrato che, procedendo in questo modo, le distanze determinate sono minime e non possono essere migliorate nei passi successivi dell'algoritmo

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)

34



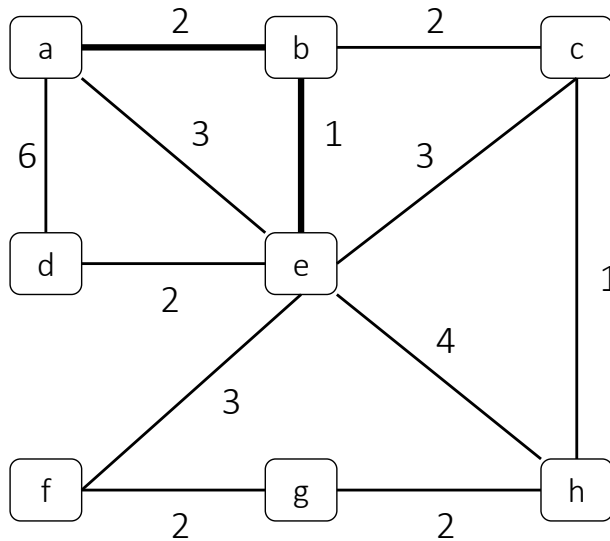
È dimostrato che, procedendo in questo modo, le distanze determinate sono minime e non possono essere migliorate nei passi successivi dell'algoritmo

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)

35

# 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

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

● ● ● ● ● ●

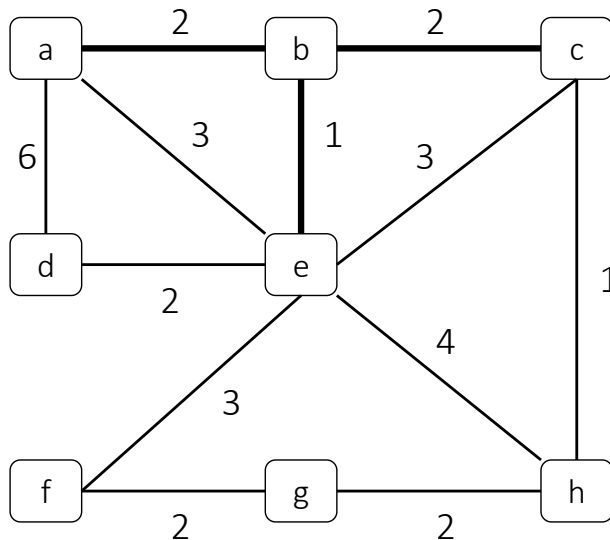
È dimostrato che, procedendo in questo modo, le distanze determinate sono minime e non possono essere migliorate nei passi successivi dell'algoritmo

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)

[illegible]

# 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

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

• • • • •

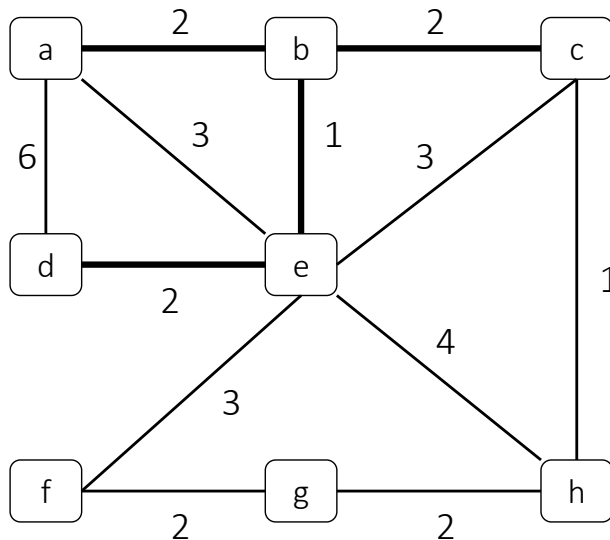
È dimostrato che, procedendo in questo modo, le distanze determinate sono minime e non possono essere migliorate nei passi successivi dell'algoritmo

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 |     |   |     |
|   | A 2 |     |     |     |     |   |     |
|   |     | B 4 |     | B 3 |     |   |     |
|   | A 2 |     |     | B 3 |     |   |     |
|   |     | E 6 | E 5 |     | E 6 |   | E 7 |
|   | A 2 | B 4 |     | B 3 |     |   |     |
|   |     |     |     |     |     |   |     |
|   |     |     |     |     |     |   |     |
|   |     |     |     |     |     |   |     |
|   |     |     |     |     |     |   |     |
|   |     |     |     |     |     |   |     |
|   |     |     |     |     |     |   |     |
|   |     |     |     |     |     |   |     |
|   |     |     |     |     |     |   |     |
|   |     |     |     |     |     |   |     |
|   |     |     |     |     |     |   |     |

# 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

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

• • • • •

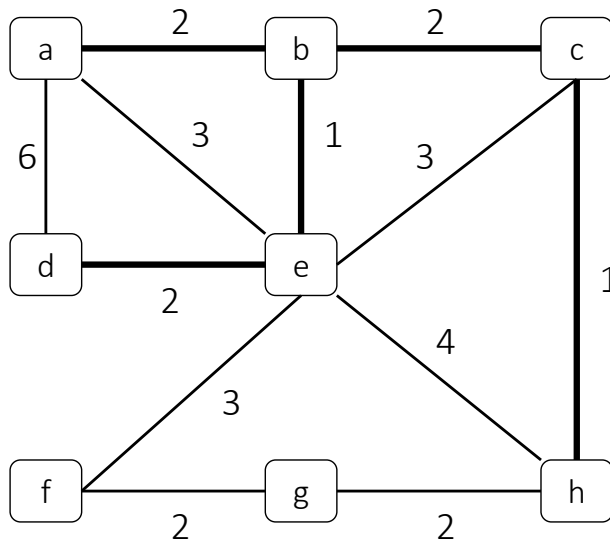
È dimostrato che, procedendo in questo modo, le distanze determinate sono minime e non possono essere migliorate nei passi successivi dell'algoritmo

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)

[illegible]

# 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

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

• • • • •

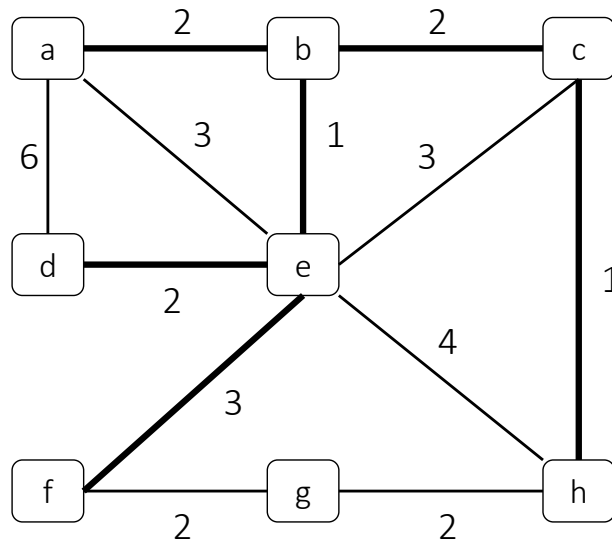
È dimostrato che, procedendo in questo modo, le distanze determinate sono minime e non possono essere migliorate nei passi successivi dell'algoritmo

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)

[illegible]

# 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

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

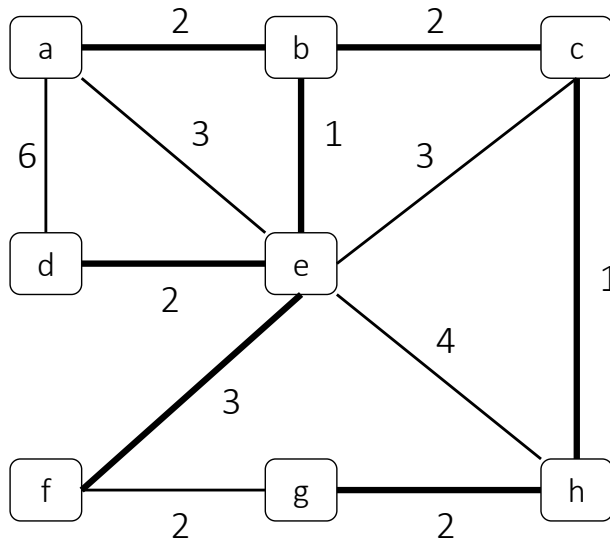
Nella riga gialla viene riassunta la soluzione dell'algoritmo. Essa riporta la distanza minima verso X ed il nodo "predecessore" di X sul relativo percorso

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 3</b> |            |     |            |
|   |            | E 6        | E 5        |            | E 6        |     | E 7        |
|   | <b>A 2</b> | <b>B 4</b> |            | <b>B 3</b> |            |     |            |
|   |            |            |            |            |            |     | C 5        |
|   | <b>A 2</b> | <b>B 4</b> | <b>E 5</b> | <b>B 3</b> |            |     |            |
|   |            |            |            |            |            |     |            |
|   | <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> |
|   |            |            |            |            |            |     |            |
|   |            |            |            |            |            |     |            |
|   |            |            |            |            |            |     |            |
|   |            |            |            |            |            |     |            |
|   |            |            |            |            |            |     |            |
|   |            |            |            |            |            |     |            |
|   |            |            |            |            |            |     |            |
|   |            |            |            |            |            |     |            |



# 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

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

.....

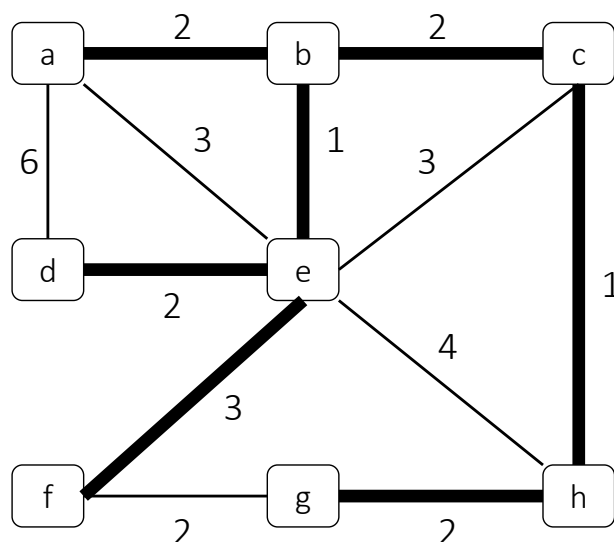
È dimostrato che, procedendo in questo modo, le distanze determinate sono minime e non possono essere migliorate nei passi successivi dell'algoritmo

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 3</b> |            |            |            |
|   |            | E 6        | E 5        |            | E 6        |            | E 7        |
|   | <b>A 2</b> | <b>B 4</b> |            | <b>B 3</b> |            |            |            |
|   |            |            |            |            |            |            | C 5        |
|   | <b>A 2</b> | <b>B 4</b> | <b>E 5</b> | <b>B 3</b> |            |            |            |
|   |            |            |            |            |            |            |            |
|   | <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



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

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 |     |     |     |
|   | A 2 |     |     |     |     |     |     |
|   |     | B 4 |     | B 3 |     |     |     |
|   | A 2 |     |     | B 3 |     |     |     |
|   |     | E 6 | E 5 |     | E 6 |     | E 7 |
|   | A 2 | B 4 |     | B 3 |     |     |     |
|   |     |     |     |     |     |     | C 5 |
|   | A 2 | B 4 | E 5 | B 3 |     |     |     |
|   |     |     |     |     |     |     |     |
|   | A 2 | B 4 | E 5 | B 3 |     |     | C 5 |
|   |     |     |     |     |     | H 7 |     |
|   | A 2 | B 4 | E 5 | B 3 | E 6 |     | C 5 |
|   |     |     |     |     |     | 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 | B 5 | B 3 | B 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

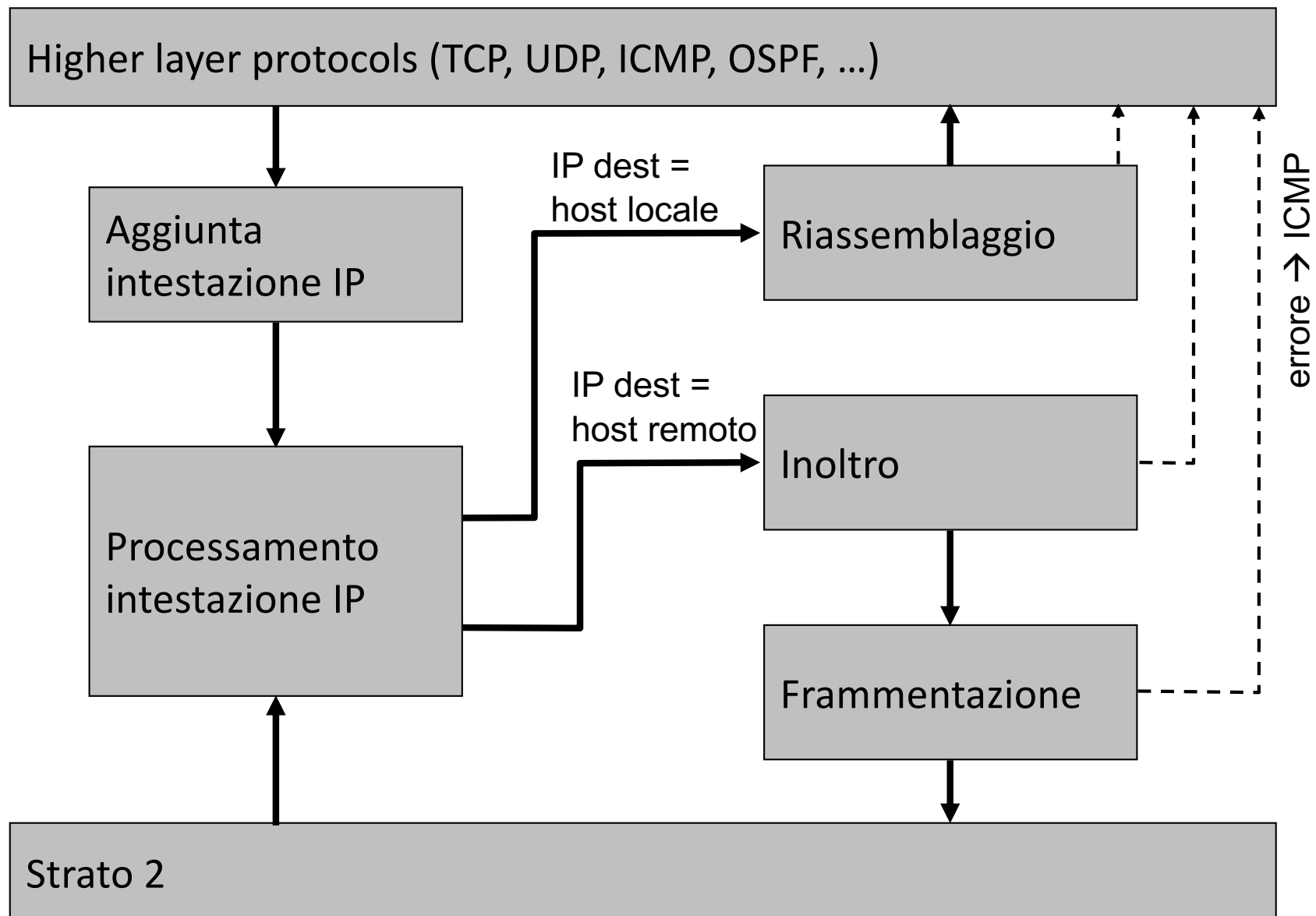
- SOHO (Small Office and HOMe) router
  - Utilizzo domestico o piccoli uffici
  - Interfaccia sulla LAN (switch con poche porte Fast Ethernet 100Mbit/s e wi-fi)
- Router di accesso
  - used by ISPs to provide access service
  - large number of medium-low speed ports (50 kbps ÷ 10 Mbps)
  - capable of several protocols and access technologies (PPP, SLIP, ADSL, FTTx, ...)
- Enterprise/campus router
  - Interconnessione fra LAN per organizzazioni di medie dimensioni
  - Poche porte ad elevata velocità (Fast o Gigabit Ethernet)
- Backbone router
  - Per reti di trasporto e connessioni inter-domain
  - Piccolo numero di porte ad elevata velocità ( $\geq 1\text{Gbps}$ )
  - Equipaggiato con sistemi di garanzia dell'affidabilità (ridondanza, monitoraggio remoto, ecc.)

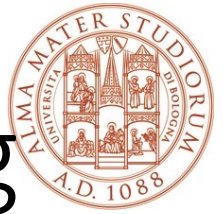


# Le 4 funzioni dei Router

- Routing
  - Scambio di informazioni con altri router (IGP/EGP)
  - Elaborazione locale (routing algorithm)
  - Popolazione delle tabelle di routing
- Forwarding
  - IP
    - Table lookup
    - Header update
- Switching
  - Trasferimento del datagramma da interfaccia di input a interfaccia di output
- Trasmissione
  - Trasmissione del datagramma sul mezzo fisico (utilizzando l'interfaccia di rete di output)

# Schema funzionale di un router

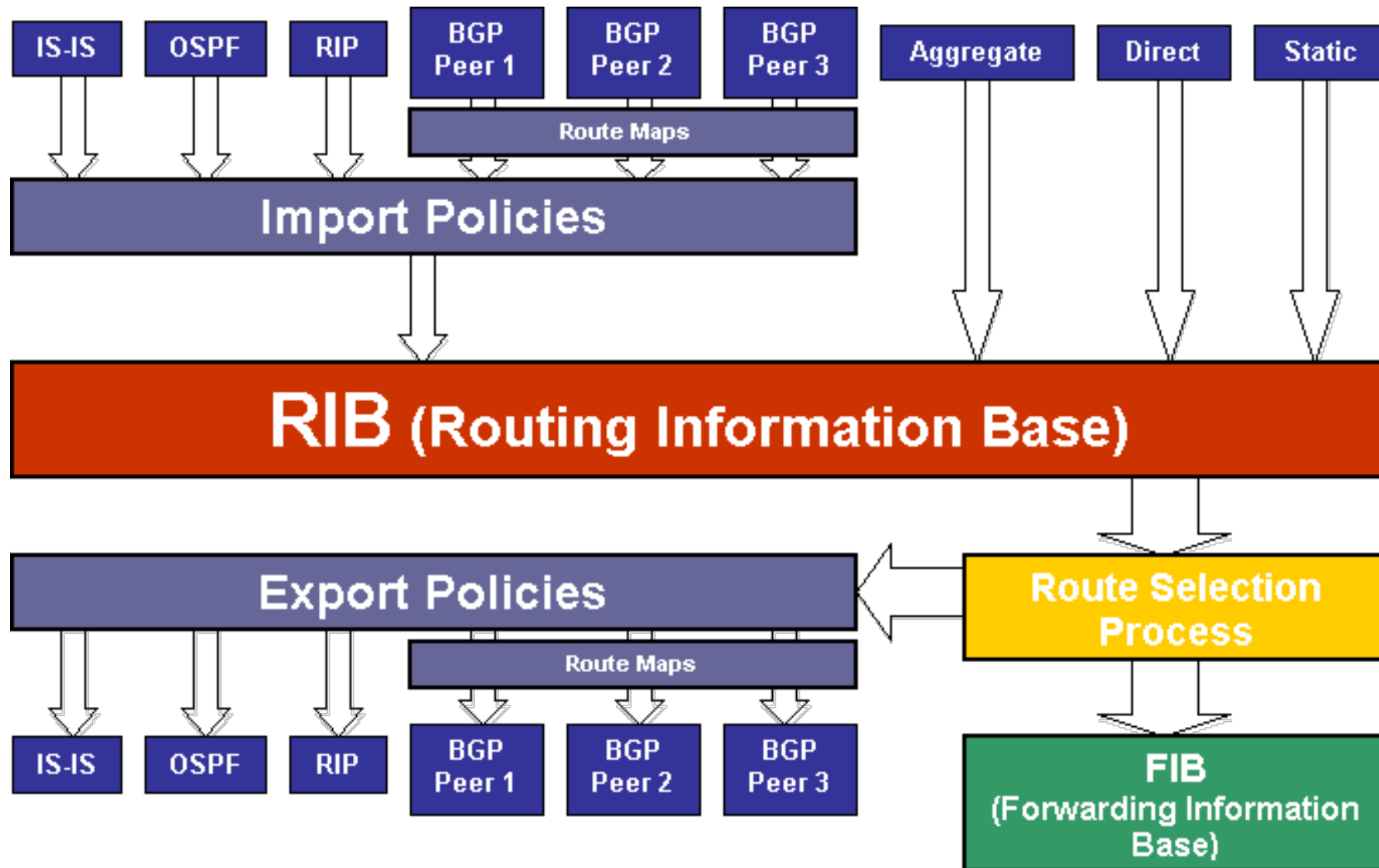




# Tabella di Routing e di forwarding

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

# 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

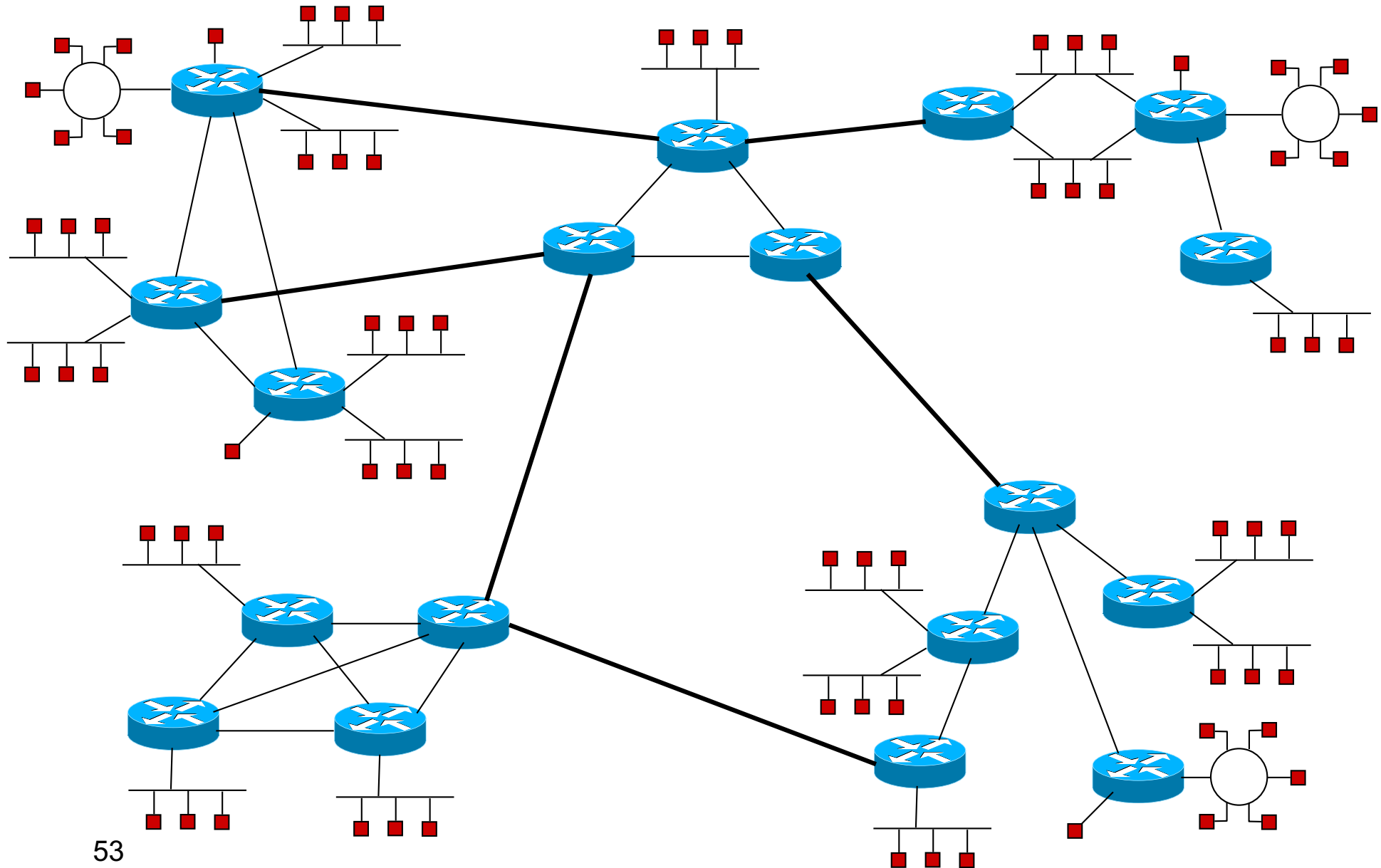
- In Internet si usa il routing gerarchico e le aree di routing sono chiamate **Autonomous System** (AS)
  - un AS può essere ulteriormente suddiviso in porzioni dette **Routing Area** (RA) interconnesse da un **backbone** (dorsale)
  - ogni network IP è tutta contenuta in un AS o in una RA
    - tradizionalmente 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
- 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)



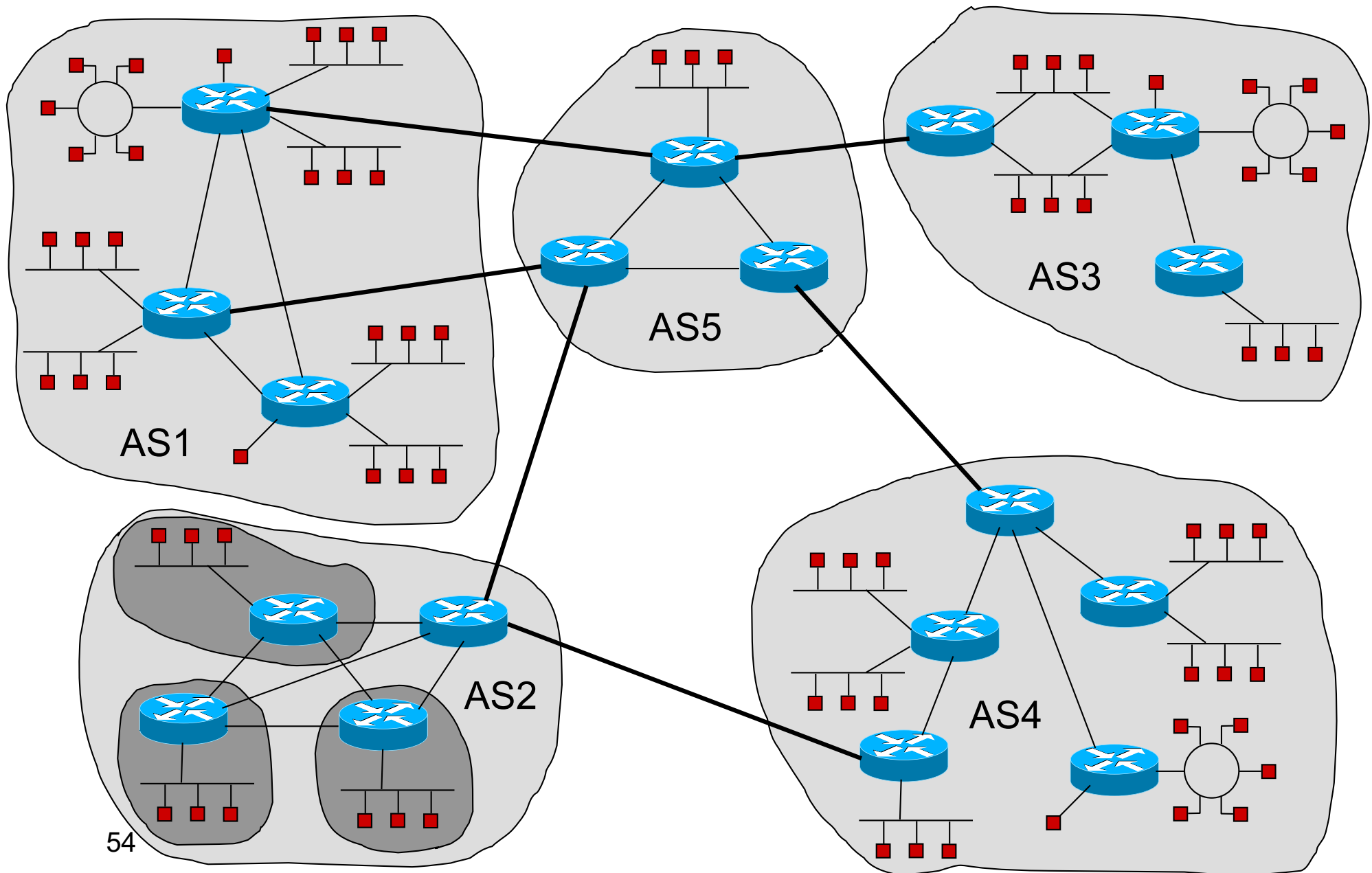
ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

# Autonomous Systems and peering

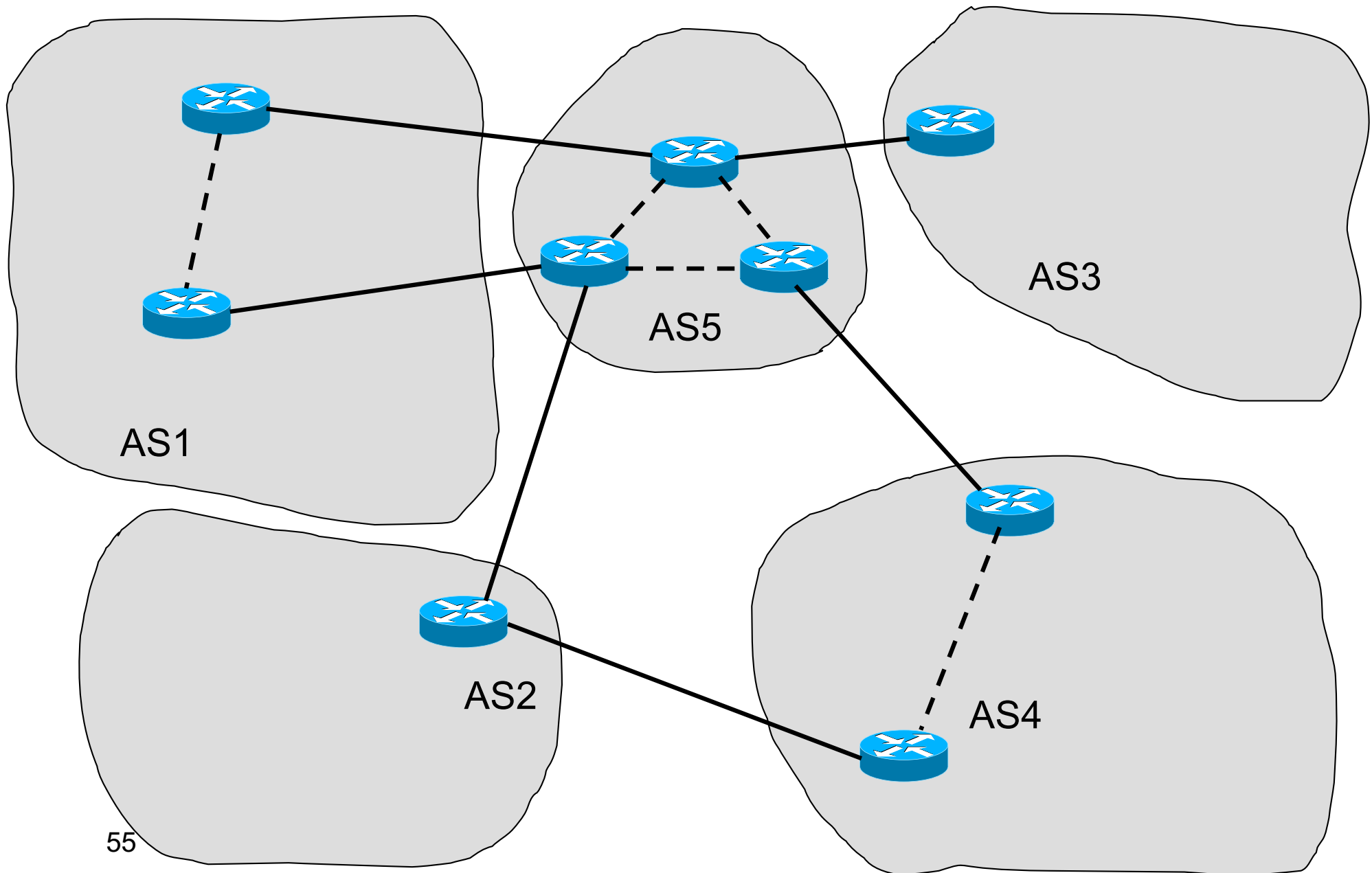
# Internet = rete di reti



# Internet = sistemi interconnessi



# Internet: grafo semplificato





# Il routing a livello globale

- 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
  - 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





# Autonomous Systems

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

## Autonomous Systems (AS)

- 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



# I protocolli di routing

- 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
  - OSPF: Open Shortest Path First
- Exterior Gateway Protocol
  - EGP: Exterior Gateway Protocol
  - BGP: Border Gateway Protocol



# 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




# 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:  [Query](#) [Advanced Query](#) [Query Help](#)

[Register Now](#) [Features](#) [Support](#) [FAQ](#) [Contact Us](#) [Log In](#)

Advanced Query

Query the RADb:  [Query](#) [Advanced Options](#) [Query Help](#)

```
aut-num: AS137
as-name: ASGARR
descr: Consortium GARR
org: ORG-GIRa1-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: *****
```

# AS 137



## Regole di Import:

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

```
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:          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 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
```

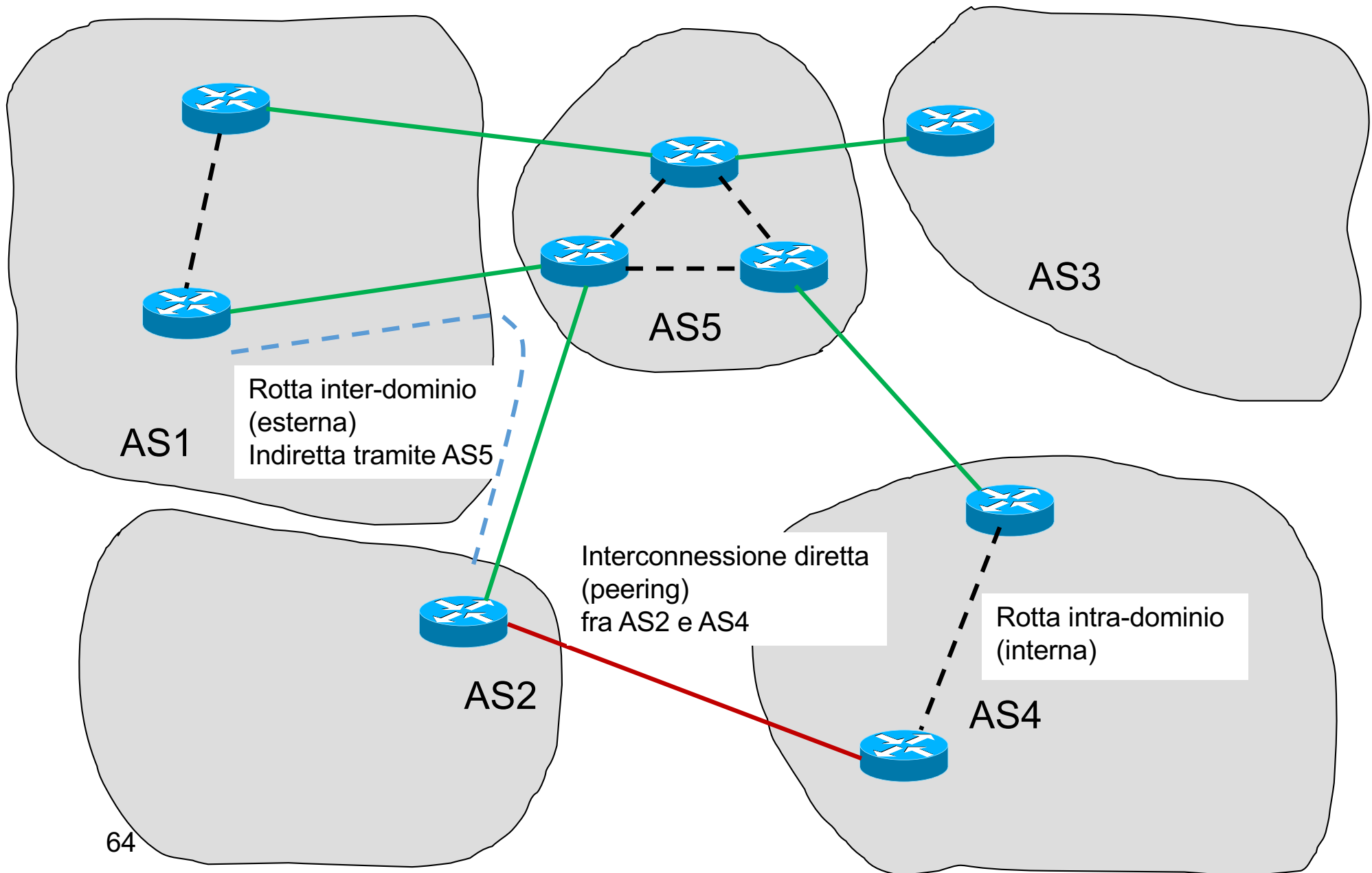
GEANT è la rete degli enti di ricerca  
Europea

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

# 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
  - ...
- Tipicamente un ISP si registra come AS



# 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

- 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

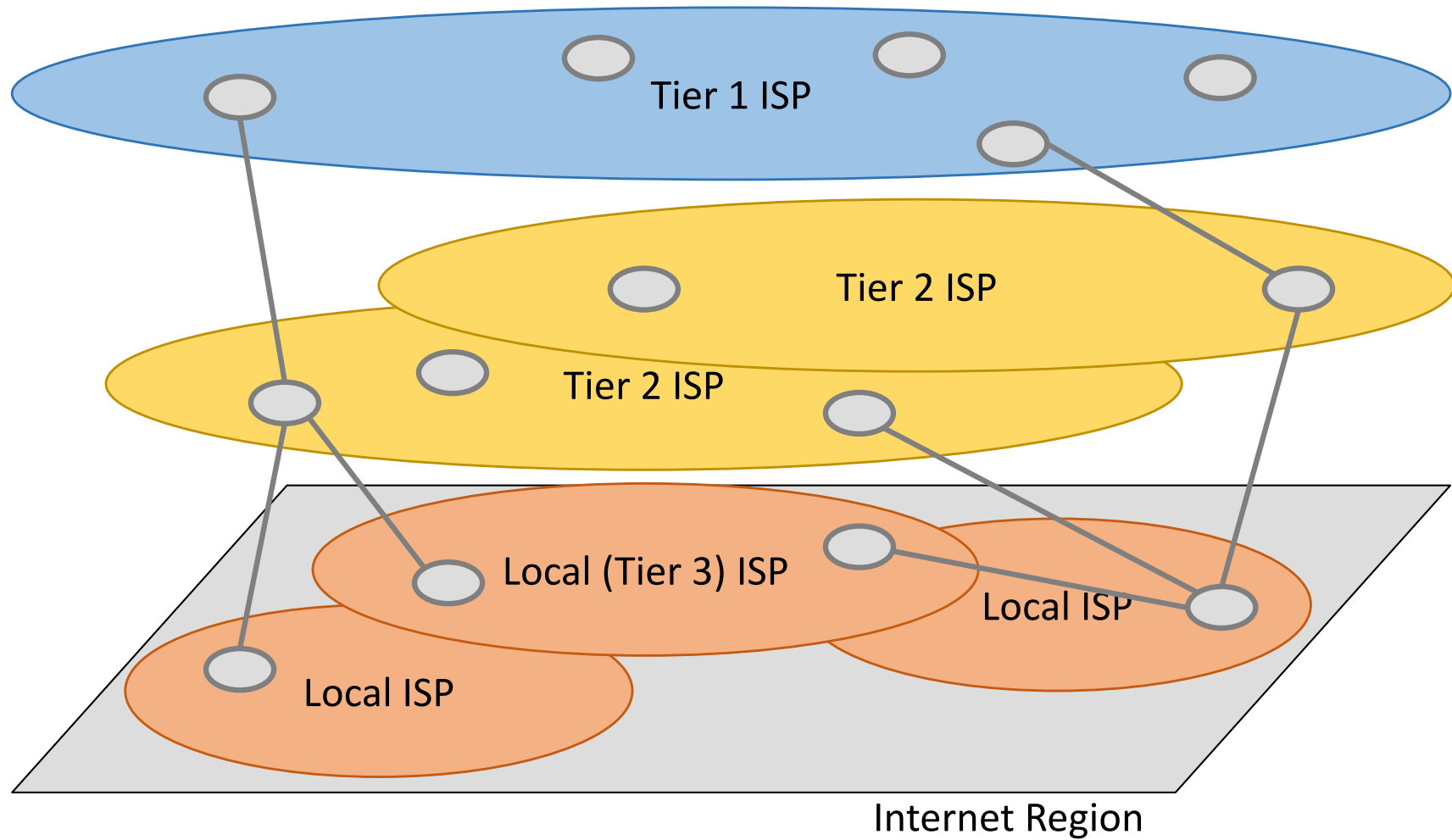
- Tier 2

- 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 stesse 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

- 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

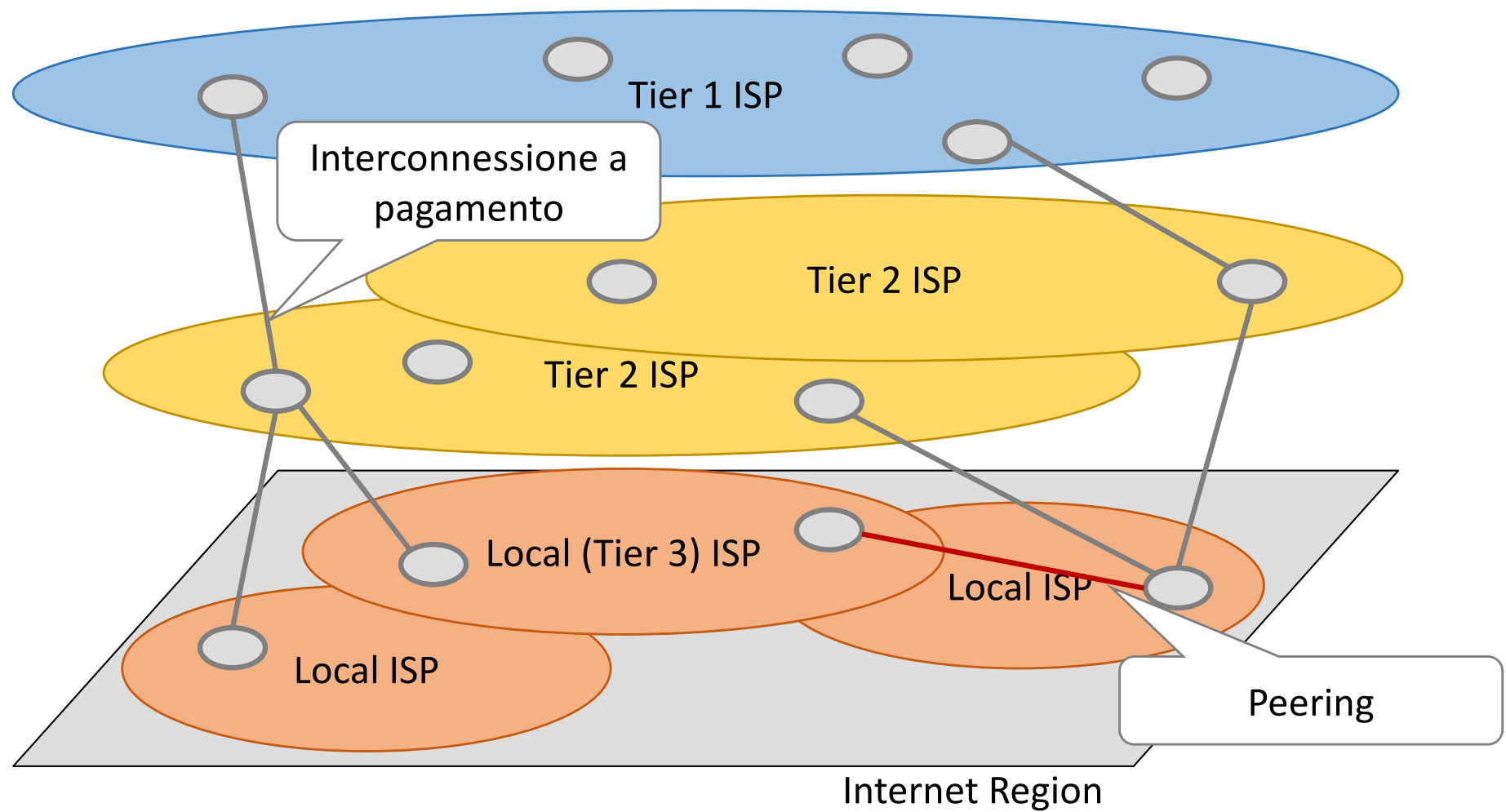


# Peering policy

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



# In sintesi

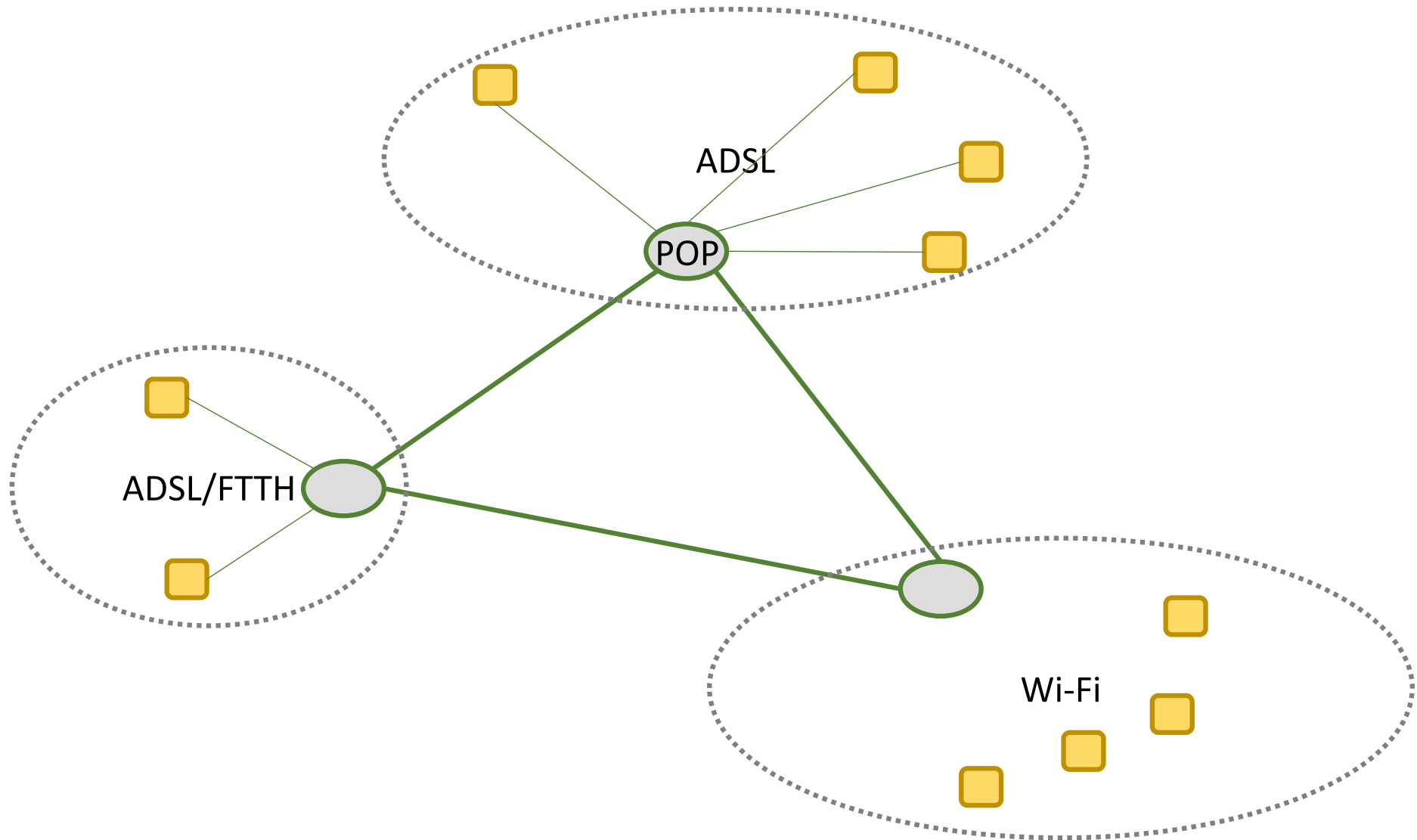




# ISP locali e POP

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

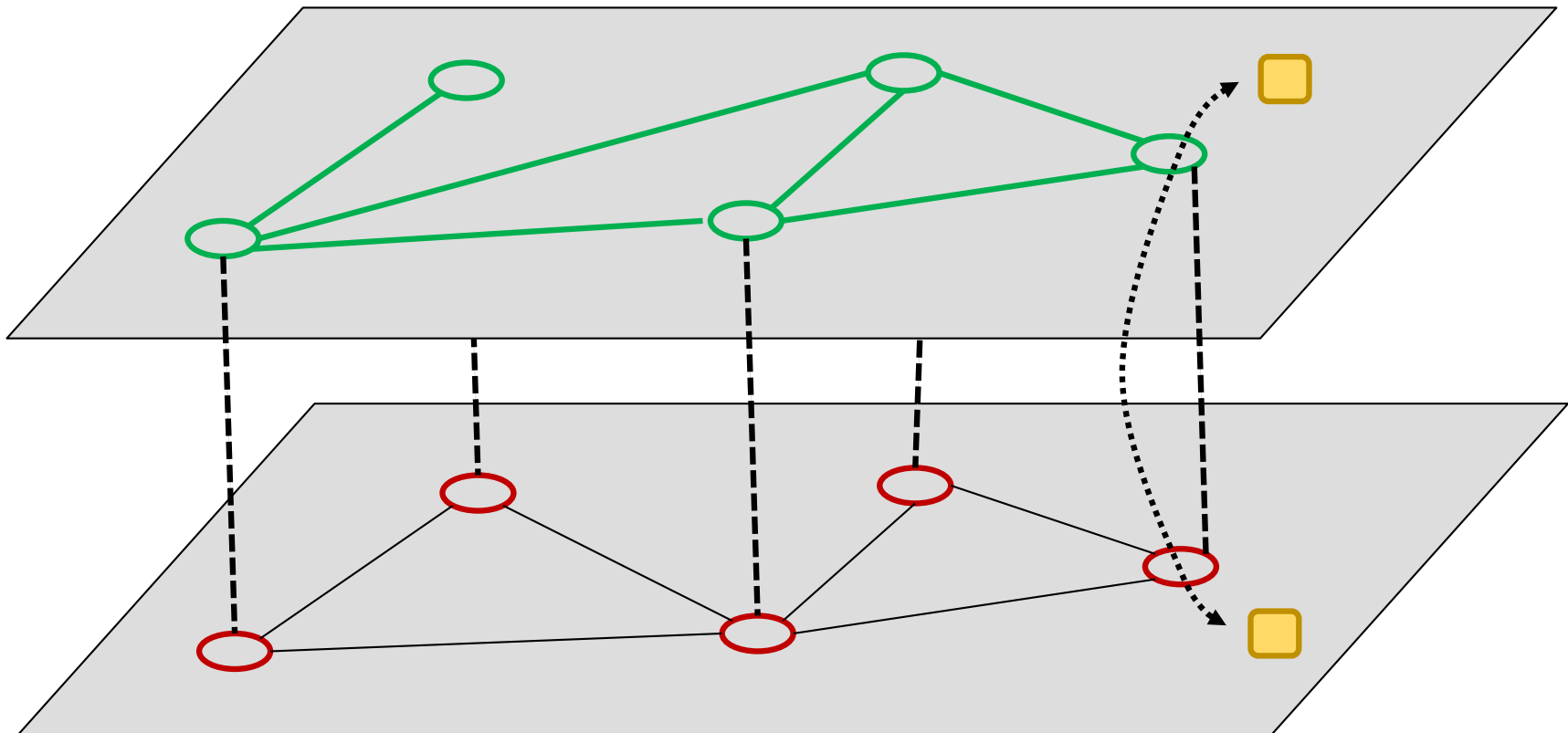
- Un ISP dispone di un sottoinsieme di numeri IP da utilizzare per i suoi clienti
  - Se sono consecutivi possono avere lo stesso prefisso quindi unico Network ID
  - Se non sono consecutivi deve gestire più prefissi quindi più Network ID
- In funzione della dimensione (numero di utenti e distanze geografiche)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



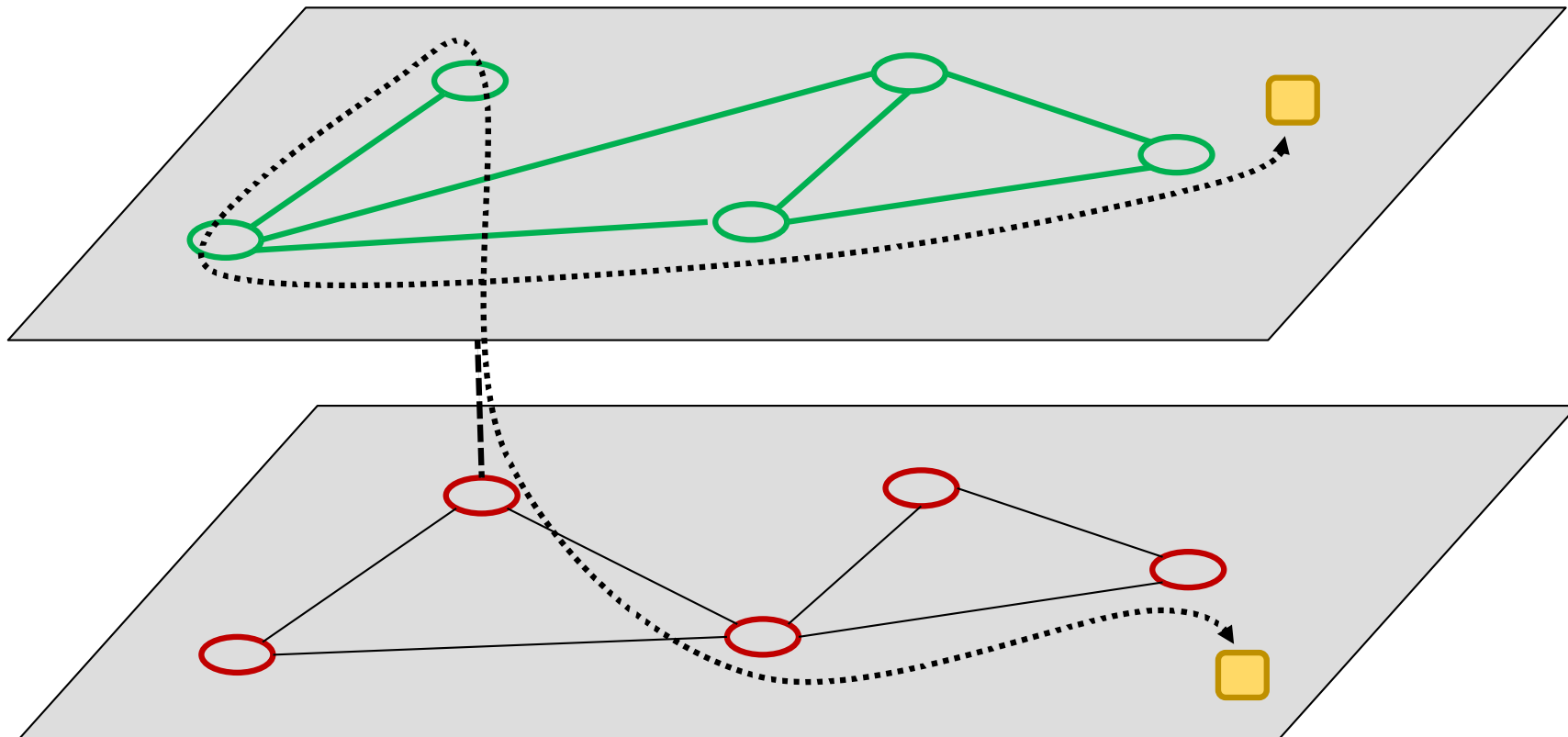
# Interconnessione

- Come scambiano traffico ISP che coprono la medesima zona geografica
- Interconnettere fra loro tutti i POP?
  - Numerosi collegamenti
  - Complessità di gestione del routing
    - Rotte specifiche per ogni POP in funzione dei numeri a loro connessi
  - Percorsi di lunghezza minima
- Interconnettere uno o pochi POP?
  - Minor numero di collegamenti
  - Routing semplificato
  - Percorsi potenzialmente più lunghi

# 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
- 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





# Internet Exchange

- 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
  - 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)



# IXP in Italia

- 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)

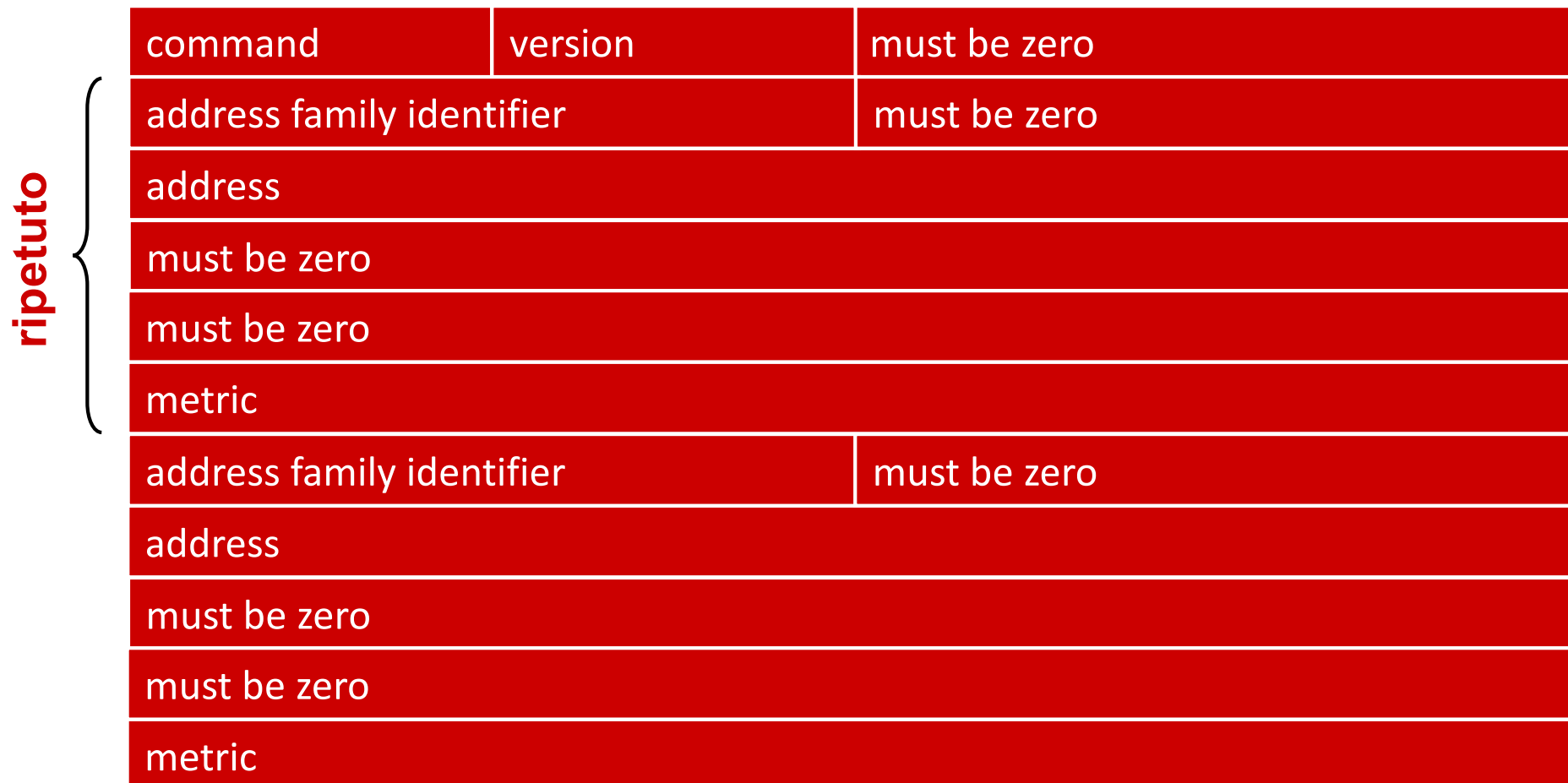
- Protocollo **distance vector**, di implementazione vecchia (RFC 1058, Giugno 1988), discende dal protocollo di routing realizzato per la rete XNS di Xerox
- Ne esiste una **versione 2** più recente (RFC 2453)
- Molto diffuso in passato perché il codice di implementazione è liberamente disponibile
- 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)
  - **RESPONSE** serve in generale per inviare informazioni di routing (cioè i distance vector)
- I messaggi RIP sono trasportati da UDP ed usano la porta 520 sia in trasmissione che in ricezione

# Response

- 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
- Response contiene il distance vector del router che lo invia
  - Destinazione
  - Distanza (hop count)

# RIP: formato dei pacchetti

- La struttura del pacchetto è basata su parole di 32 bit
- Il pacchetto può avere lunghezza variabile fino a 512 byte (max 25 entry)



# 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)
  - inizialmente pensati per adattarsi ad altri protocolli
- **command**: distingue tra REQUEST (1) e RESPONSE (2)
- **version**: versione del RIP
- **address family identifier**: indica il tipo di indirizzo di rete utilizzato, vale 2 per IP
- **address**: identifica la destinazione per la quale viene data la distanza
- **metrica**: è la distanza dalla destinazione indicata

# 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)
    - 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



# 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  $d_i < \infty$
  - 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

# RIP: problematiche

- 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



# 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

10.1.1.0-----router----200.200.200.0----router---100.100.100.0---router---10.2.2.0

- 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 appartenenti 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

# RIP versione 2

- I miglioramenti introdotti riguardano soprattutto:
  - subnetting e CIDR
  - autenticazione

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

# RIP versione 2

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

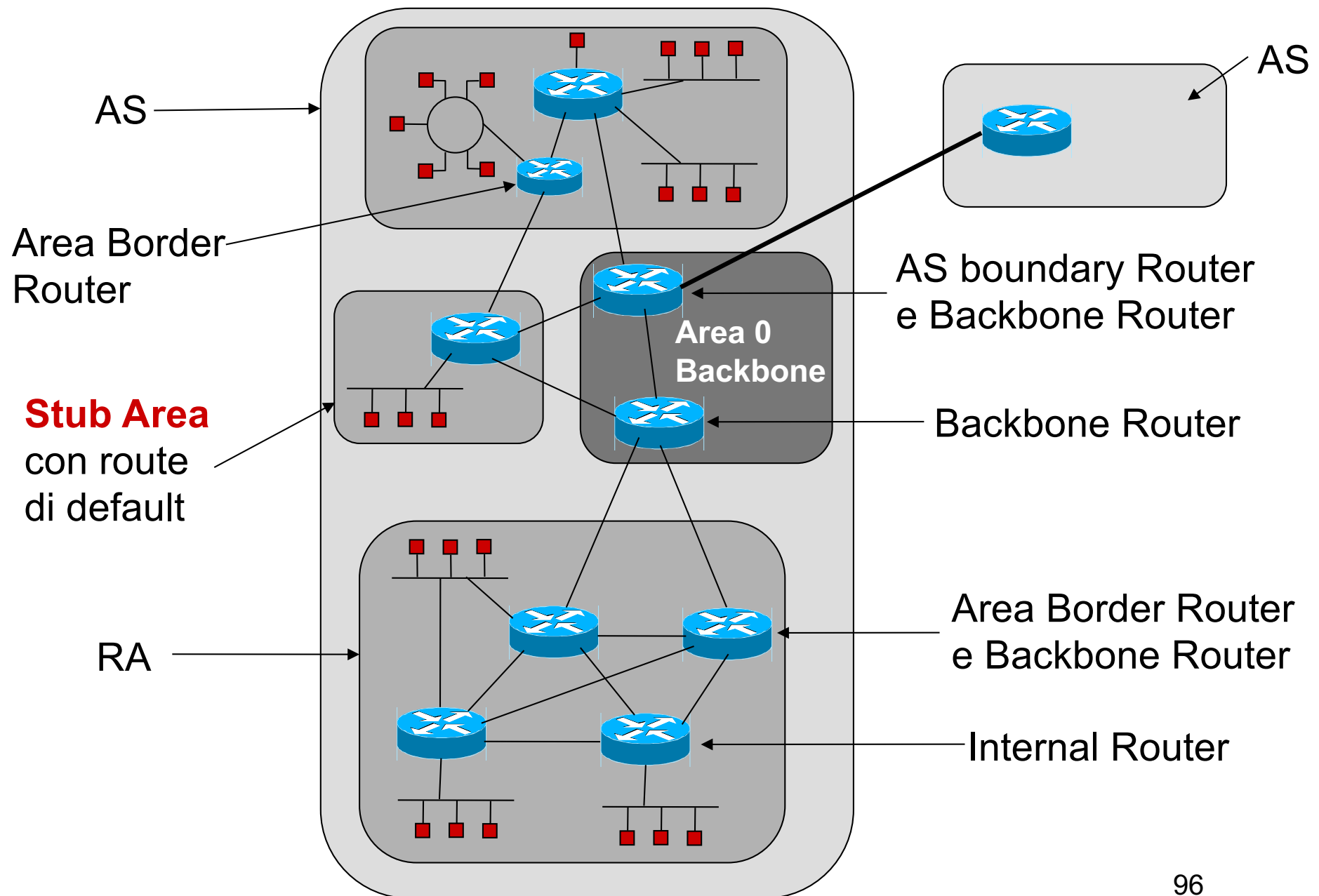
# Open Shortest Path First (OSPF)

- Divenuto standard nella versione 2 (RFC 2328)
- Oggi è il più diffuso IGP
- Protocollo di tipo **link state**
  - invio di **Link State Advertisement** (LSA) a tutti gli altri router
- Incapsulato direttamente in IP
  - 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: aree di routing

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

# OSPF: aree di routing e tipologie di router





# Tipi di route



- Route intra-area
  - aggiornamento delle informazioni di routing pertinenti all'area
- Route inter-area
  - Aggiornamento delle informazioni di routing pertinenti ad aree diverse da quella considerata
- Route esterni
  - Aggiornamenti delle informazioni di route provenienti da altri protocolli al di fuori del dominio OSPF
  - Inoltrati nel dominio OSPF dal ASBR



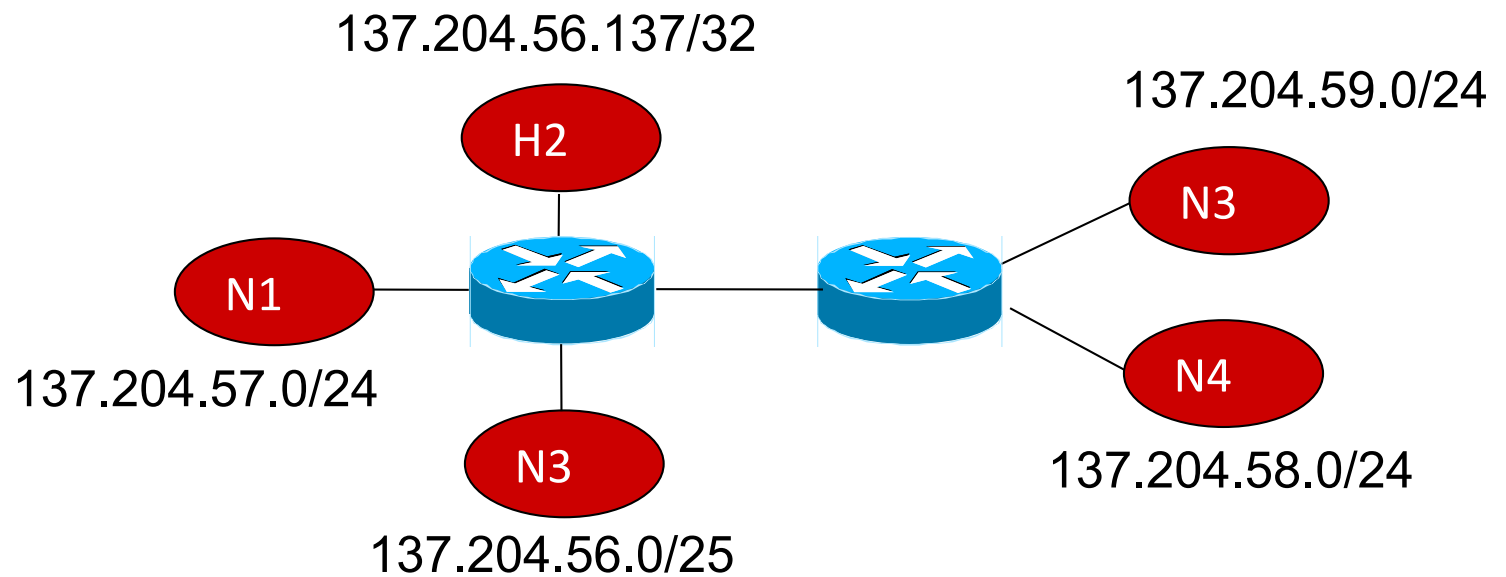
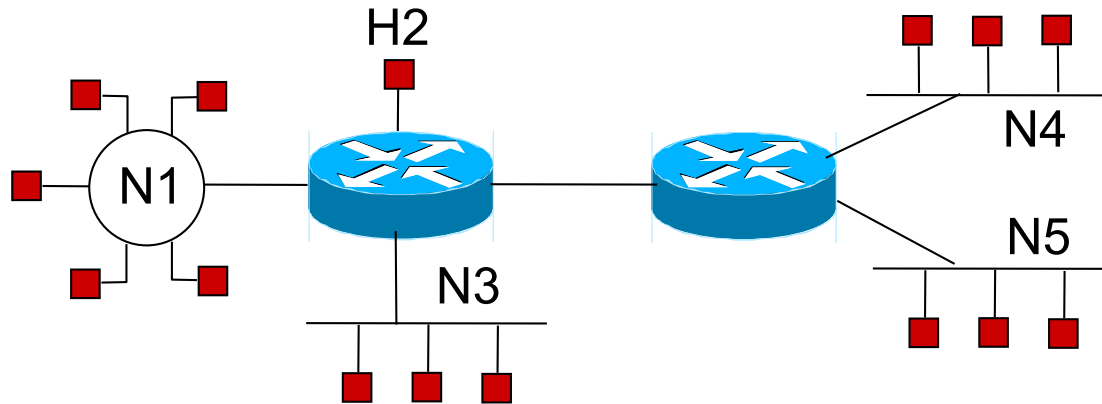
# Tipi di aree

- Area normale
  - 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

# OSPF: ulteriori caratteristiche

- **Bilanciamento del carico:** se un router ha più percorsi di uguale lunghezza verso una certa destinazione, il carico viene ripartito equamente su di essi
- **Autenticazione:** per garantire maggiore sicurezza nello scambio delle informazioni di routing è prevista autenticazione con password ed uso di crittografia
- **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

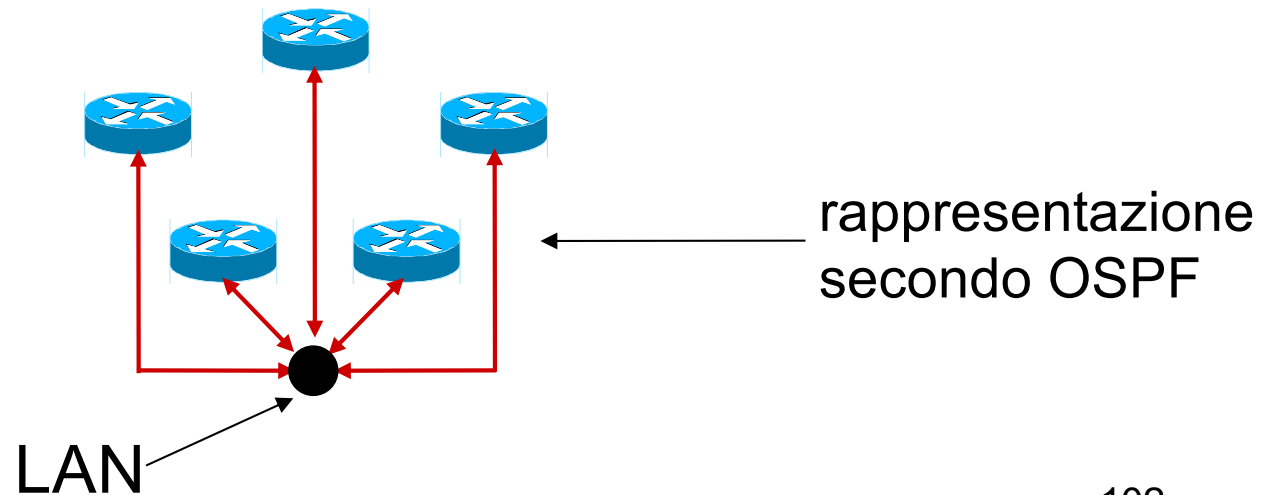
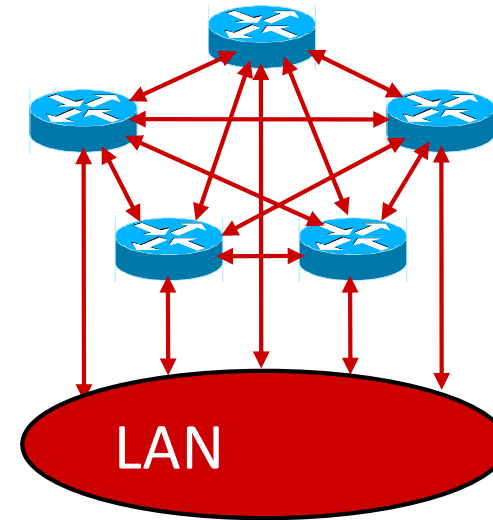
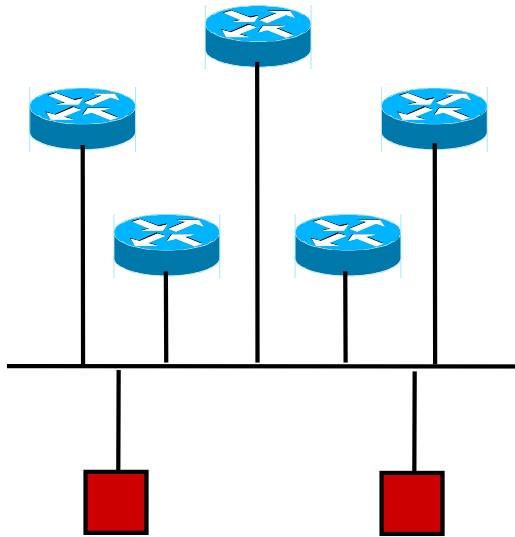
# OSPF: rappresentazione di host e router



# OSPF: tipologie di rete

- OSPF è progettato per operare correttamente con reti:
  - **Point-to-Point**
  - **Broadcast Multi-Access** (BMA: LAN 802)
  - **Non-Broadcast Multi-Access** (NBMA: X.25, ATM, Frame Relay)
- 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 alla 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

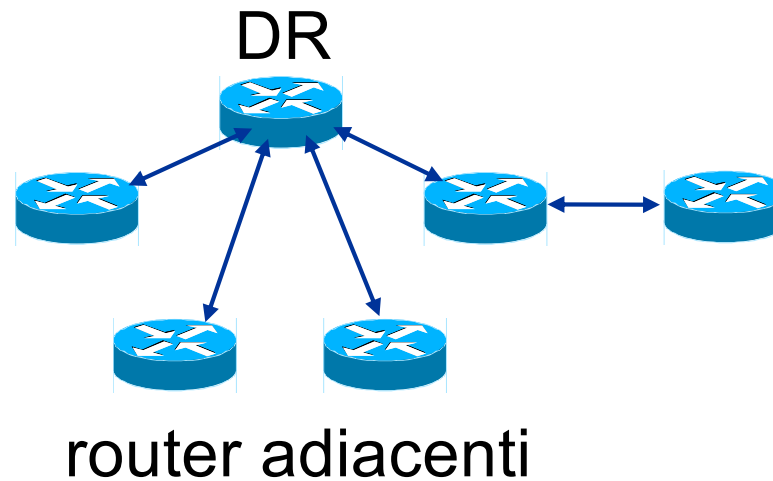
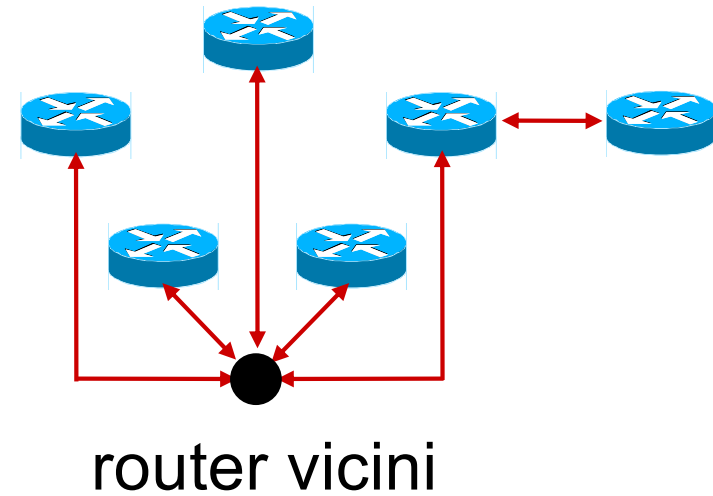
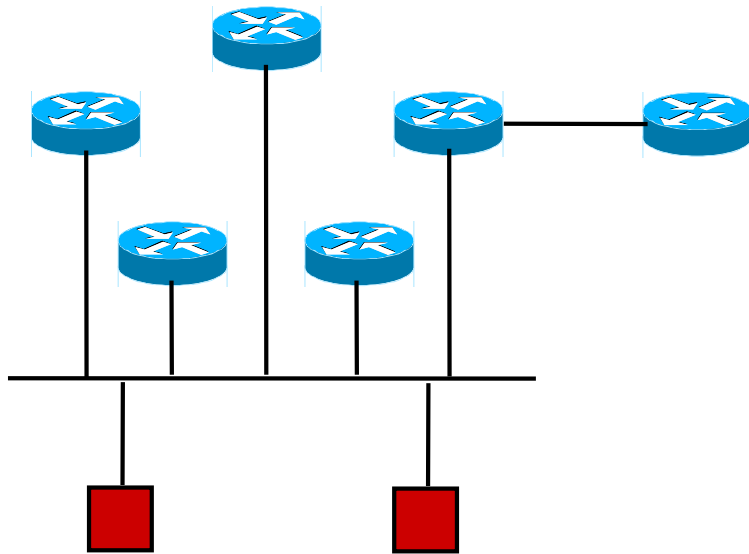
# OSPF: rappresentazione di reti multi-accesso



# OSPF: vicinanza e adiacenza tra router

- **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
- 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
  - Per ragioni di affidabilità occorre avere anche un **Backup Designated Router** (BDR) adiacente a tutti i router locali

# OSPF: vicinanza e adiacenza tra router





# OSPF: identificazione di router e priorità

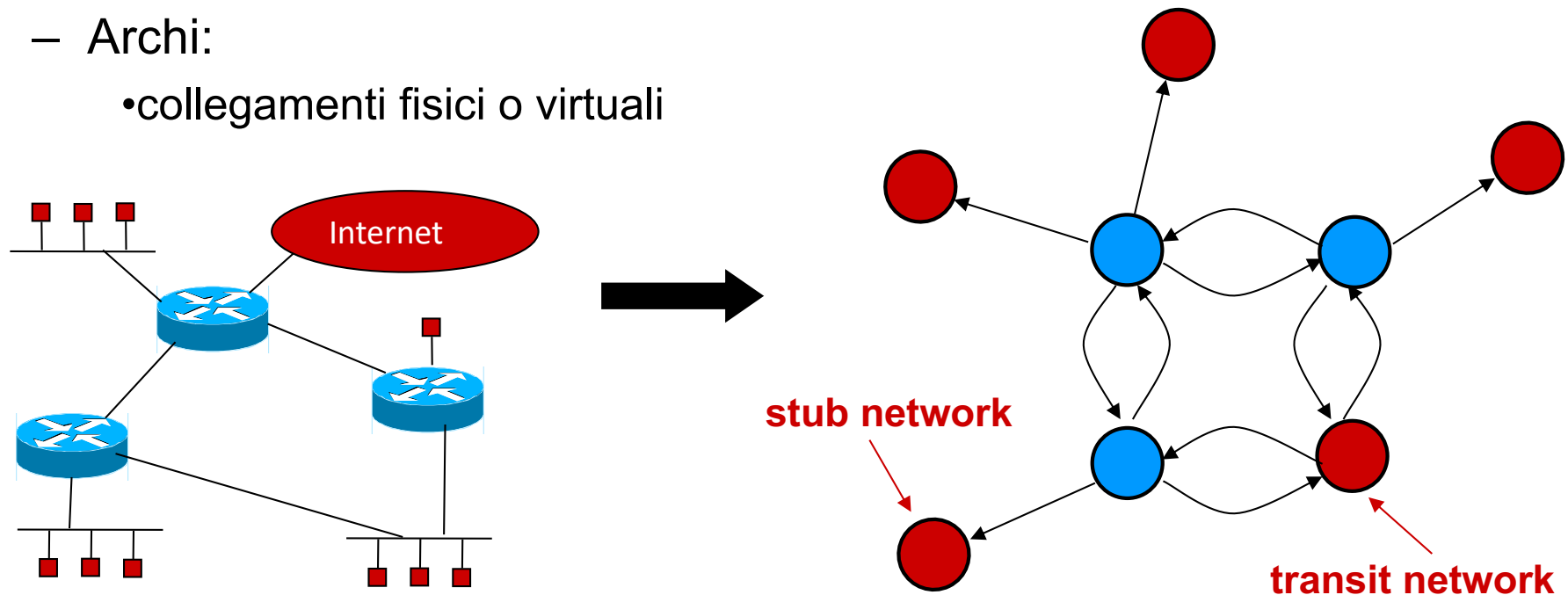
- Ogni router di un AS utilizzante OSPF deve avere un identificativo univoco (**router ID**):
  - di default si prende l'indirizzo IP più alto fra quelli assegnati alle interfacce del router
  - si può assegnare manualmente un router ID ad ogni router configurando opportunamente l'interfaccia di loop-back
  - 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
- Ai singoli router di un'area possono essere associate delle priorità
  - utilizzate nell'elezione del DR
  - valore di priorità compreso tra 0 e 255 (8 bit)
  - di default tutti i router hanno priorità 0 (più bassa)

# 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 risSelected 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 grafo orientato della rete sul quale ciascun router calcola lo **shortest path tree** è rappresentato dal **Link State Database** presente in ogni router
  - Nodi:
    - router
    - reti o host singoli
    - nodi virtuali delle topologie a stella equivalenti
    - destinazioni esterne
  - Archi:
    - collegamenti fisici o virtuali



# OSPF: i protocolli

- 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)

| Version        | Type | Packet Length |
|----------------|------|---------------|
| Router ID      |      |               |
| Area ID        |      |               |
| Checksum       |      | AuType        |
| Authentication |      |               |
| Authentication |      |               |
| ...            |      |               |

# OSPF: intestazione comune

- **Version** indica la versione di OSPF (versione 2)
- **Type** indica il tipo di pacchetto
- **Packet Length** numero di byte del pacchetto
- **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 backbone
- **Checksum** calcolata su tutto il pacchetto OSPF escludendo gli 8 byte del campo authentication
  - si utilizza l'algoritmo classico di IP
- **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
  - Hello (Hello protocol, neighbour discovery)
- Type 2
  - Database description (exchange protocol)
- Type 3
  - Link state request
- Type 4
  - Link state update
- Type 5
  - Link state acknowledge

# OSPF: Hello protocol

- Unico tipo di pacchetto: **Hello** (Type = 1)
- Utilizzato per:
  - controllare l'operatività dei link
  - scoprire e mantenere relazioni fra vicini
  - eleggere DR e BDR

|                          |         |                 |
|--------------------------|---------|-----------------|
| OSPF Header (24 byte)    |         |                 |
| Network Mask             |         |                 |
| HelloInterval            | Options | Router Priority |
| RouterDeadInterval       |         |                 |
| Designated Router        |         |                 |
| Backup Designated Router |         |                 |
| Neighbor                 |         |                 |
| Neighbor                 |         |                 |
| ...                      |         |                 |

# OSPF: Hello protocol

- 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



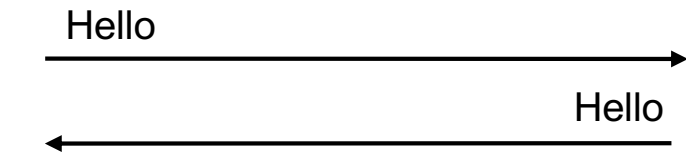
# OSPF: Exchange protocol

- 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
  - 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, 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)

# OSPF: Flooding protocol

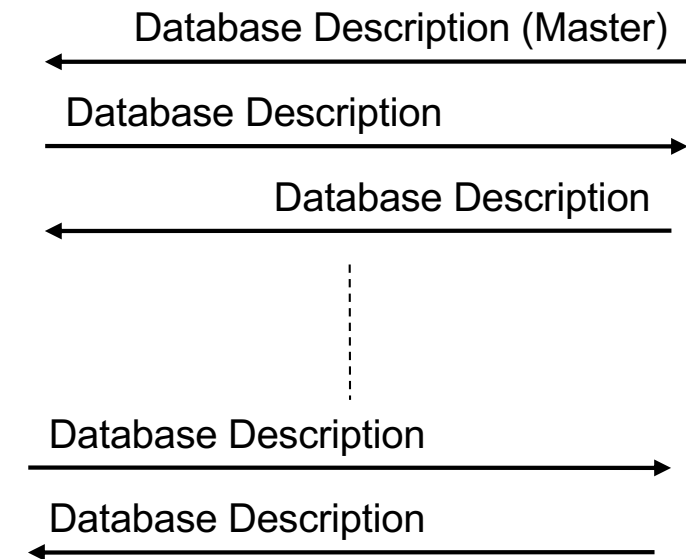
- La diffusione dei LSA a tutti i router della rete avviene tramite l'invio di pacchetti **Link State Update** (Type = 4)
  - a fronte di un cambiamento nello stato di un collegamento
  - a fronte di una Link State Request
  - periodicamente (ogni 30 minuti)
- Si esegue in modalità flooding per fare 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

# OSPF: sincronizzazione e aggiornamento



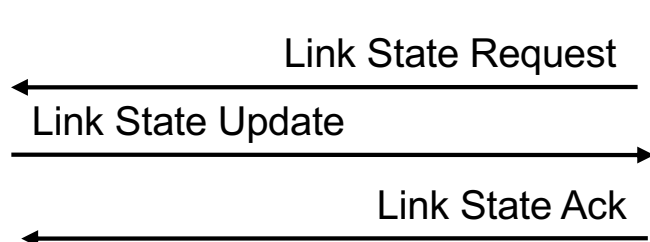
## Fase di **Hello**

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

- **Stub Area** = tipicamente un'area con uno solo punto di interconnessione con il resto della rete
- 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 propagate 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 possibile



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

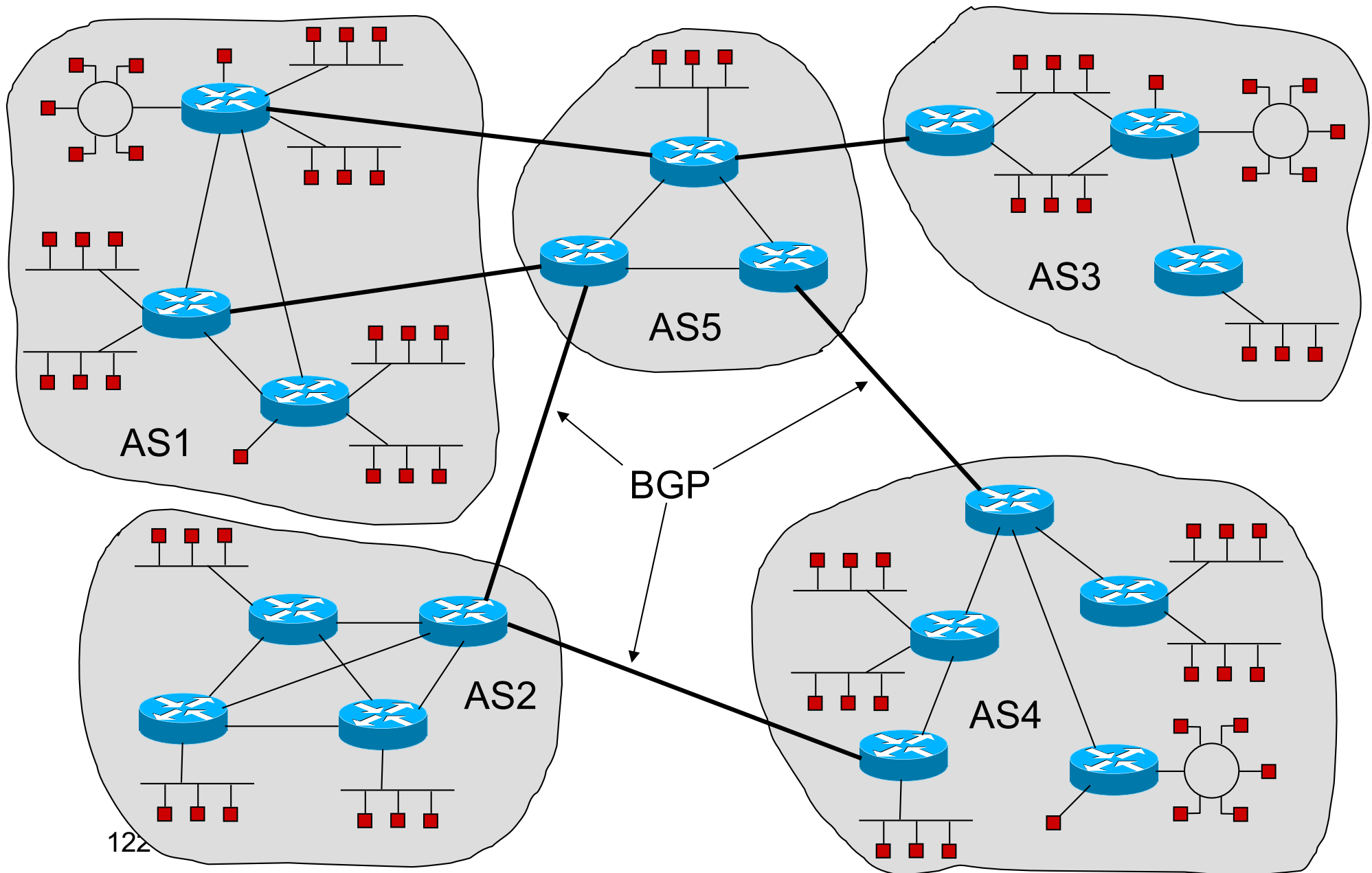




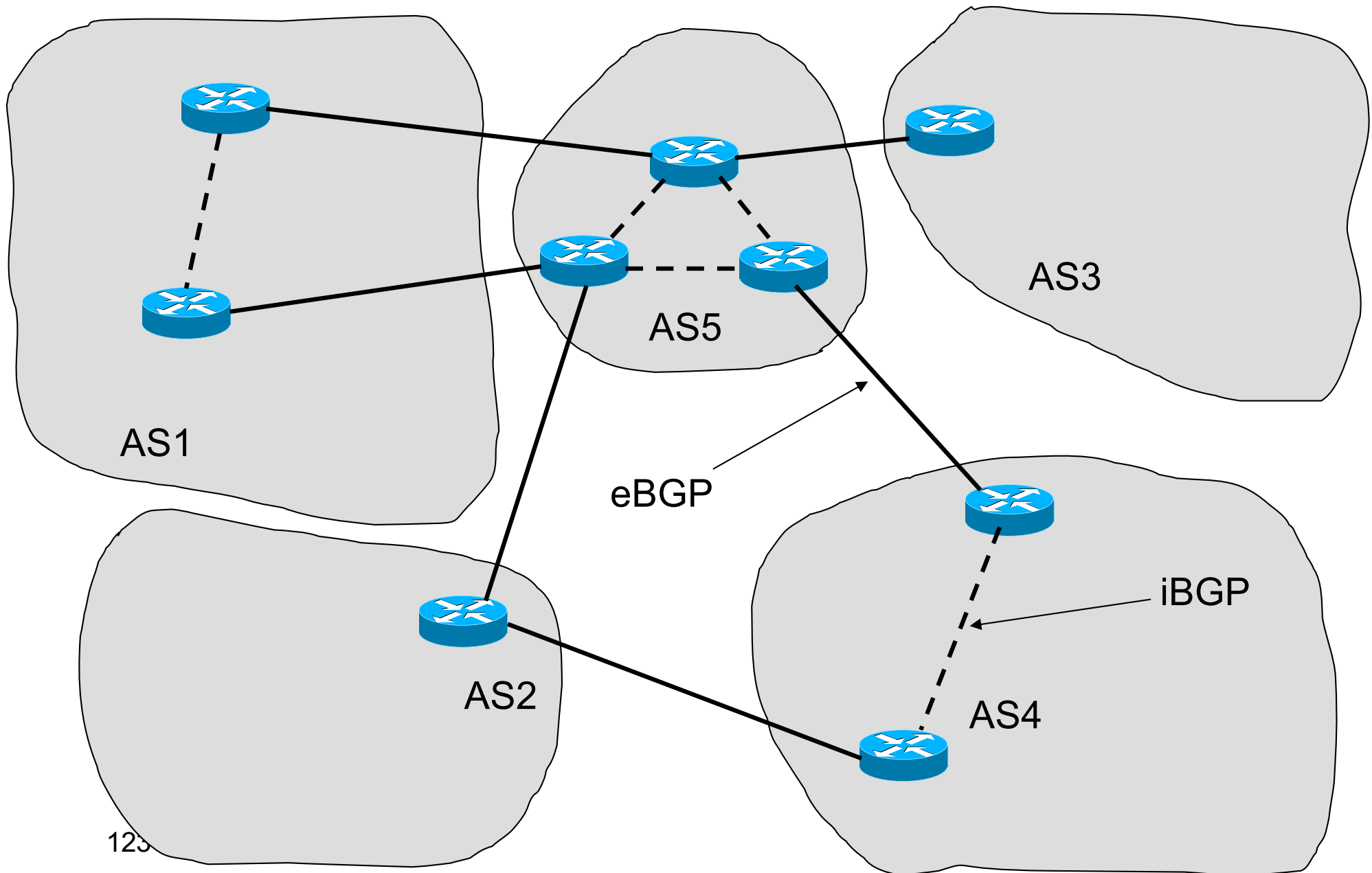
# BGP: Border Gateway Protocol

- **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





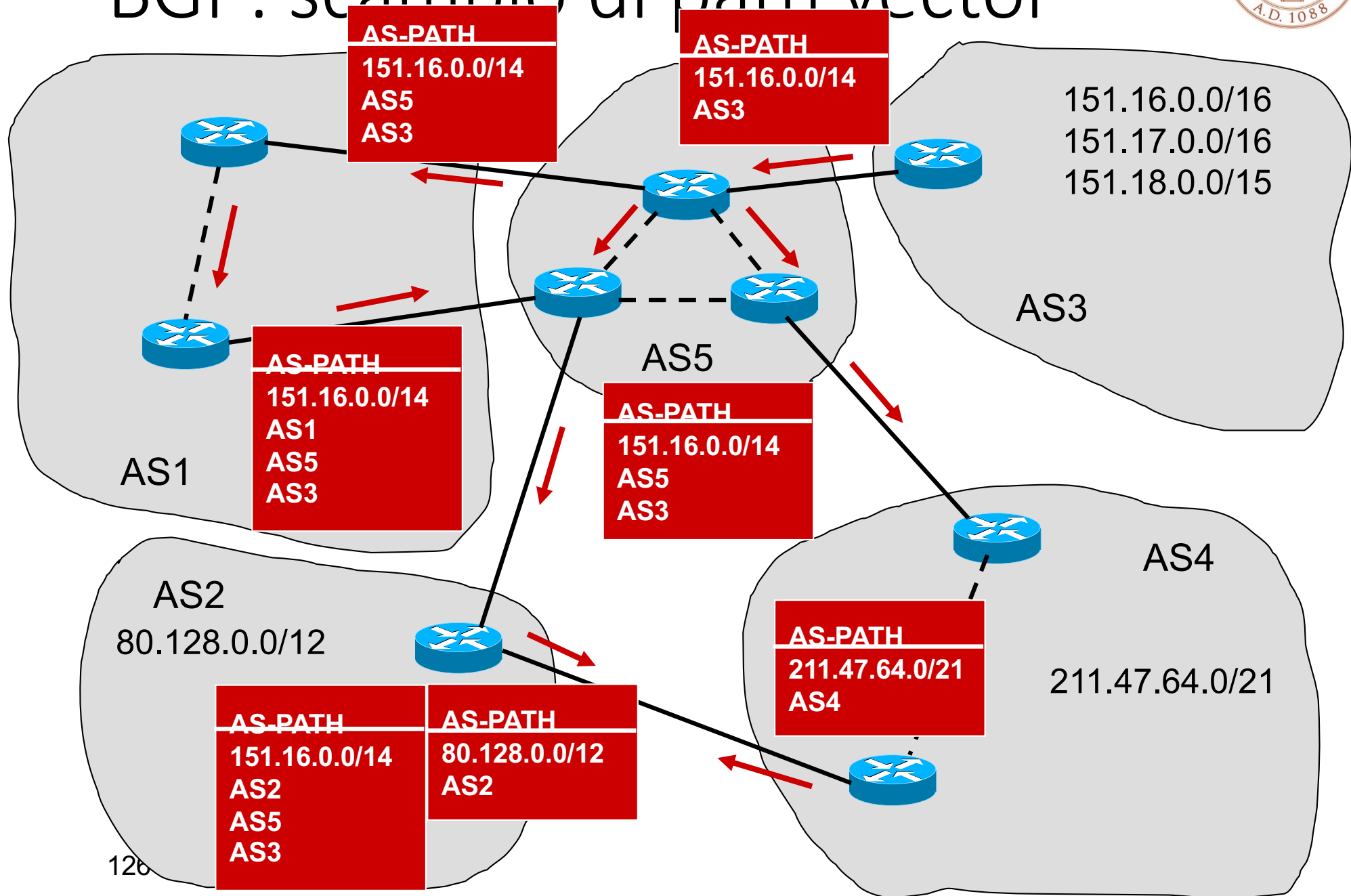
# BGP: Path Vector

- 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



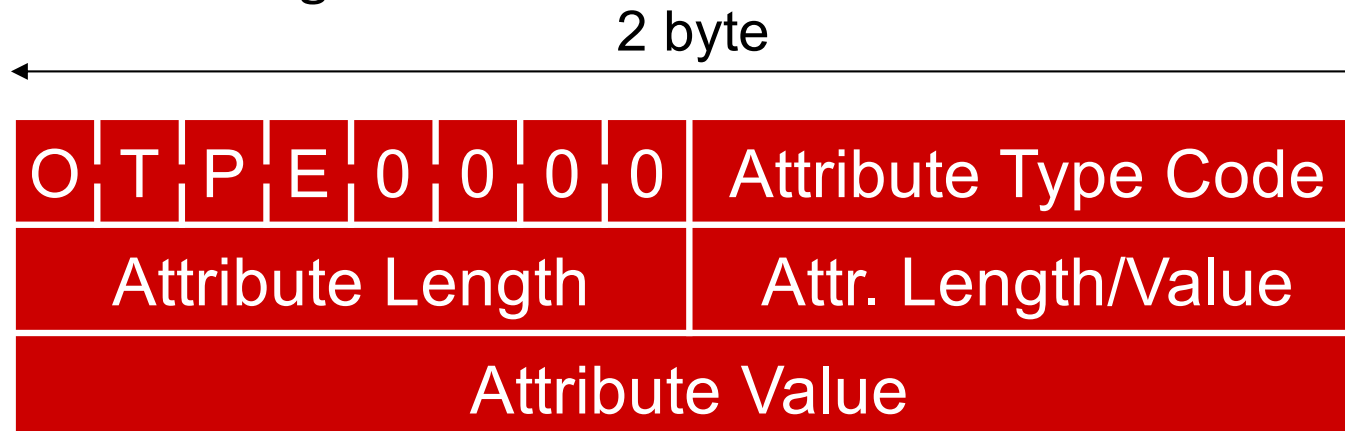
# BGP: attributi

- 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



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

# 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
  - Notification
  - Update
  - 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