

Livello di Rete

Instradamento in Internet

Obiettivo. La funzione di instradamento (routing).

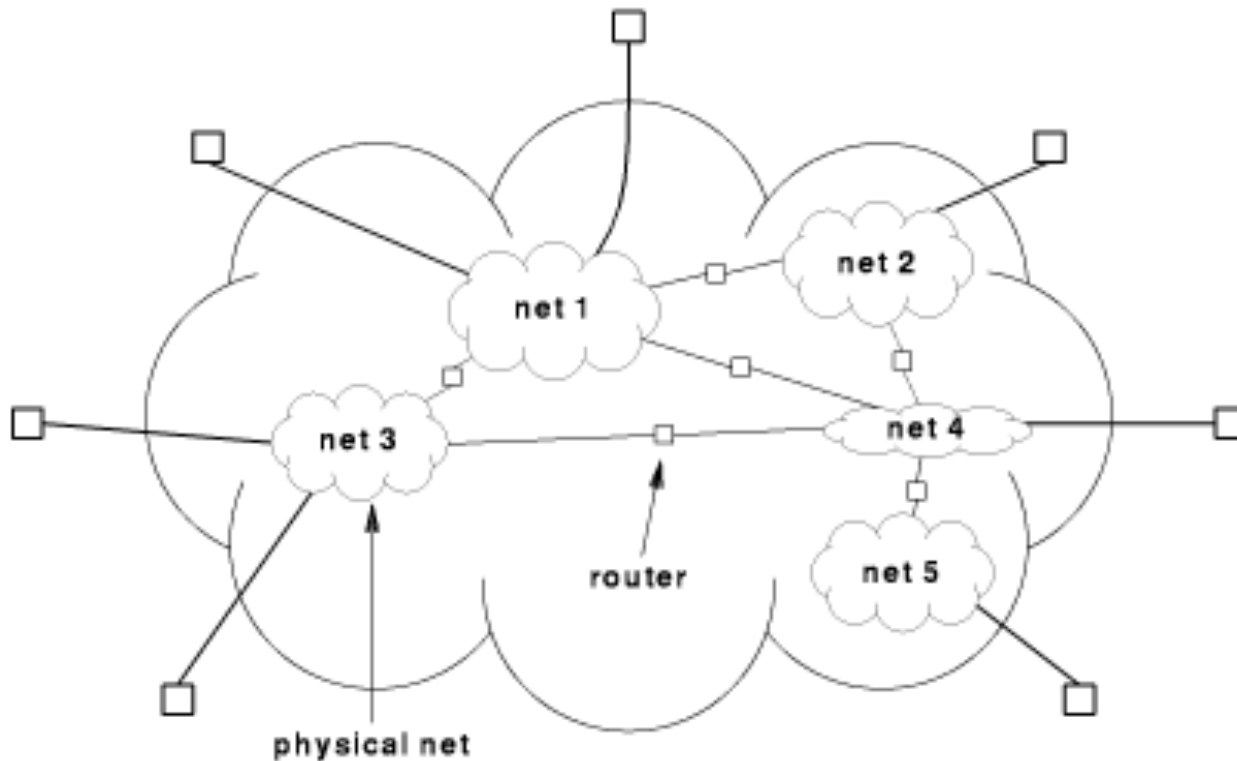
Costruzione della tabella di routing. Algoritmi di routing adattivi: distance vector routing e link-state routing. Routing gerarchico.

Indirizzamento in Internet

IPv4 datagram. Classi di indirizzi IP. Notazione decimale a punti. Indirizzi speciali e privati. Assegnazione degli indirizzi IP. Subnetting e supernetting. CIDR. Inoltro dei datagrammi IP. DHCP. NAT. IPv6

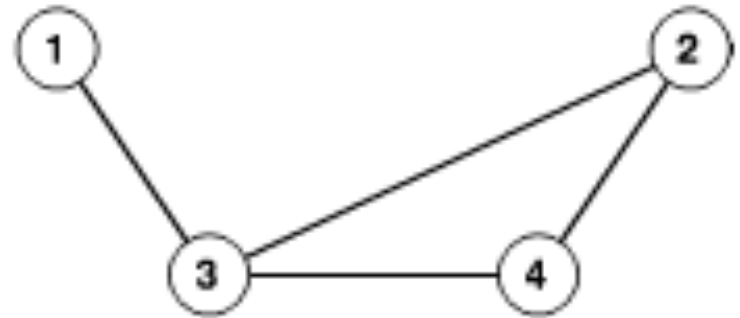
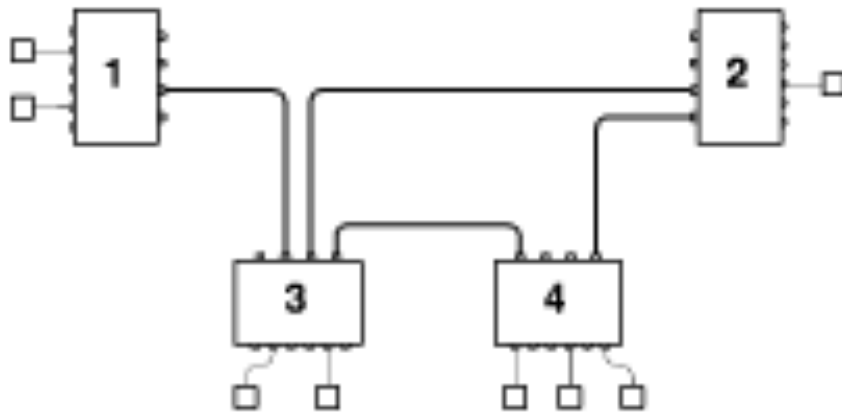
Obiettivo

- Instradare i pacchetti da una sorgente ad una destinazione
- Può essere necessario attraversare diversi router intermedi e quindi scegliere il percorso più appropriato



La funzione di instradamento

- Una rete (o inter-rete) è modellabile come un grafo con nodi che rappresentano gli host ed elementi intermedi (Router) e legami che rappresentano le linee di comunicazione



- La funzione di **instradamento** (*routing*) ha la responsabilità di determinare il percorso seguito dai pacchetti dall'origine alla destinazione
- La funzione di **inoltro** (*forwarding*) ha la responsabilità di trasferire i pacchetti da un'interfaccia di ingresso a una di uscita di un router

Livello di rete in Internet

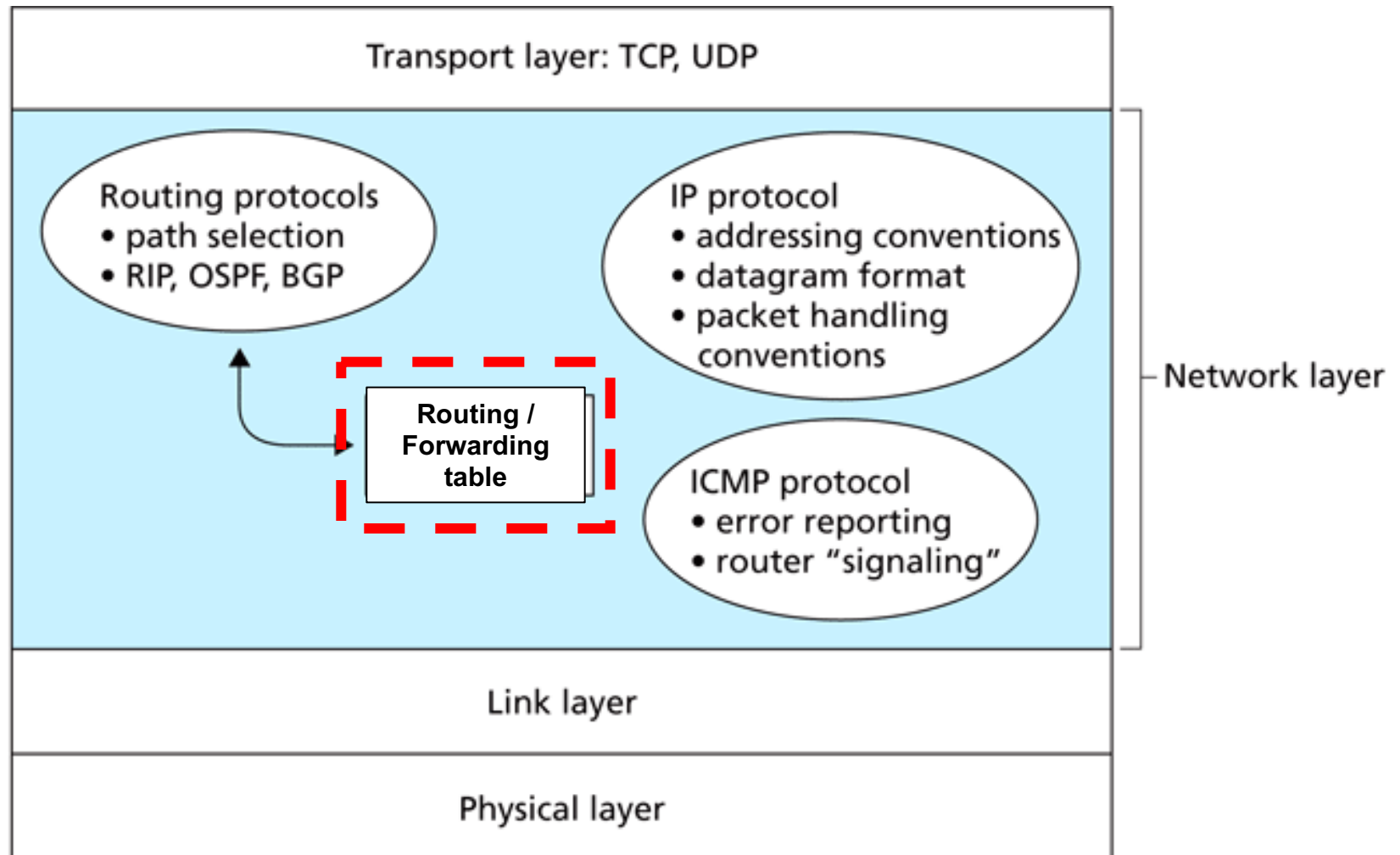


Figure 4.12 ♦ A look inside the Internet's network layer

Routing Table

- Un router realizza la funzione di routing e forwarding per mezzo di una routing table
 - una registrazione per ogni destinazione
 - per ogni registrazione, indirizzo del salto successivo (canale su cui inoltrare il pacchetto)

destin- ation	next hop	destin- ation	next hop	destin- ation	next hop	destin- ation	next hop
1	-	1	(2,3)	1	(3,1)	1	(4,3)
2	(1,3)	2	-	2	(3,2)	2	(4,2)
3	(1,3)	3	(2,3)	3	-	3	(4,3)
4	(1,3)	4	(2,4)	4	(3,4)	4	-
<i>node 1</i>		<i>node 2</i>		<i>node 3</i>		<i>node 4</i>	

Costruzione della routing table

- La routing table può essere costruita come output di un **algoritmo di routing**
- Il problema da risolvere è: ***trovare il cammino minimo tra due nodi***
- Gli archi però sono pesati!
 - Metrica di prestazione/costo: distanza, lunghezza delle code, ritardo medio di un pacchetto, larghezza di banda del canale
- Fattori critici: topologia e traffico

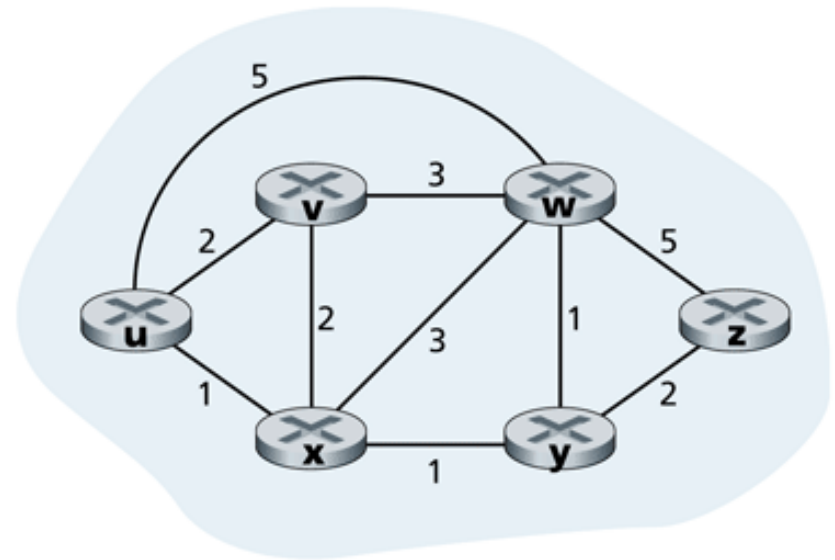


Figure 4.27 ♦ Abstract graph model of a computer network

Classificazione degli algoritmi di routing

statico vs dinamico

- Routing statico
(algoritmi non adattivi)
 - la scelta del cammino è calcolata in anticipo e memorizzata in una memoria non volatile
- Routing dinamico
(algoritmi adattivi)
 - la scelta del cammino varia secondo i cambiamenti di topologia e di traffico

globale vs decentralizzato

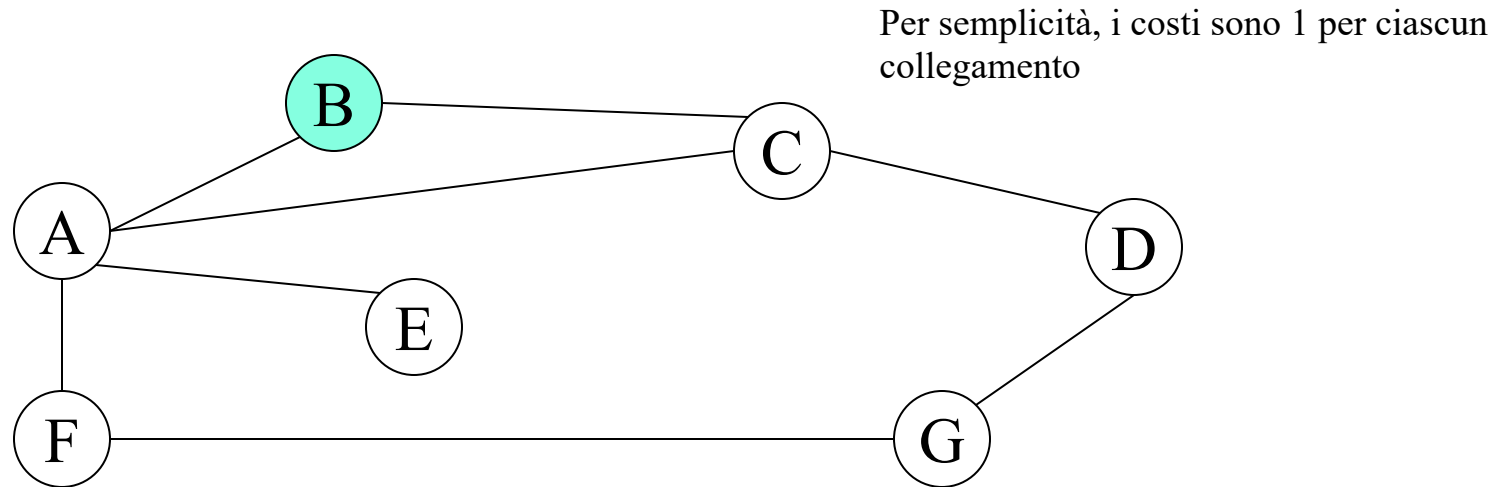
- Routing globale (o centralizzato)
 - L'algoritmo riceve in ingresso tutti i collegamenti tra i nodi e i loro costi
- Routing decentralizzato
 - Nessun nodo possiede informazioni complete sulla rete
 - Il cammino a costo minimo viene calcolato in modo distribuito e iterativo

Algoritmo del vettore distanza (o di Bellman-Ford)

Distance-Vector routing

- Routing **dinamico** e **decentralizzato**
- Implementato dal protocollo RIP (routing information protocol)
- Ogni router mantiene una tabella contenente la migliore distanza (costo) conosciuta per ogni destinazione e quale canale utilizzare per raggiungerla
(Destination, Cost, NextHop)

Algoritmo del vettore distanza: rete di esempio



Routing table
del nodo B

Destination	Cost	Next Hop
A	1	A
C	1	C
D	2	C
E	2	A
F	2	A
G	3	A

Algoritmo del vettore distanza: stato iniziale

- ogni nodo conosce i costi dei collegamenti ai nodi adiacenti
- i collegamenti ai nodi non adiacenti o i collegamenti interrotti hanno costo infinito

	A	B	C	D	E	F	G
A	0	1	1	∞	1	1	∞
B	1	0	1	∞	∞	∞	∞
C	1	1	0	1	∞	∞	∞
D	∞	∞	1	0	∞	∞	1
E	1	∞	∞	∞	0	∞	∞
F	1	∞	∞	∞	∞	0	1
G	∞	∞	∞	1	∞	1	0

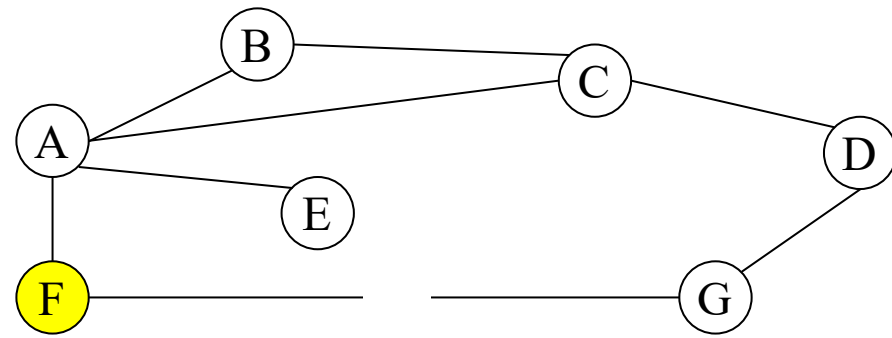
Algoritmo del vettore distanza: aggiornamento tabelle

- Ogni nodo spedisce aggiornamenti ai suoi nodi adiacenti e riceve aggiornamenti dai suoi nodi adiacenti
 - periodicamente
 - se la tabella cambia (triggered update)
- Ogni aggiornamento è il vettore distanza
 - lista di coppie (Destination, Cost)
- Un nodo aggiorna la tabella locale se riceve un cammino migliore:
 - somma il costo minore al costo del nodo che lo ha proposto
 - es. A propone a B un costo 1 per raggiungere E: B aggiorna il costo per raggiungere E con $2 = 1 + 1$
 - prende nota (come Next Hop) del nodo che lo ha proposto
- Dopo aver scambiato diversi aggiornamenti con i nodi adiacenti, tutti i nodi conosceranno i cammini migliori verso tutti gli altri nodi

Algoritmo del vettore distanza: stato finale

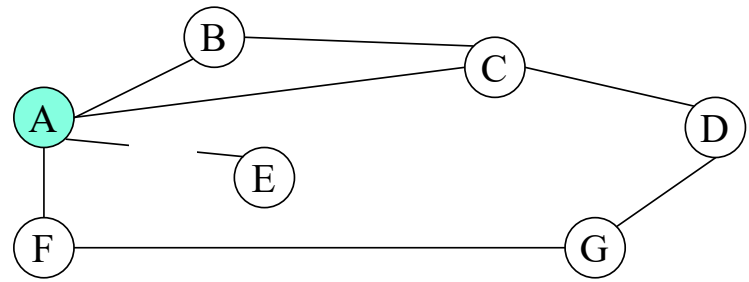
	A	B	C	D	E	F	G
A	0	1	1	2	1	1	2
B	1	0	1	2	2	2	3
C	1	1	0	1	2	2	2
D	2	2	1	0	3	2	1
E	1	2	2	3	0	2	3
F	1	2	2	2	2	0	1
G	2	3	2	1	3	1	0

Che succede se il collegamento tra F e G si interrompe?



- Il sistema aggiorna le proprie tabelle e ritorna di nuovo stabile
 - F si accorge che il collegamento a G è interrotto
 - F aggiorna a infinito la distanza da G e spedisce l'aggiornamento ad A
 - A aggiorna a infinito la distanza da G perché usa F per raggiungere G
 - A riceve un aggiornamento periodico da C con un percorso verso G di costo 2
 - A aggiorna a costo 3 via C la distanza da G e spedisce l'aggiornamento a F
 - F aggiorna a costo 4 via A la distanza da G

Che succede se il collegamento tra A ed E si interrompe?



- Problema del conteggio all'infinito: Il sistema non si stabilizza e nessuno riesce a sapere che E è irraggiungibile
 - A si accorge che il collegamento ad E è interrotto
 - A aggiorna a infinito la distanza da E spedisce l'aggiornamento a B e C (ignoriamo F per semplicità)
 - B aggiorna ad infinito la distanza da E perché usa A per raggiungere E
 - Supponiamo che B riceva un aggiornamento da C (spedito prima che C ricevesse l'aggiornamento da A) che lo informa che E è raggiungibile a costo 2

NO!

- B aggiorna a costo 3 via C la distanza da E e spedisce l'aggiornamento ad A
- A aggiorna a costo 4 via B la distanza da E e spedisce l'aggiornamento a C
- C (che ora ha un costo infinito per E perché ha ricevuto da A l'aggiornamento nel frattempo) aggiorna a costo 5 via A la distanza da E e spedisce l'aggiornamento a B ...

- Soluzione:
 - tagliare i cicli di routing (split horizon): non si rimandano al nodo adiacente la notifica di aggiornamenti effettuati su informazioni provenienti da quel nodo (cioè B non rimanda ad A aggiornamento su E)

Algoritmo dello stato della connessione

Link-State routing

- Routing **dinamico** e **globale**
- Implementato dal protocollo OSPF (open shortest path first)
- Funzioni di base
 - Distribuzione affidabile a tutti i nodi delle informazioni sullo stato dei collegamenti (link-state broadcast)
 - Ogni nodo sarà in grado di costruire una mappa completa della rete
 - Calcolo presso ciascun nodo del cammino minimo verso una data destinazione

Distribuzione affidabile dello stato delle connessioni

Strategia

- spedire a tutti i nodi (non solo quelli adiacenti) le informazioni sui collegamenti diretti

Tattica: reliable flooding

- il nodo emette su tutte le sue linee in uscita le informazioni sui collegamenti diretti
- i nodi riceventi propagano le informazioni ricevute su tutte le loro linee in uscita
- la distribuzione va avanti finchè le informazioni non si sono propagate a tutti i nodi della rete

Pacchetto di stato della connessione (**Link State Packet, LSP**)

- id del nodo che ha creato l'LSP
- costo del collegamento per ogni nodo adiacente
- numero di sequenza (SEQNO a 64 bit)
- tempo di vita (TTL)

Reliable flooding

- memorizza l'LSP più recente da ogni nodo
- Se riceve un nuovo LSP (SEQNO maggiore) propaga l'LSP a tutti i nodi eccetto il nodo da cui proviene
- genera periodicamente un nuovo LSP incrementando il SEQNO
- inizializza SEQNO a 0 al reboot
- decrementa TTL di ogni LSP memorizzato: un pacchetto con TTL a 0 è scartato

Calcolo del cammino minimo

- Con la ricezione di tutti i pacchetti LSP ogni nodo è in grado di costruire la mappa completa della rete
- Algoritmo di Dijkstra per il calcolo del cammino minimo
 - Dato un grafo con un costo non negativo assegnato ad ogni arco e un nodo sorgente designato
 - Calcola la distanza (costo) minima dal nodo sorgente a ogni altro nodo e una next hop routing table
- Esistono diverse varianti

Una variante dell'algoritmo di Dijkstra

- Grafo $G = (N, E)$, N insieme dei nodi ed E l'insieme degli archi
- $weight(i,j)$ denota il costo (weight) associato all'arco (i,j)
- $s \in N$ denota il nodo sorgente considerato
- $M \subseteq N$ denota l'insieme dei nodi già esaminati
- $D(n)$ denota la distanza (costo) del cammino minimo da s al nodo n
- $R(n)$ denota il next hop nel cammino minimo da s al nodo n

```
/* inizializzazione */
```

```
 $M = \{s\}$ 
```

```
for each  $n \in N - \{s\}$  {
```

```
    if  $(s,n) \in E$  {
```

```
         $D(n) = weight(s,n)$ 
```

```
         $R(n) = n$ 
```

```
    }
```

```
    else {
```

```
         $D(n) = \infty$ 
```

```
         $R(n) = 0$ 
```

```
    }
```

```
}
```

C'è collegamento diretto tra n e s

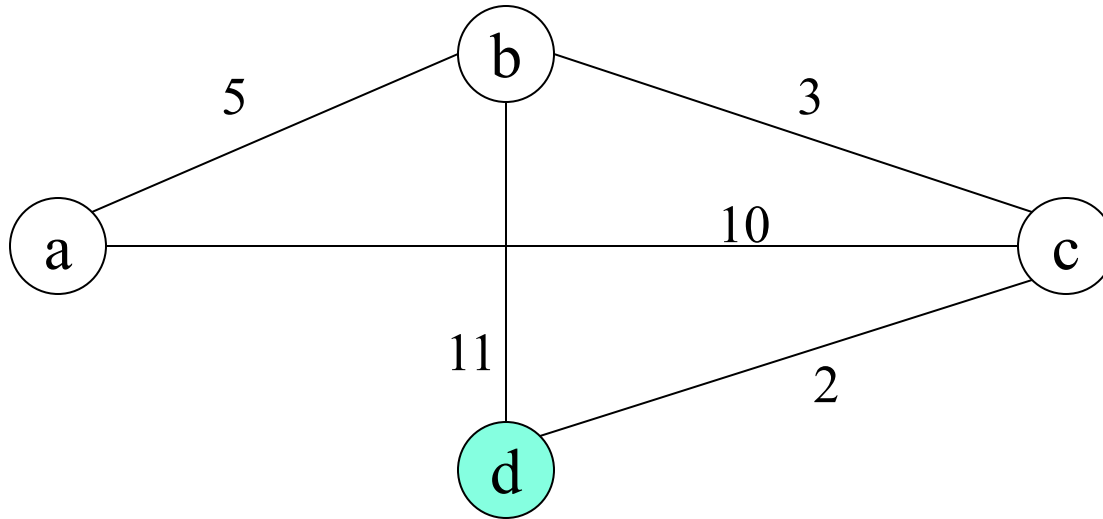
∞ se non c'è collegamento diretto

Una variante dell'algoritmo di Dijkstra

- Grafo $G = (N, E)$, N insieme dei nodi ed E l'insieme degli archi
- $weight(i,j)$ denota il costo (weight) associato all'arco (i,j)
- $s \in N$ denota il nodo sorgente considerato
- $M \subseteq N$ denota l'insieme dei nodi già esaminati
- $D(n)$ denota la distanza (costo) del cammino minimo da s al nodo n
- $R(n)$ denota il next hop nel cammino minimo da s al nodo n

```
/* visita dei nodi */
while ( $M \neq N$ ) {
    scegli  $u$  tale che  $D(u)$  è il minimo  $\forall u \in (N - M)$ 
     $M = M \cup \{u\}$ 
    for each  $n \in (N - M)$  tale che  $(u,n) \in E$  {
         $c = D(u) + weight(u,n)$ 
        if ( $c < D(n)$ ) {
             $R(n) = R(u)$ 
             $D(n) = c$ 
        }
    }
}
```

Algoritmo dello stato delle connessioni: rete di esempio

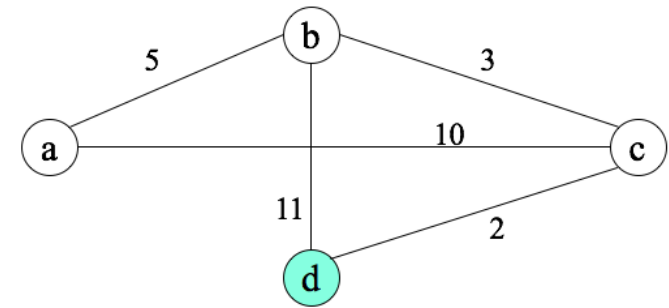


Routing table
del nodo d

Destination	Distance (Cost)	Next Hop
a	10	c
b	5	c
c	2	c

Costruzione routing table nodo d

Inizializzazione



$M = \{d\}$

Destination	Distance (cost)	Next Hop
a	∞	0
b	11	b
c	2	c

$N = \{a,b,c,d\}$ $s = d$

$E = \{(a,b), (b,c), (c,d), (a,c), (b,d)\}$

weight(i,j):

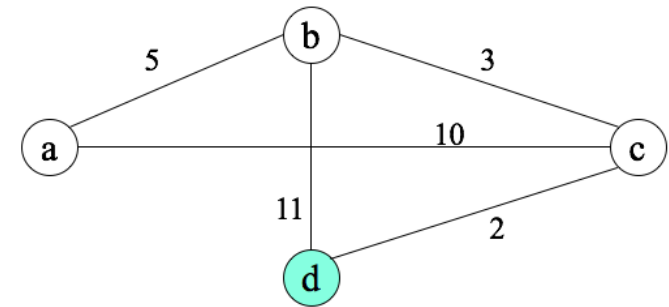
weight (a,b) = 5

weight (b,c) = 3

...

Costruzione routing table nodo d

Visita nodi



Destination	Distance (cost)	Next Hop
a	∞ 12	0 c
b	11 5	b c
c	2	c

$M = \{d\}$

$u = c$

$N = \{a,b,c,d\}$ $s = d$

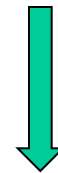
$E = \{(a,b), (b,c), (c,d), (a,c), (b,d)\}$

weight(i,j):

weight (a,b) = 5

weight (b,c) = 3

...

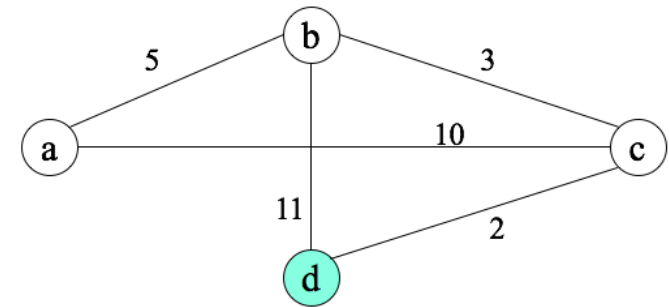


$M = \{d, c\}$

$M \subseteq N$, avanti

Costruzione routing table nodo d

Visita nodi



Destination	Distance (cost)	Next Hop
a	∞ 12 10	0 c
b	11 5	b c
c	2	c

$M = \{d, c\}$

$u = b$

$N = \{a, b, c, d\}$ $s = d$

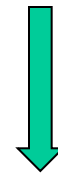
$E = \{(a, b), (b, c), (c, d), (a, c), (b, d)\}$

weight(i,j):

weight (a,b) = 5

weight (b,c) = 3

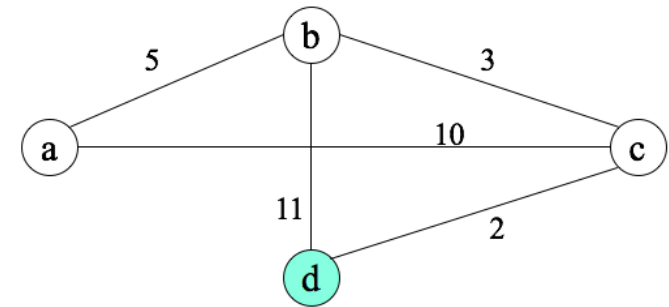
...



$M = \{d, c, b\}$

$M \subseteq N$, avanti

Costruzione routing table nodo d



Visita nodi

Destination	Distance (cost)	Next Hop
a	∞ 12 10	0 c
b	11 5	b c
c	2	c

$M = \{d, c, b\}$

$u = a$ Nessun
cammino
migliore

$N = \{a, b, c, d\}$ $s = d$

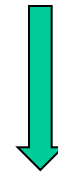
$E = \{(a, b), (b, c), (c, d), (a, c), (b, d)\}$

weight(i,j):

weight (a,b) = 5

weight (b,c) = 3

...



$M = \{d, c, b, a\}$

$M = N$ stop

Confronto tra i due algoritmi

Distance Vector Routing

- un nodo scambia informazioni (vettore distanza) solo con i nodi adiacenti
- un nodo invia tutte le informazioni di routing in suo possesso
- Converge più lentamente: richiede $O(|N| \cdot |E|)$ messaggi
- non richiede molta memoria

Link State Routing

- un nodo scambia informazioni (LSP) con tutti gli altri nodi
- un nodo invia le informazioni di routing *sicure* cioè solo quelle relative ai nodi adiacenti
- Converge più rapidamente: da $O(|N|^2)$ fino a $O(|E| + |N| \log |N|)$ messaggi (div. implement.)
- molta memoria richiesta per memorizzare gli LSP

Routing gerarchico

- Problema delle dimensioni della routing table
- Soluzione: suddivisione gerarchica dei router in regioni, anche dette sistemi autonomi (AS)
 - ogni routing table contiene:
 - una registrazione per ogni router del proprio AS
 - una registrazione per almeno un router di frontiera (gateway router)
- Interior Gateway Protocols:
protocolli di routing interni a un AS
 - RIP (Routing Information Protocol) basato su distance vector routing
 - OSPF (Open Shortest Path First) basato su link state routing
- Exterior Gateway Protocols:
protocolli di routing tra AS
 - BGP (Border Gateway Protocol)

IP - Internet Protocol

- Fornisce un metodo best-effort per trasportare datagram dalla sorgente alla destinazione, indipendentemente dall'esistenza di reti intermedie lungo il percorso
- Non orientato alla connessione:
 - ogni datagram contiene l'indirizzo di partenza e di destinazione e può seguire un percorso diverso
- Non affidabile:
 - i datagram possono non arrivare o arrivare nell'ordine sbagliato
- Versione più diffusa: IPv4 (indirizzo 32 bit)
- Versione più recente: IPv6 (indirizzo 128 bit, sicurezza, ...)

Formato di un datagramma IPv4

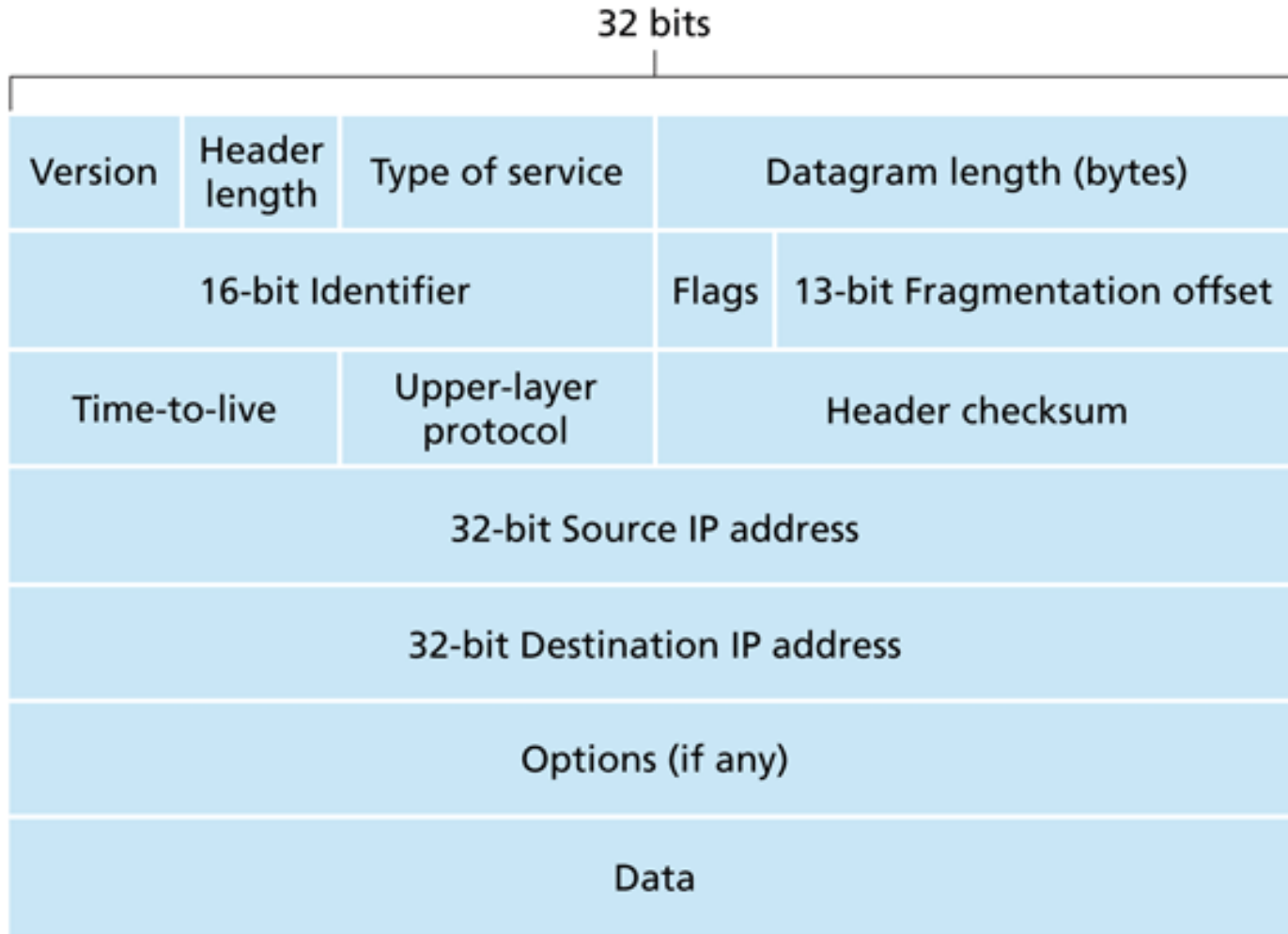
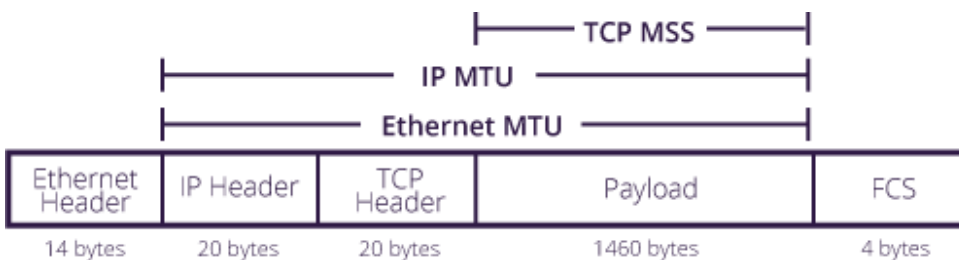
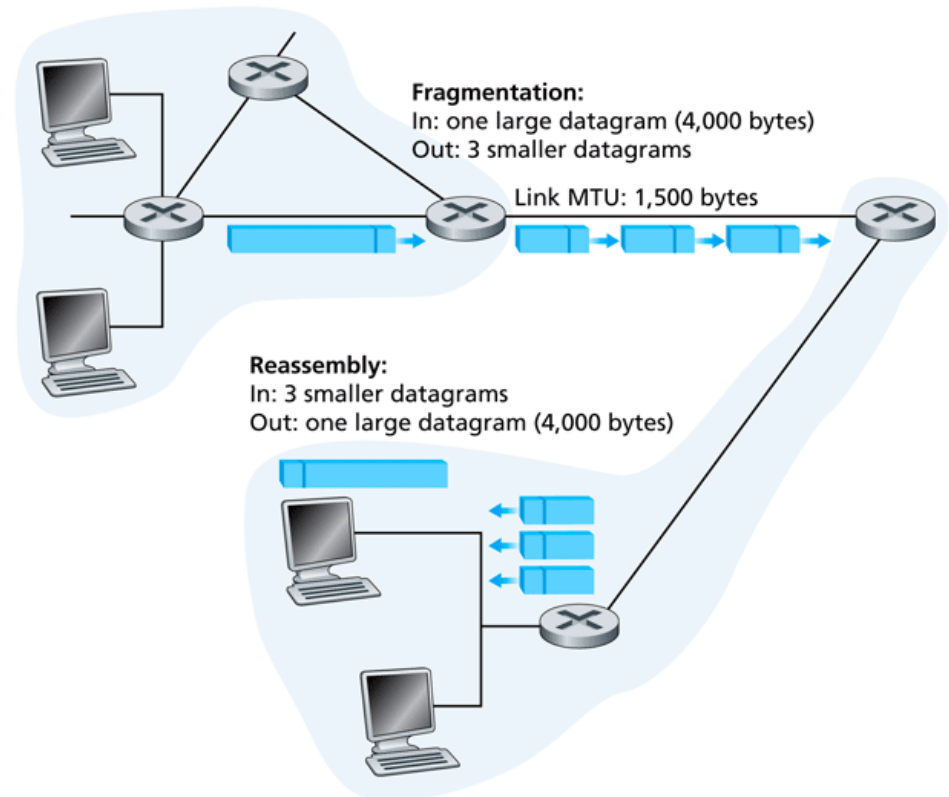


Figure 4.13 ♦ IPv4 datagram format

Frammentazione datagrammi IP

- L'unità massima di trasmissione (MTU) è la massima quantità di dati che un frame a livello di collegamento può trasportare
 - Differenti tipi di link, differenti MTU
- I datagram IP possono essere frammentati e poi ricostruiti alla destinazione
- I datagram IP possono arrivare teoricamente fino a 64KB
 - quando un host è collegato a una LAN Ethernet sono lunghi circa 1500 byte per non dover frammentare già in partenza



IP fragmentation and reassembly

Nodi e indirizzi IP

- Un'interfaccia di rete è il confine tra un nodo e il collegamento fisico
- Ogni interfaccia di host e router ha un indirizzo IP
 - Un host, in genere, ha unico indirizzo IP
 - I router hanno almeno due indirizzi IP
 - Anche un proxy server ha 2 indirizzi IP

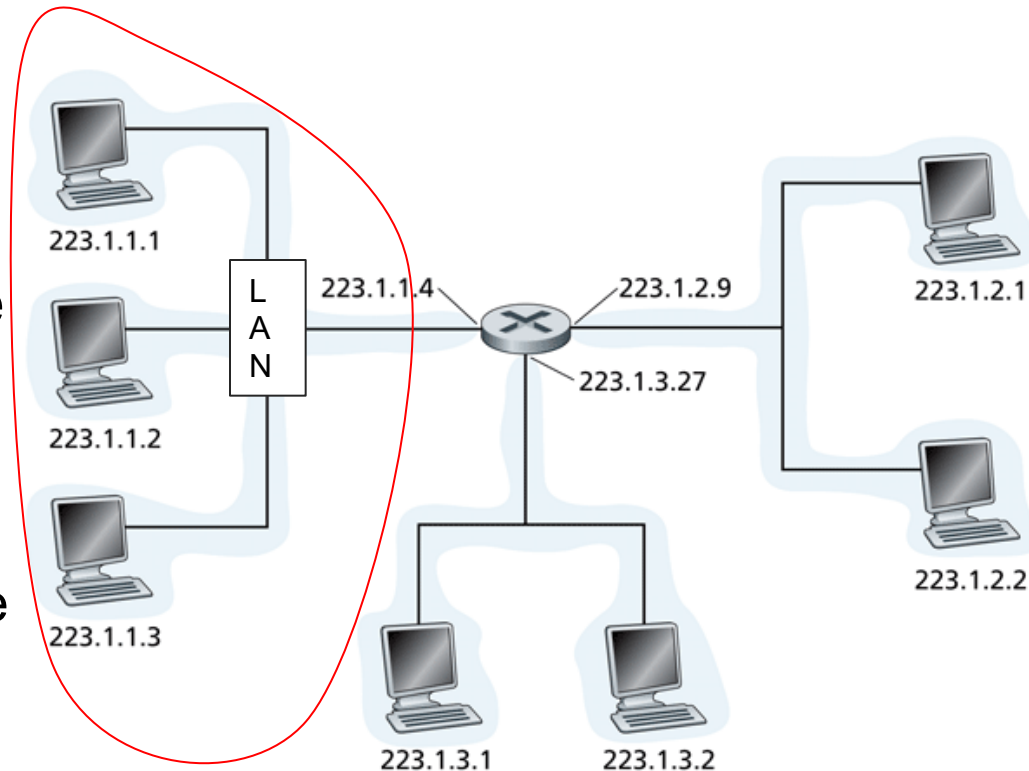
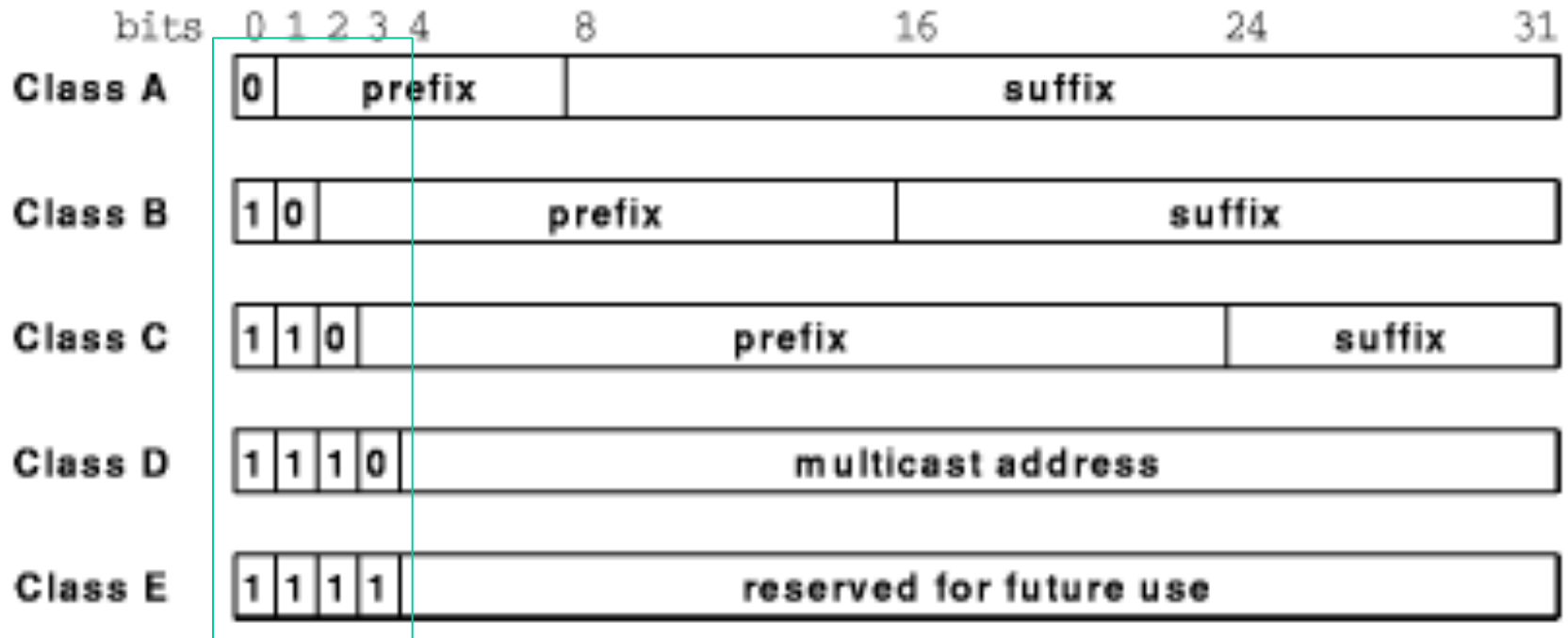


Figure 4.15 ♦ Interface addresses and subnets

Formato di un indirizzo IP

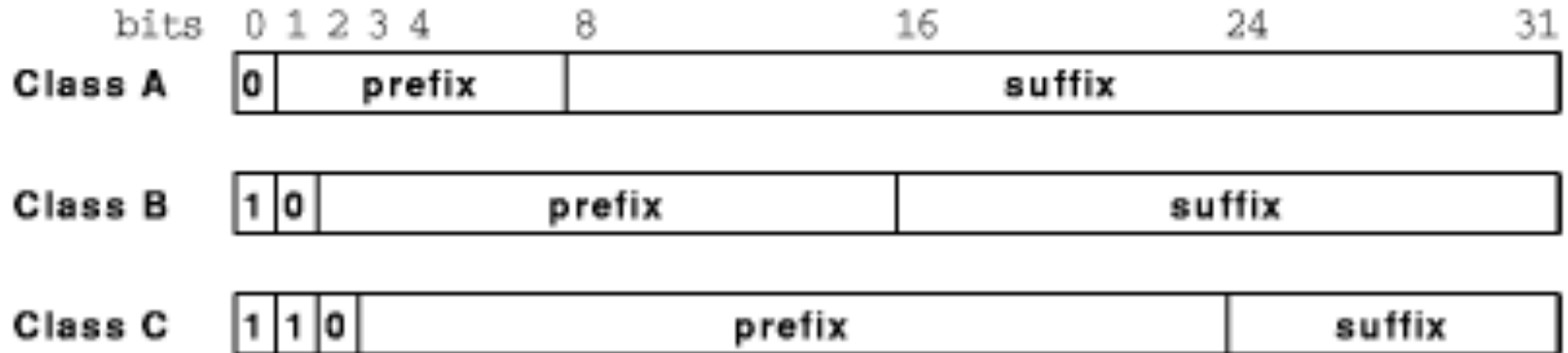
- Indirizzo formato da un prefisso e un suffisso
- Il prefisso identifica la rete (netid)
 - ogni rete in Internet ha un netid unico che è specificato dal NIC (network interface controller)
- Il suffisso identifica l'host all'interno della rete (hostid)
 - l'hostid è unico per quel prefisso ed è gestito dall'amministratore della rete

Classi di indirizzi IP



Address Class	Bits In Prefix	Maximum Number of Networks	Bits In Suffix	Maximum Number Of Hosts Per Network
A	7	128	24	16777216
B	14	16384	16	65536
C	21	2097152	8	256

Maschera di rete



Class A: 11111111 . 00000000 . 00000000 . 00000000
 255 . 0 . 0 . 0

Class B: 11111111 . 11111111 . 00000000 . 00000000
 255 . 255 . 0 . 0

Class C: 11111111 . 11111111 . 11111111 . 00000000
 255 . 255 . 255 . 0

Notazione decimale a punti

32-bit Binary Number	Equivalent Dotted Decimal
10000001 00110100 00000110 00000000	129 . 52 . 6 . 0
11000000 00000101 00110000 00000011	192 . 5 . 48 . 3
00001010 00000010 00000000 00100101	10 . 2 . 0 . 37
10000000 00001010 00000010 00000011	128 . 10 . 2 . 3
10000000 10000000 11111111 00000000	128 . 128 . 255 . 0

Class	Range of Values
A	0 through 127
B	128 through 191
C	192 through 223
D	224 through 239
E	240 through 255

Indirizzi speciali



Prefix	Suffix	Type Of Address	Purpose
all-0s	all-0s	this computer	used during bootstrap
network	all-0s	network	identifies a network
network	all-1s	directed broadcast	broadcast on specified net
all-1s	all-1s	limited broadcast	broadcast on local net
127	any	loopback	testing

Subnetting

- Sottorete
 - una rete isolata i cui punti terminali sono collegati all'interfaccia di un host o di un router
- Necessità di decomporre in sottoreti blocchi di indirizzi troppo grandi
 - Specificare la sottorete usare non solo i bit del prefisso ma anche i primi bit del suffisso
 - Non è più sufficiente conoscere la classe di indirizzo per riconoscere il netid
- Subnet mask (maschera di sottorete)
 - specifica quale parte dell'indirizzo indica la sottorete e quale indica l'host
 - 255.0.0.0 subnet mask di default per classe A
 - 255.255.0.0 subnet mask di default per classe B
 - 255.255.255.0 subnet mask di default per classe C

Notazione CIDR

Classless InterDomain Routing

- Schema di indirizzamento che ha sostituito lo schema classfull per indicare un blocco di indirizzi IP
 - *Blocco di indirizzi IP / x*
x è il numero di bit (a partire da sinistra) del netid
 - 25.0.0.0/8 \Leftrightarrow 25.0.0.0 – 25.255.255.255
 - 25.50.0.0/16 \Leftrightarrow 25.50.0.0 – 25.50.255.255
 - 193.204.187.0 / 24 \Leftrightarrow 193.204.187.0 - 193.204.187.255
- CIDR consente di definire ossia subnet mask diverse da quelle standard
 - 193.204.187.0 / 28 \Leftrightarrow 193.204.187.0 - 193.204.187.15
n° host 14 validi
subnet mask 255.255.255.240

Subnetting: Esercizi

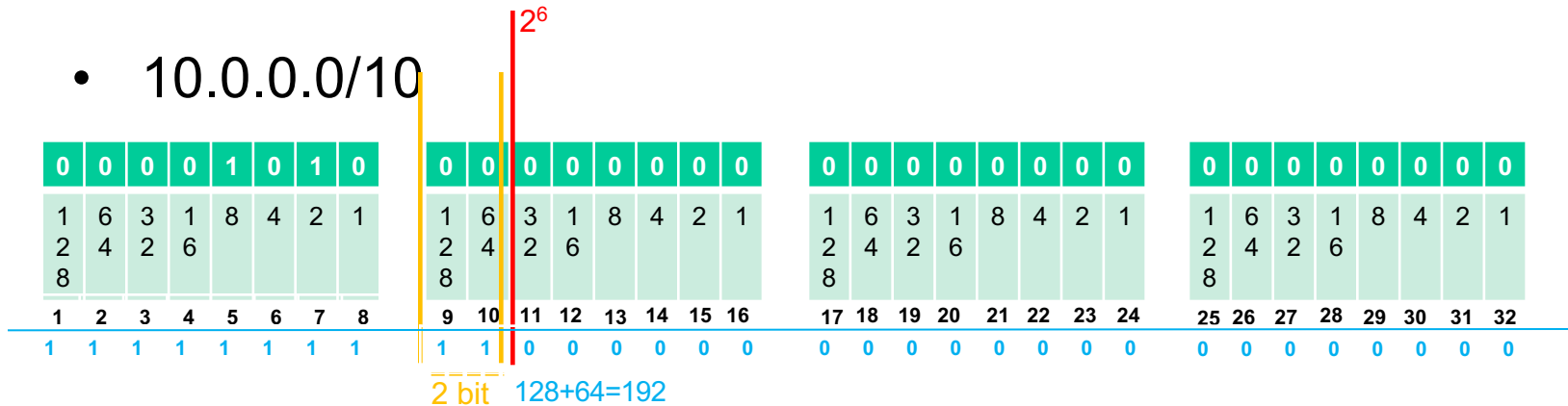
- <https://www.calculator.net/ip-subnet-calculator.html>
- <https://www.rapidtables.com/convert/number/decimal-to-binary.html>

Esempio #1: 2° ottetto

- 10.0.0.0/10
- Domande
 - Quante sottoreti?
 - Quanti indirizzi host per sottorete?
 - Quanti validi?
 - Qual è la subnet mask?
 - Qual è il netid della prima sottorete
 - Qual è il primo indirizzo valido? E l'ultimo?
 - Qual è l'indirizzo di broadcast
 - Qual è il range di indirizzi della seconda sottorete?

Esempio #1: 2° ottetto

- 10.0.0.0/10



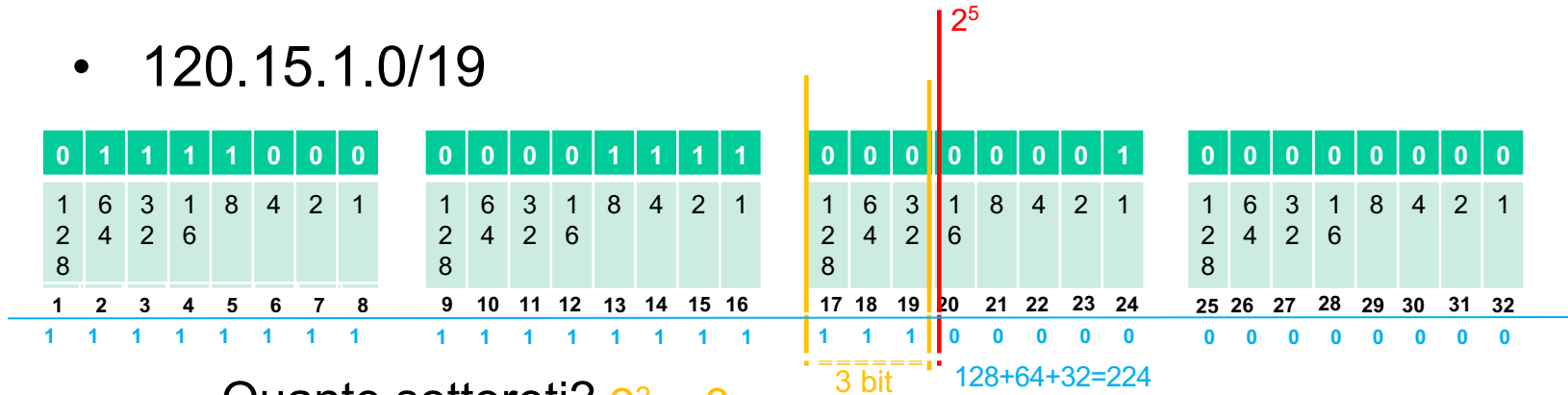
- Quante sottoreti? $2^2 = 4$
- Quanti indirizzi host per sottorete? $2^{22} = 4\ 194\ 304$
 - Quanti validi? $2^{22} - 2 = 4\ 194\ 302$
- Qual è la subnet mask? 255.192.0.0
- Qual è il netid della prima sottorete 10.0.0.0
 - Qual è il primo indirizzo valido? E l'ultimo? 10.0.0.1 - 10.63.255.254
 - Qual è l'indirizzo di broadcast 10.63.255.255
- Qual è il range della seconda sottorete? 10.64.0.0 - 10.127.255.255

Esempio #2: 3° ottetto

- 120.15.1.0/19
- Domande
 - Quante sottoreti?
 - Quanti indirizzi host per sottorete?
 - Quanti validi?
 - Qual è la subnet mask?
 - Qual è il netid della prima sottorete
 - Qual è il primo indirizzo valido? E l'ultimo?
 - Qual è l'indirizzo di broadcast
 - Qual è il range della seconda sottorete?

Esempio #2: 3° ottetto

- 120.15.1.0/19



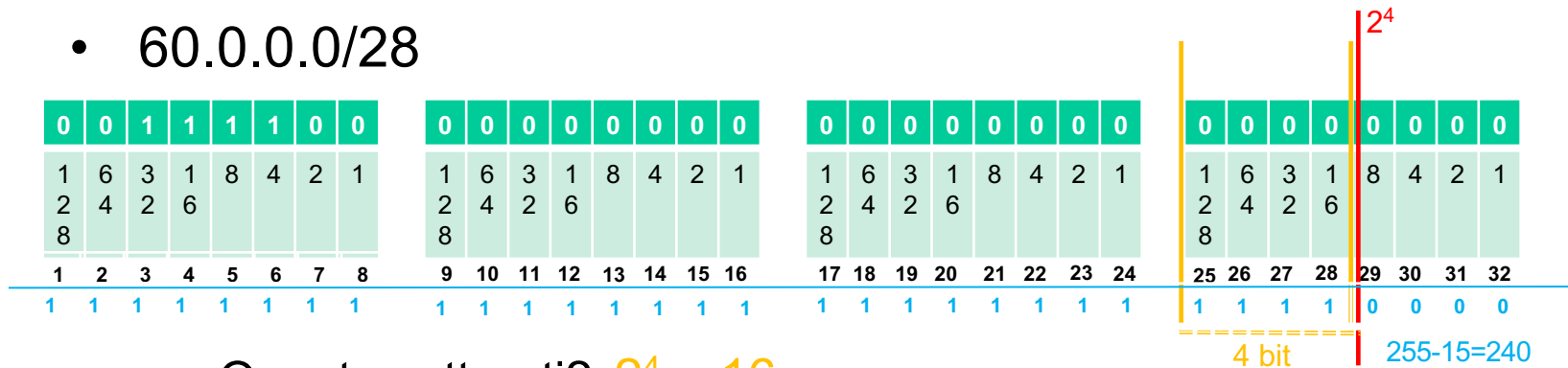
- Quante sottoreti? $2^3 = 8$
- Quanti indirizzi host per sottorete? $2^{13} = 8192$
 - Quanti validi? $2^{13} - 2 = 8190$
- Qual è la subnet mask? 255.255.224.0
- Qual è il netid della prima sottorete 120.15.0.0
 - Qual è il primo indirizzo valido? E l'ultimo? 120.15.0.1 – 120.15.31.254
 - Qual è l'indirizzo di broadcast 120.15.31.255
- Qual è il range della seconda sottorete? 120.15.32.0 - 120.15.63.255

Esempio #3: 4° ottetto

- 60.0.0.0/28
- Domande
 - Quante sottoreti?
 - Quanti indirizzi host per sottorete? Quanti validi?
 - Qual è la subnet mask?
 - Qual è il netid della prima sottorete
 - Qual è il primo indirizzo valido? E l'ultimo?
 - Qual è l'indirizzo di broadcast
 - Qual è il range della seconda sottorete?

Esempio #3: 4° ottetto

- 60.0.0.0/28



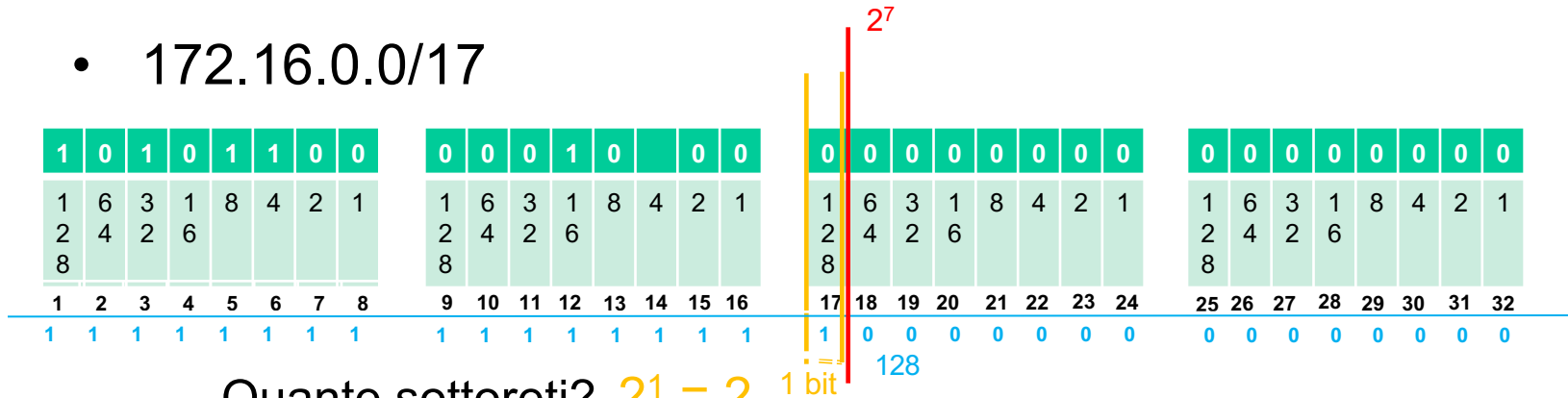
- Quante sottoreti? $2^4 = 16$
- Quanti indirizzi host per sottorete? $2^4 = 16$
 - Quanti validi? $2^4 - 2 = 14$
- Qual è la subnet mask? 255.255.255.240
- Qual è il netid della prima sottorete 60.0.0.0
 - Qual è il primo indirizzo valido? E l'ultimo? 60.0.0.1 – 60.0.0.14
 - Qual è l'indirizzo di broadcast 60.0.0.15
- Qual è il range della seconda sottorete? 60.0.0.16 - 60.0.0.31

Esempio #4: 3° ottetto

- 172.16.0.0/17
- Domande
 - Quante sottoreti?
 - Quanti indirizzi host per sottorete?
 - Quanti validi?
 - Qual è la subnet mask?
 - Qual è il netid della prima sottorete
 - Qual è il primo indirizzo valido? E l'ultimo?
 - Qual è l'indirizzo di broadcast
 - Qual è il netid della seconda sottorete?

Esempio #4: 3° ottetto

- 172.16.0.0/17



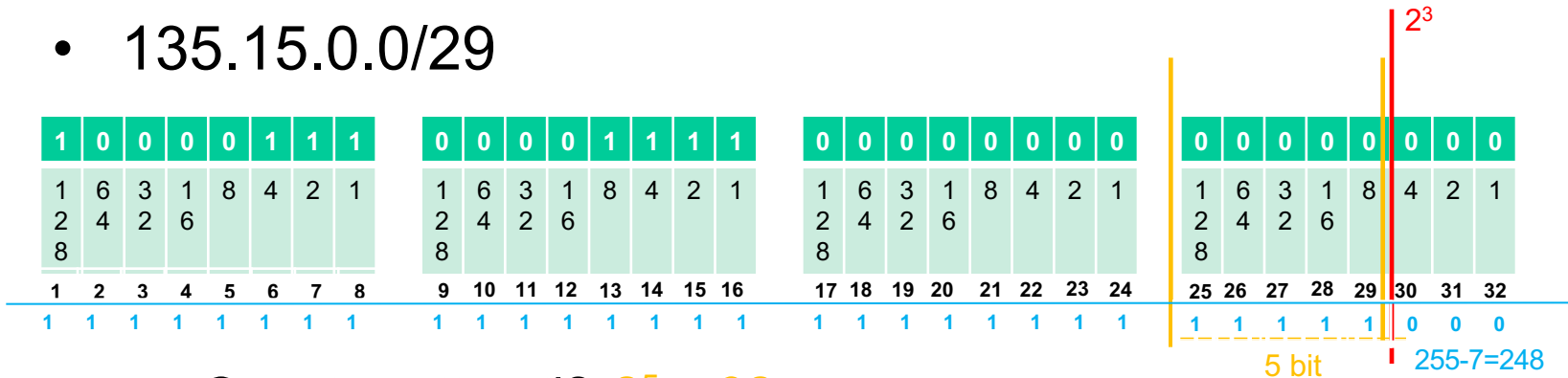
- Quante sottoreti? $2^1 = 2$ 1 bit
- Quanti indirizzi host per sottorete? $2^{15} = 32\,768$
 - Quanti validi? $2^{15} - 2 = 32\,766$
- Qual è la subnet mask? 255.255.128.0
- Qual è il netid della prima sottorete 172.16.0.0
 - Qual è il primo indirizzo valido? E l'ultimo? 172.16.0.1 – 172.16.127.254
 - Qual è l'indirizzo di broadcast 172.16.127.255
- Qual è il netid della seconda sottorete? 172.16.128.0

Esempio #5: 4° ottetto

- 135.15.0.0/29
- Domande
 - Quante sottoreti?
 - Quanti indirizzi host per sottorete?
 - Quanti validi?
 - Qual è la subnet mask?
 - Qual è il netid della prima sottorete
 - Qual è il primo indirizzo valido? E l'ultimo?
 - Qual è l'indirizzo di broadcast
 - Qual è il netid della seconda sottorete?

Esempio #5: 4° ottetto

- 135.15.0.0/29



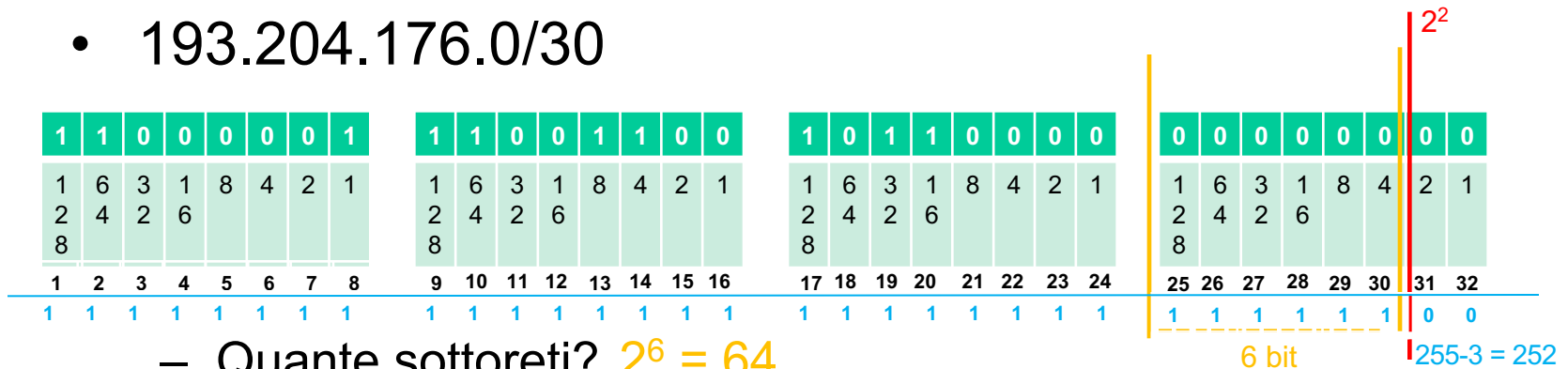
- Quante sottoreti? $2^5 = 32$
- Quanti indirizzi host per sottorete? $2^3 = 8$
 - Quanti validi? $2^3 - 2 = 6$
- Qual è la subnet mask? 255.255.255.248
- Qual è il netid della prima sottorete 135.15.0.0
 - Qual è il primo indirizzo valido? E l'ultimo? 135.15.0.1 - 135.15.0.6
 - Qual è l'indirizzo di broadcast 135.15.0.7
- Qual è il netid della seconda sottorete? 135.15.0.8

Esempio #6: 4° ottetto

- 193.204.176.0/30
- Domande
 - Quante sottoreti?
 - Quanti indirizzi host per sottorete? Quanti validi?
 - Qual è la subnet mask?
 - Qual è il netid della prima sottorete
 - Qual è il primo indirizzo valido? E l'ultimo?
 - Qual è l'indirizzo di broadcast
 - Qual è il netid della seconda sottorete?

Esempio #6: 4° ottetto

- 193.204.176.0/30



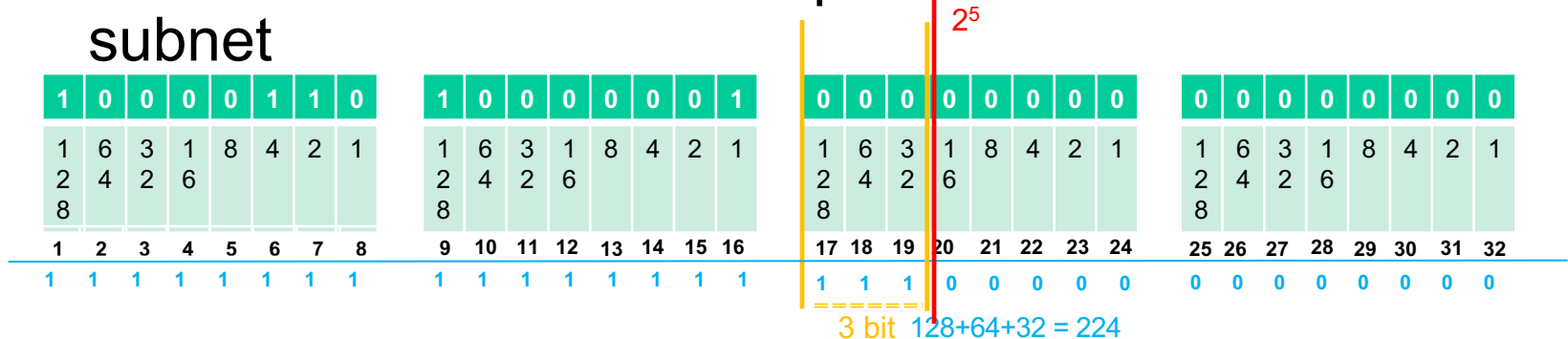
- Quante sottoreti? $2^6 = 64$
- Quanti indirizzi host per sottorete? $2^2 = 4$
 - Quanti validi? $2^2 - 2 = 2$
- Qual è la subnet mask? 255.255.255.252
- Qual è il netid della prima sottorete 193.204.176.0
 - Qual è il primo indirizzo valido? E l'ultimo? 193.204.176.1 - 193.204.176.2
 - Qual è l'indirizzo di broadcast? 193.204.176.3
- Qual è il netid della seconda sottorete? 193.204.176.4

Altro esempio sul subnetting

- Data la rete 134.65.0.0/16 partizionarla in ulteriori 6 subnet
 - Qual è la subnet mask di default per la rete iniziale?
 - Quale subnet mask è necessario usare per la ripartizione?
 - Quali sono i netid delle 6 sottoreti? Quanti gli host validi per rete?

Altro esempio sul subnetting

- Data la rete 134.65.0.0/16 partizionarla in ulteriori 6 subnet



- Qual è la subnet mask di default per la rete iniziale?
134.65.0.0/16 255.255.0.0
- Quale subnet mask è necessario usare per la ripartizione?
134.65.0.0/19 255.255.224.0
- Quali sono i netid delle 6 sottoreti? Quanti gli host validi? $2^{13}-2$
134.65.0.0 134.65.32.0 134.65.64.0
134.65.96.0 134.65.128.0 134.65.160.0

Subnetting – Esercizio 1

- Dato l'indirizzo IP 200.110.12.0 con maschera di sottorete 255.255.255.224, specificare quante sottoreti e quanti host per sottorete si possono ottenere

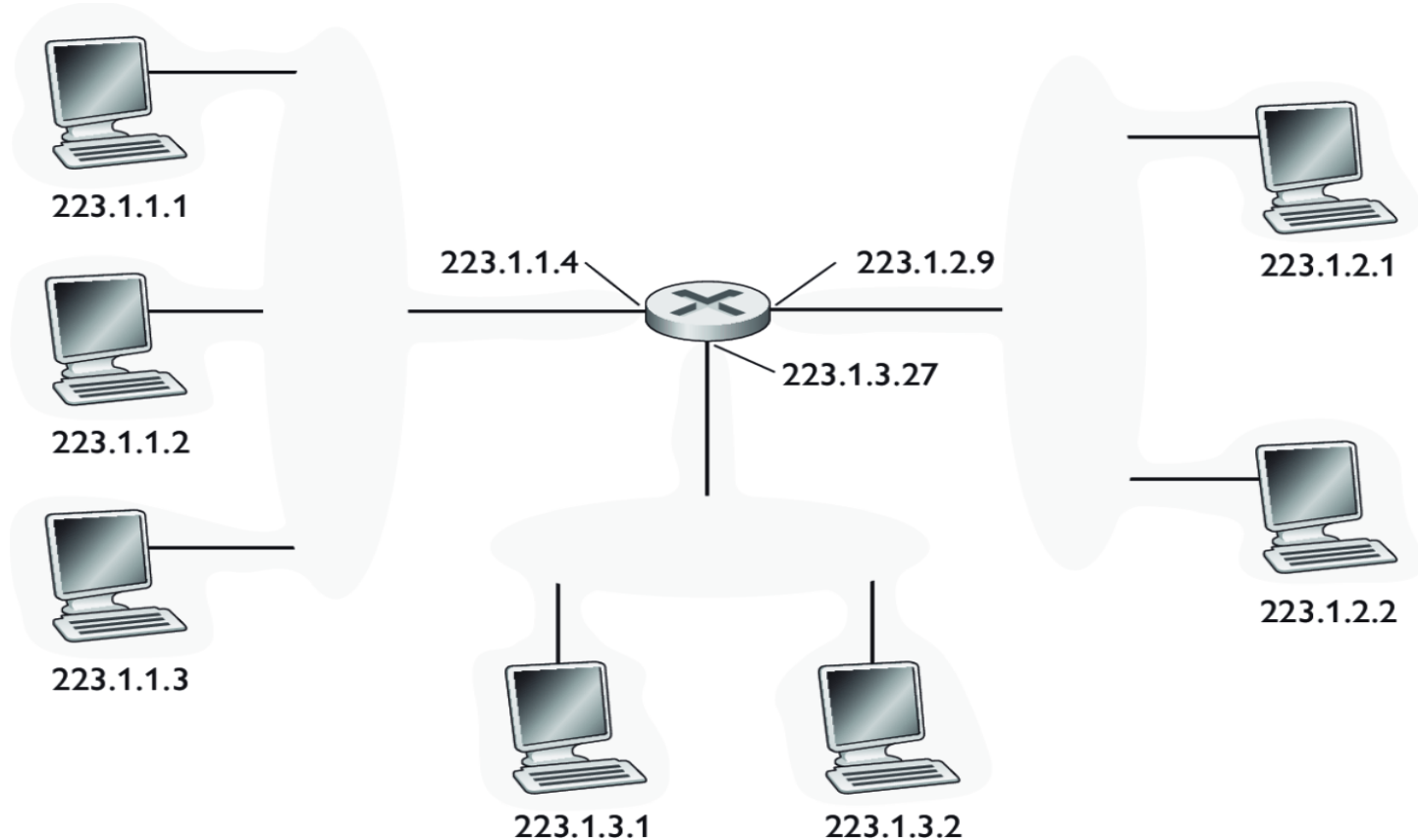
Subnetting – Esercizio 2

- 192.168.23.87/26 e 192.168.23.67/26 appartengono alla stessa sottorete?

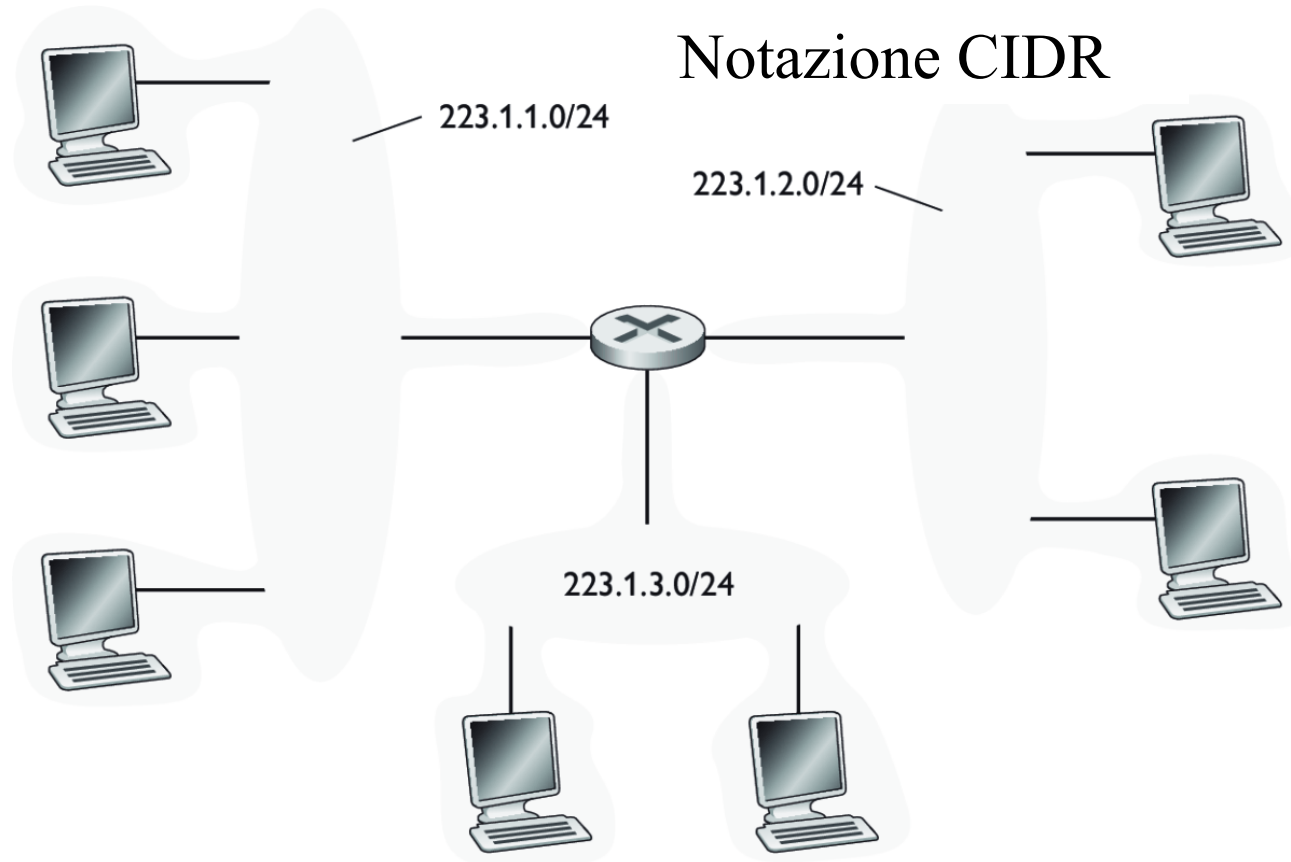
Subnetting - Esercizio 3

- Dato l'indirizzo IP 131.175.21.1/22
- determinare:
 - Il netid di appartenenza
 - la maschera di sottorete
 - l'indirizzo di broadcast
 - il numero di subnet
 - Il numero di host per ciascuna subnet
 - Indirizzo del primo e dell'ultimo host per ogni subnet

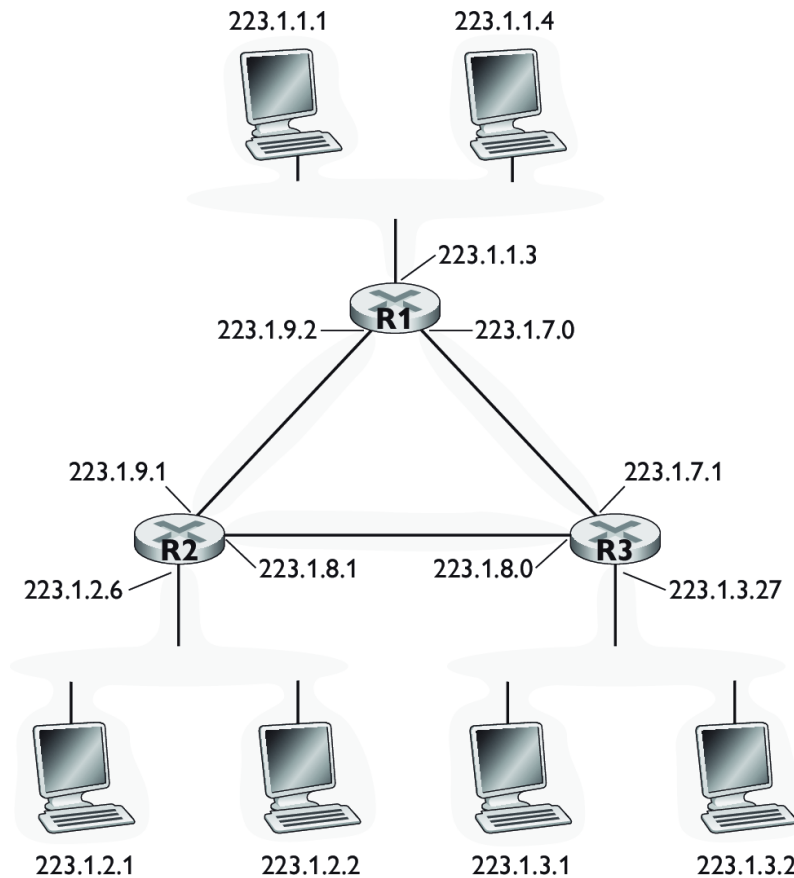
Indirizzi delle interface e sottoreti



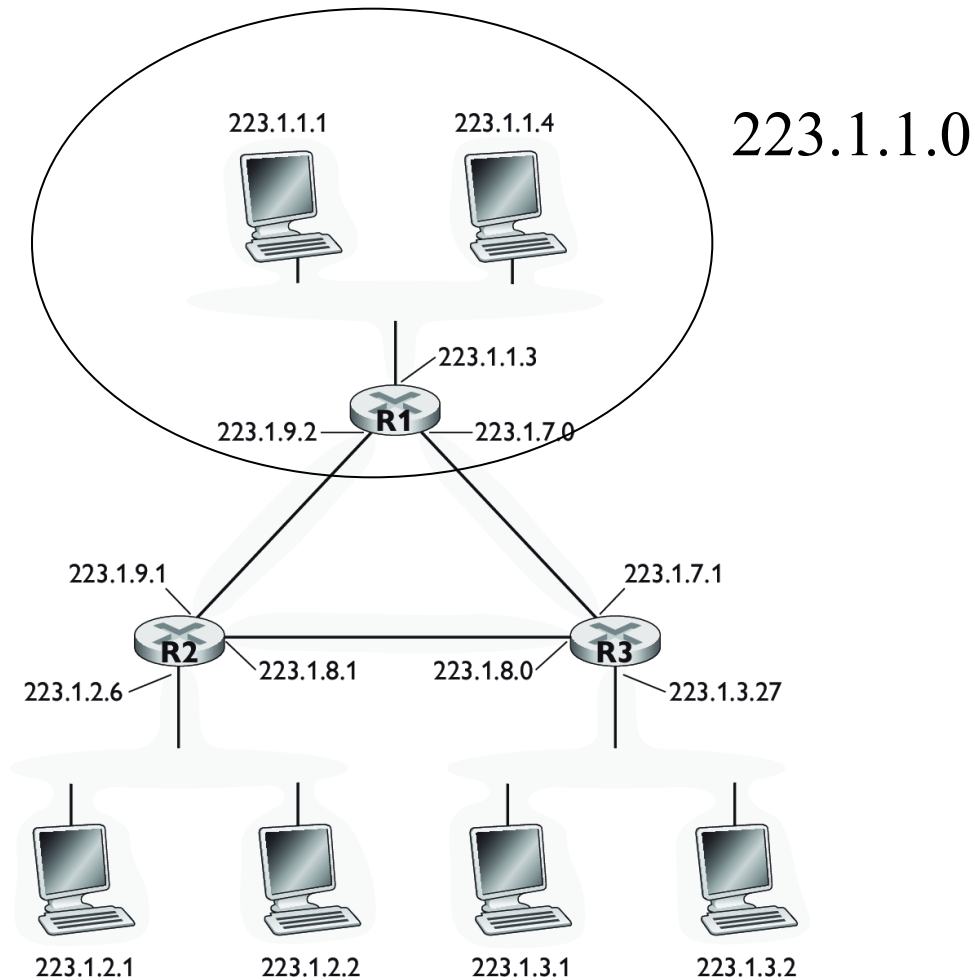
Indirizzi di sottorete



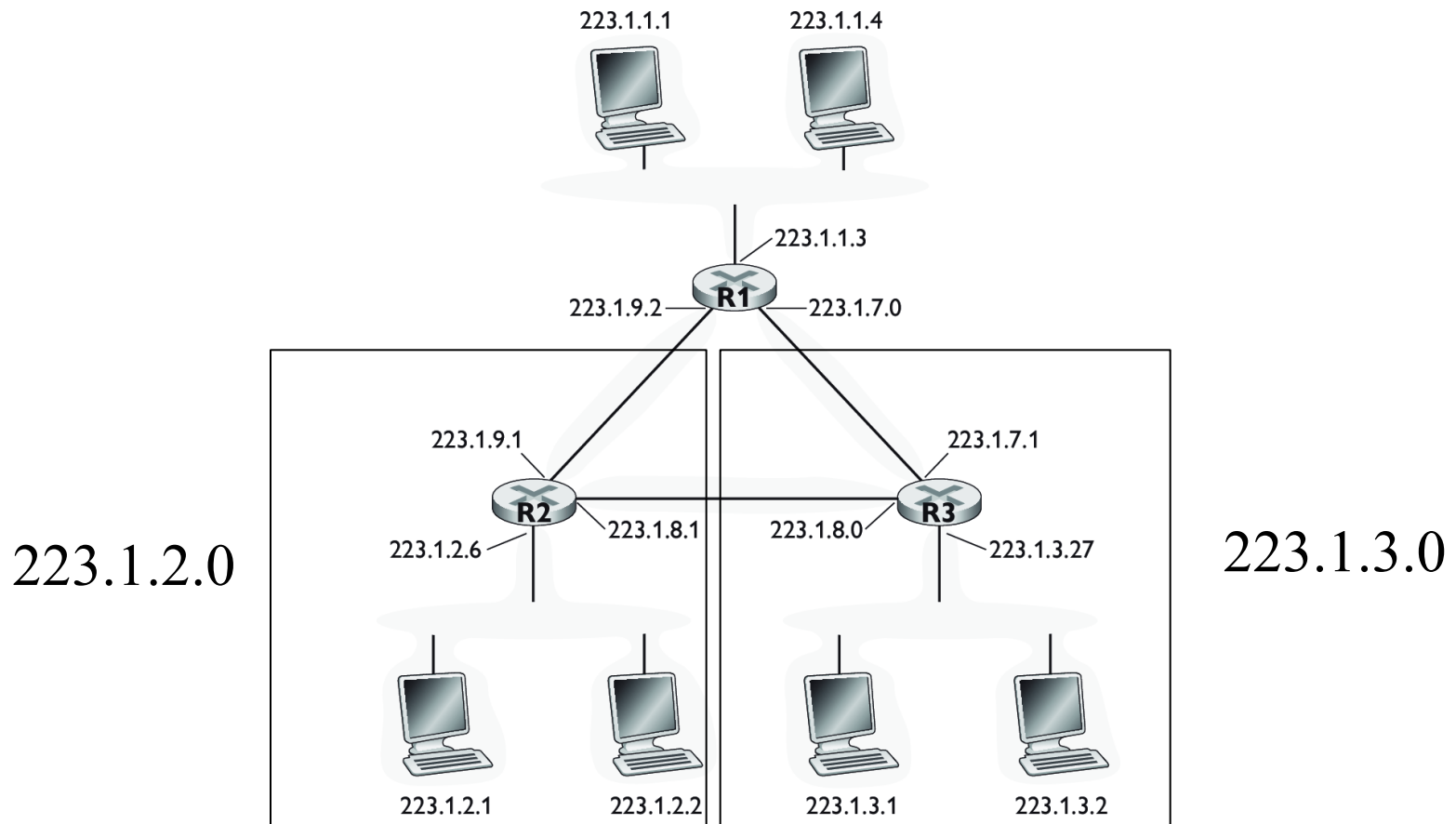
Quante sottoreti?



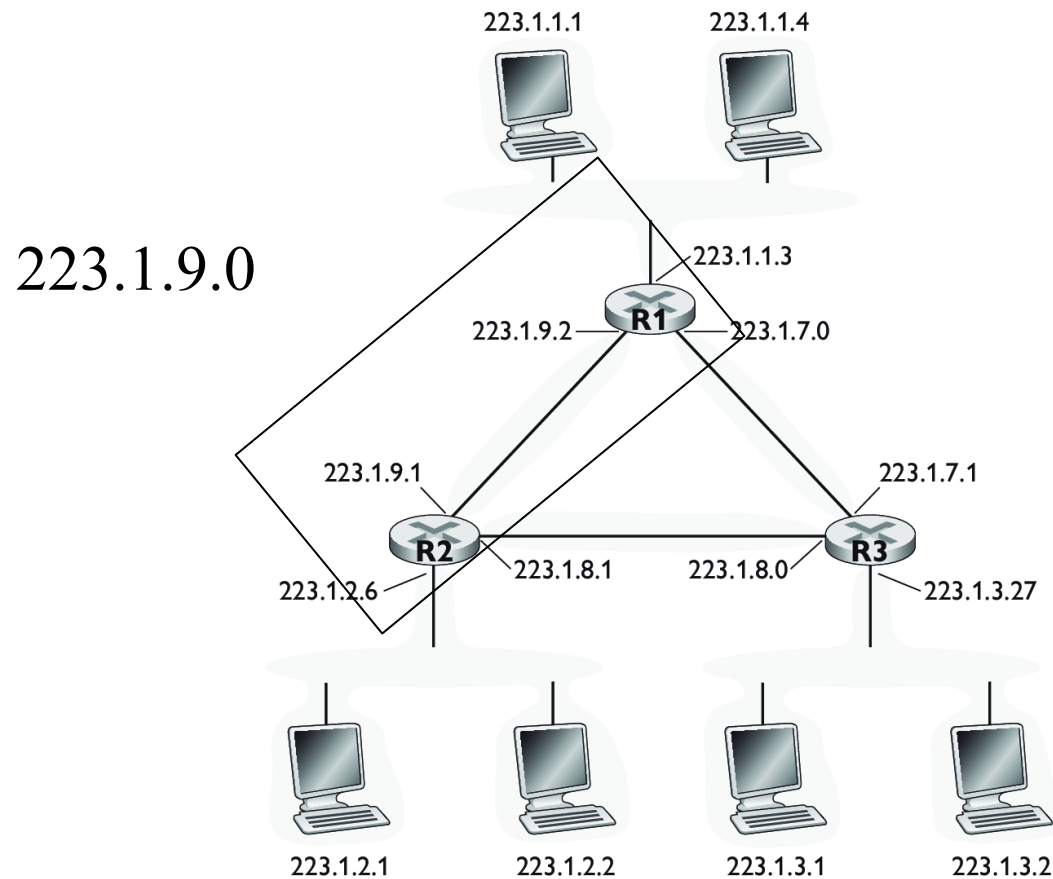
Quante sottoreti?



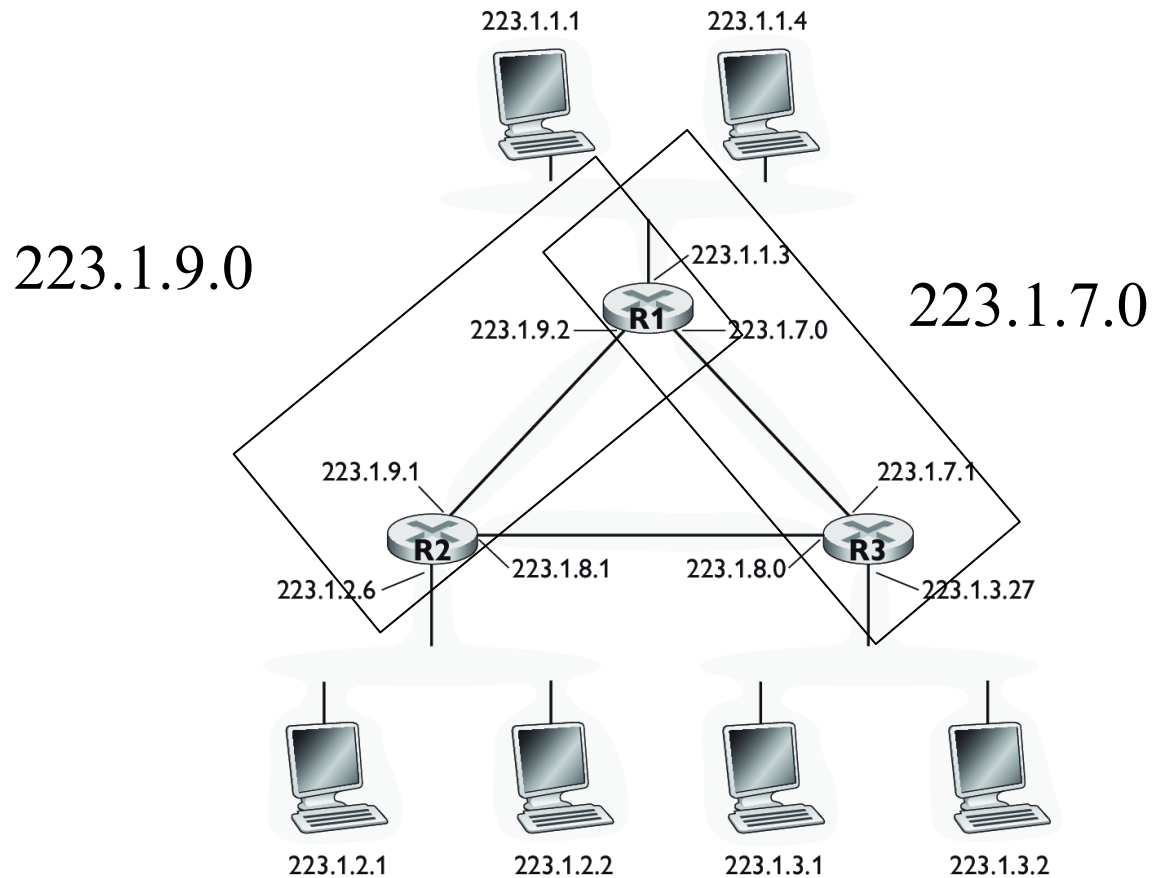
Quante sottoreti?



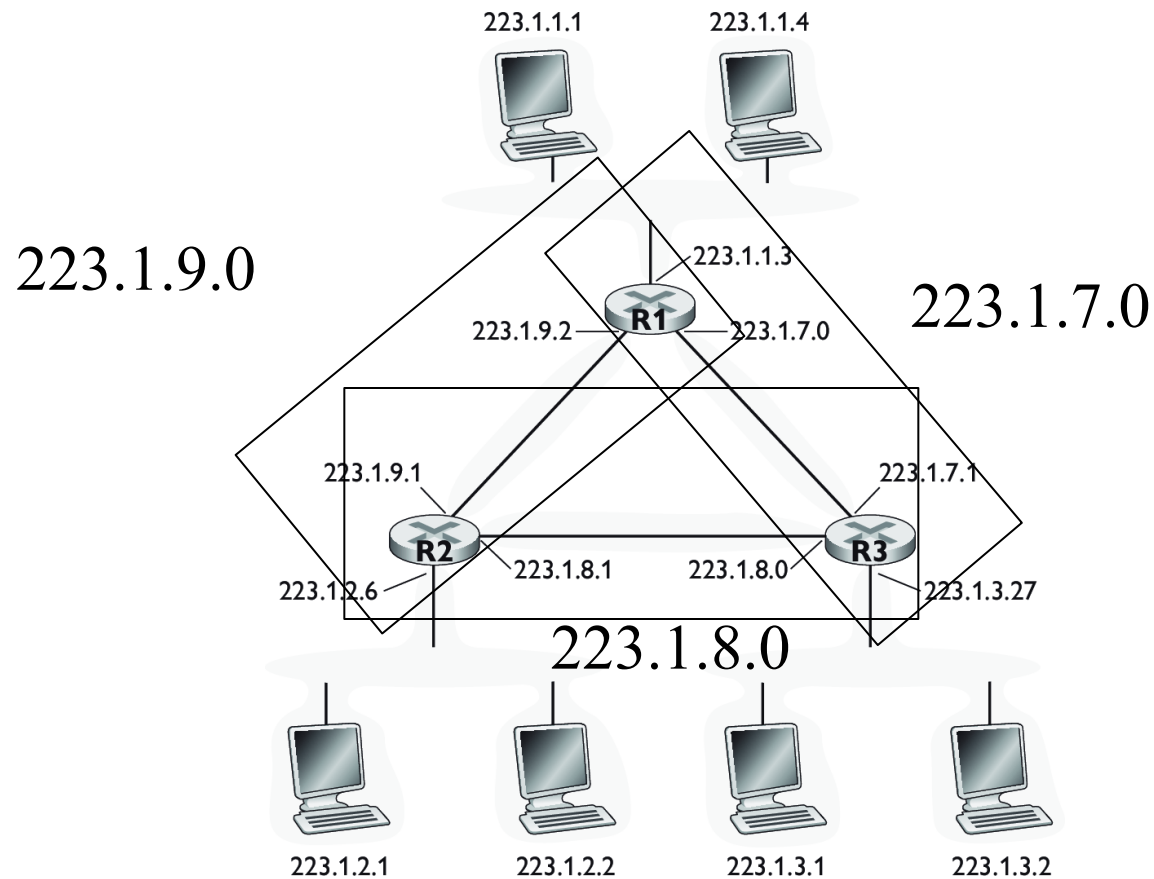
Quante sottoreti?



Quante sottoreti?



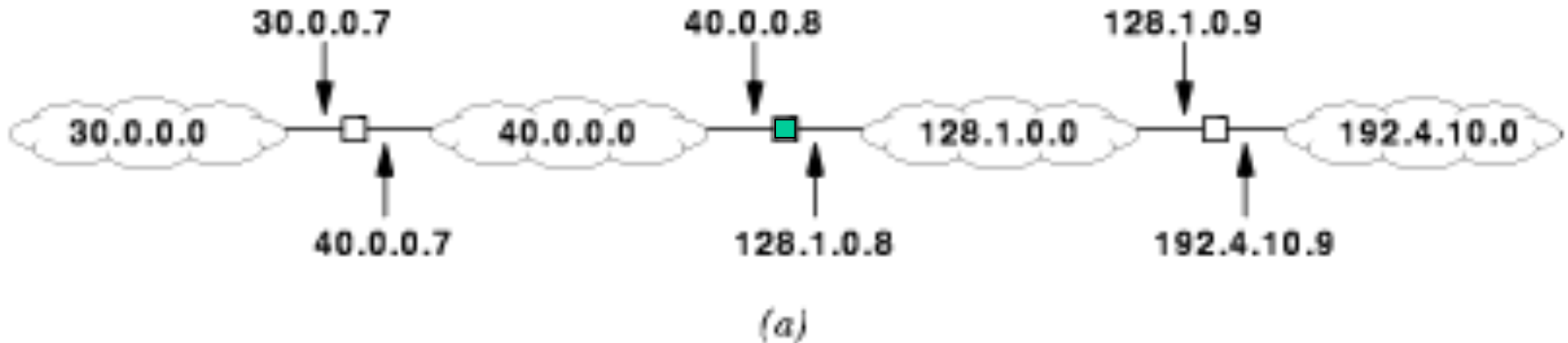
Quante sottoreti?



Supernetting

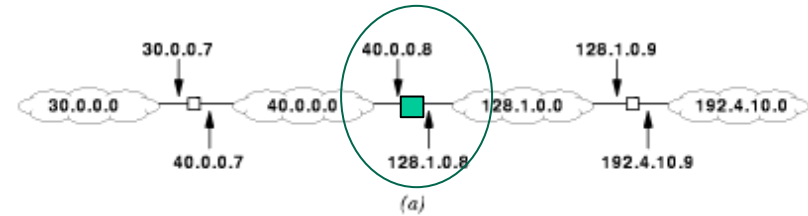
- Super-rete
 - blocco contiguo di reti IP accorpate insieme e indirizzate come fossero un'unica rete IP
 - Tipico per autonomous system (AS, descritti a breve)
- Le tabelle di routing possono contenere un'unica registrazione per la super-rete
 - 193.0.0.0/8 ⇔ 193.0.0.0 - 193.255.255.255
 - invece di
 - 193.0.0.0 /24
 - 193.0.1.0/24
 - ...
 - 193.255.255.0/24

Inoltro dei datagrammi IP



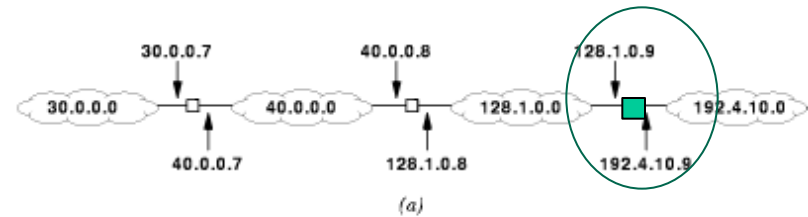
Destination	Mask	Next Hop
30.0.0.0	255.0.0.0	40.0.0.7
40.0.0.0	255.0.0.0	deliver direct
128.1.0.0	255.255.0.0	deliver direct
192.4.10.0	255.255.255.0	128.1.0.9

- Quando arriva un datagramma con indirizzo di destinazione D, il router esegue in loop la seguente istruzione:
 for i=1 to n:
 if ((Mask[i] & D)==Destination[i]) then forward to NextHop[i];



- Il router 128.1.0.8/40.0.0.8 riceve un datagramma per 192.4.10.3
- Il router 128.1.0.8/40.0.0.8 ricava indirizzo di rete
 - $\text{AND}(192.4.10.3, 255.0.0.0) \rightarrow 192.0.0.0$
 - $\text{AND}(192.4.10.3, 255.0.0.0) \rightarrow 192.0.0.0$
 - $\text{AND}(192.4.10.3, 255.255.0.0) \rightarrow 192.4.0.0$
 - $\text{AND}(192.4.10.3, 255.255.255.0) \rightarrow 192.4.10.0$
- Next hop = 128.1.0.9
- Inoltro del datagramma al router 128.1.0.9

Destination	Mask	Next Hop
30.0.0.0	255.0.0.0	40.0.0.7
40.0.0.0	255.0.0.0	deliver direct
128.1.0.0	255.255.0.0	deliver direct
192.4.10.0	255.255.255.0	128.1.0.9

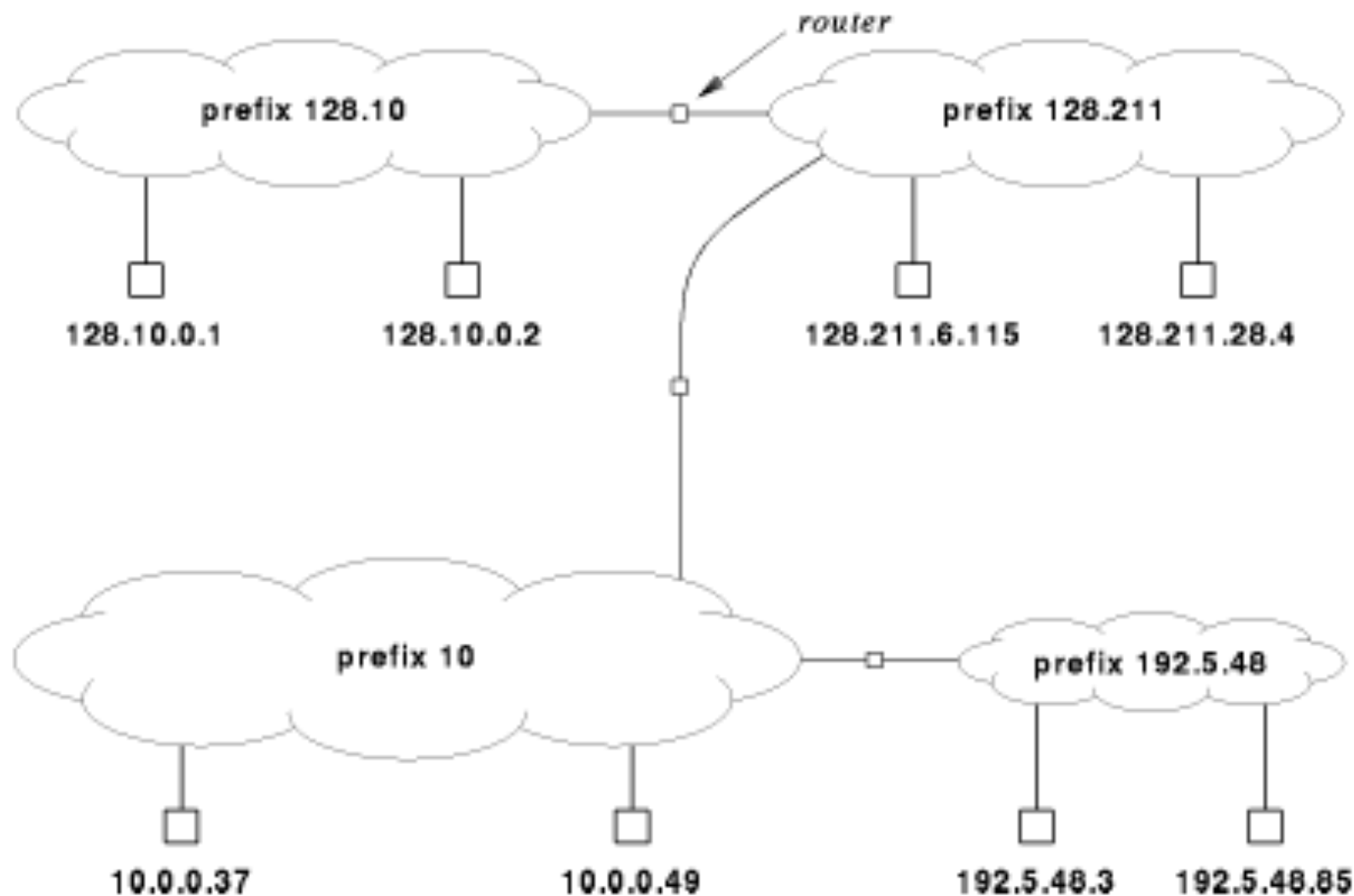


Destination	Mask	Next Hop
30.0.0.0	255.0.0.0	40.0.0.7
40.0.0.0	255.0.0.0	deliver direct
128.1.0.0	255.255.0.0	deliver direct
192.4.10.0	255.255.255.0	128.1.0.9

- Il router 128.1.0.8/40.0.0.8 riceve un datagramma per 192.4.10.3
- Il router 128.1.0.8/40.0.0.8 ricava indirizzo di rete
 - $\text{AND}(192.4.10.3, 255.0.0.0) \rightarrow 192.0.0.0$
 - $\text{AND}(192.4.10.3, 255.0.0.0) \rightarrow 192.0.0.0$
 - $\text{AND}(192.4.10.3, 255.255.0.0) \rightarrow 192.4.0.0$
 - $\text{AND}(192.4.10.3, 255.255.255.0) \rightarrow \mathbf{192.4.10.0}$
- Next hop = 128.1.0.9
- Inoltro del datagramma al router 128.1.0.9
- Router 128.1.0.9/192.4.10.9 riceve il datagramma 192.4.10.3
- In base alla sua routing table, il router ricava indirizzo di rete
 - $\text{AND}(192.4.10.3, 255.0.0.0) \rightarrow 192.0.0.0$
 - $\text{AND}(192.4.10.3, 255.0.0.0) \rightarrow 192.0.0.0$
 - $\text{AND}(192.4.10.3, 255.255.0.0) \rightarrow 192.4.0.0$
 - $\text{AND}(192.4.10.3, 255.255.255.0) \rightarrow \mathbf{192.4.10.0}$

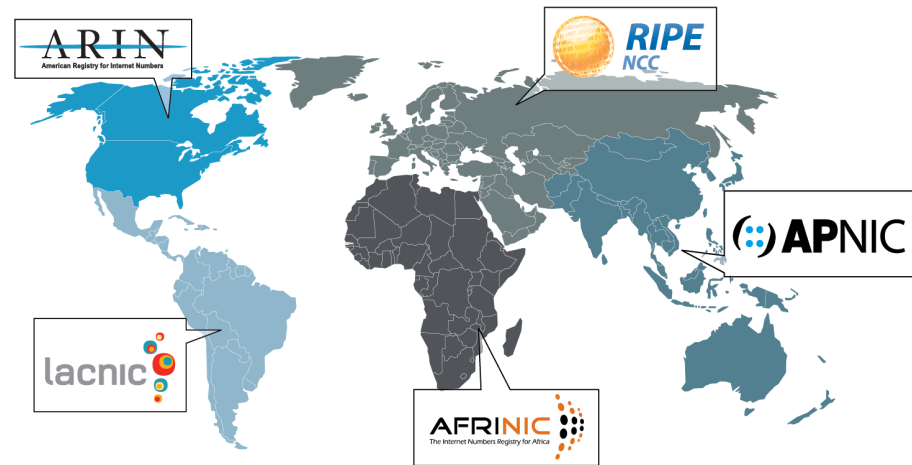
Destination	Mask	Next Hop
30.0.0.0	255.0.0.0	128.1.0.8
40.0.0.0	255.0.0.0	128.1.0.8
128.1.0.0	255.255.0.0	Deliver direct
192.4.10.0	255.255.255.0	Deliver direct

Assegnazione di indirizzi IP in una inter-rete

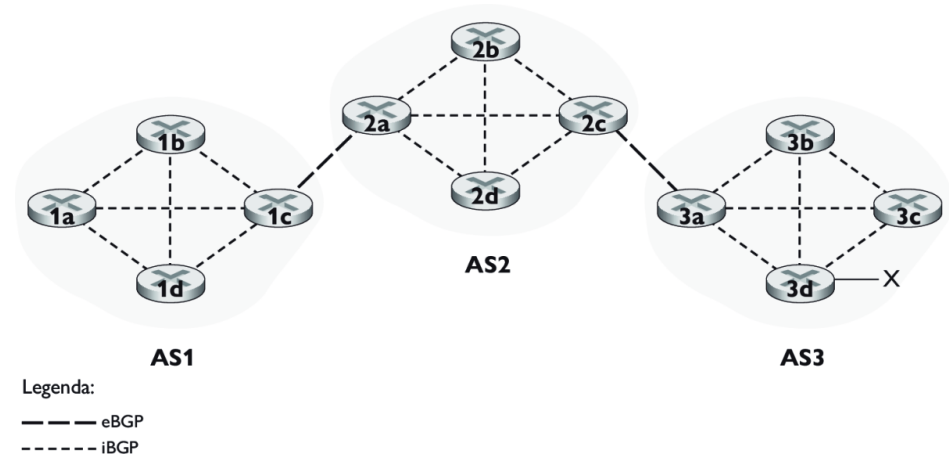
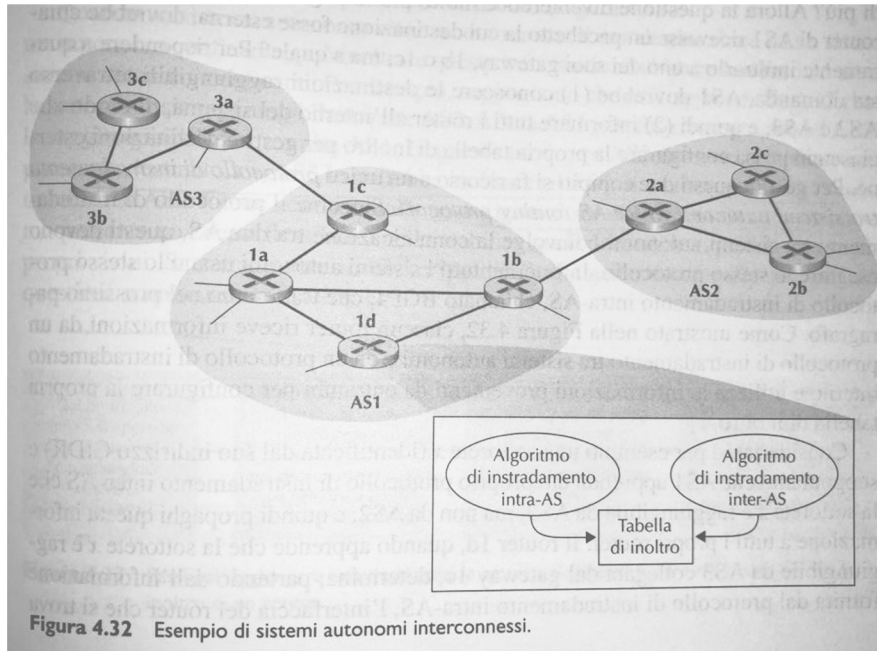


Sistemi autonomi interconnessi

- Sono i domini gestiti da grandi organizzazioni (ISP, governi, grandi compagnie)
- Controllano intervalli di indirizzi IP (non necessariamente contigui o nella stessa nazione)
- Ha un numero univoco (es. 1403)
- Registrati presso un Regional Internet Registry (RIR)
 - AfriNIC (African Network Information Centre)
 - APNIC (Asia Pacific Network Information Centre)
 - ARIN (American Registry for Internet Numbers) –
 - LACNIC (Latin America and Caribbean Network Information Centre)]
 - RIPE NCC (Europa, Medio Oriente e Asia Centrale)



Sessioni BGP

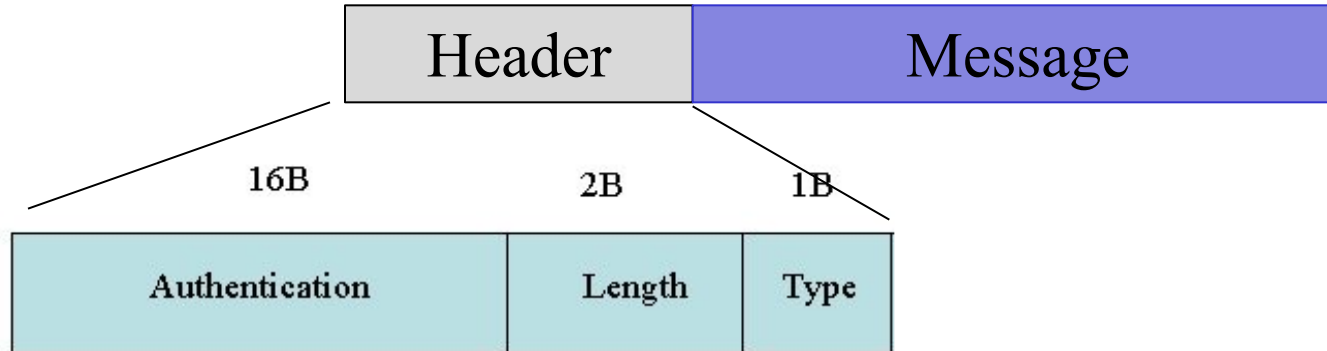


- Connessioni semi-permanenti stabilite sulla porta TCP 179
 - Esterne, tra router BGP in AS diversi
 - Interne, tra router nello stesso AS
- Connessioni BGP \neq collegamento fisico

BGP: Instradamento inter-AS

- Algoritmo basato sul vettore di distanza (decentralizzato, dinamico)
- Consente ai 'router gateway' di frontiera di:
 - Annunciare ad altri AS confinanti l'esistenza del proprio AS e come raggiungerlo
 - Calcolare percorsi ottimi (BGP route) verso altre sottoreti
- Router gateway hanno tabella di routing per l'inoltro all'esterno verso blocchi di indirizzi espressi in notazione CIDR (prefissi di rete)

Formato pacchetto BGP



- Authentication: marker di identificazione
- Length: lunghezza totale del message
- Type: 5 tipi di messaggi (RFC 2918, RFC 1771)
 1. Open: Apertura di una connessione semiperm. con router BGP vicino
 2. Keepalive: Messaggio periodico per mantenimento di connessione
 3. Update: Messaggio fondamentale, vedi slide successiva
 4. Notification: Notifica di errori
 5. Route-Refresh: Invio in bulk di informazioni rotte e risposte in bulk dai router BGP vicini connessi

Formato messaggio di tipo Update



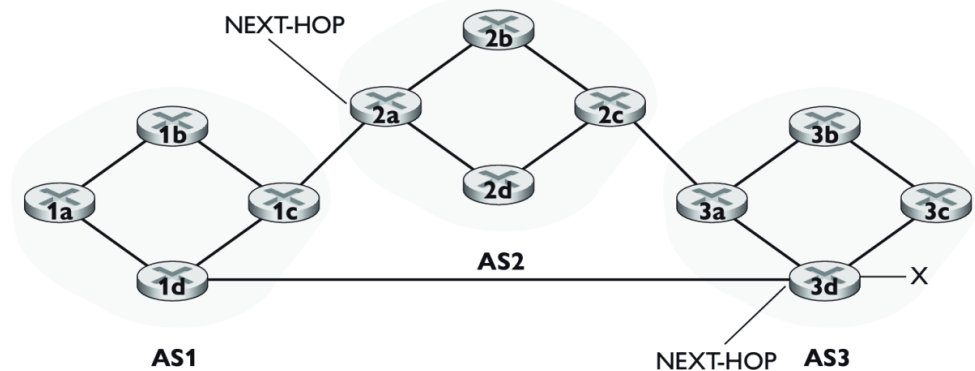
- Responsabile dello scambio di informazioni tra router BGP di AS vicini
 - Unfeasible Routes Len:
 - Numero di rotte da ritirare
 - Withdrawn Routes: elenco rotte da ritirare
 - Total Path Attribute Len:
 - Numero di NLRI
 - Path Attribute
 - AS-PATH: elenco di AS attraversati dal messaggio
 - NEXT-HOP: interfaccia del router che inizia l'AS-PATH
 - Network Layer Reachability Information (NLRI): coppie che specificano in formato CIDR il prefisso (indirizzo di rete) e la lunghezza della maschera (eg. /25) per gli indirizzi raggiungibili

Unfeasible Routes Length (2 bytes)
Withdrawn Routes (variable length)
Total Path Attribute Length (2 bytes)
Path Attribute (variable length)
Network Layer Reachability Information (variable length)

Una rotta (route) BGP è il vettore:
NEXT-HOP, AS-PATH, prefisso rete dest

Selezione rotte migliori

- Rotte possibili
AS1→AS3→x
 1. AS-PATH: (AS2,AS3),
NEXT-HOP: indirizzo IP dell'interfaccia router 2a
 2. AS-PATH: (AS3),
NEXT-HOP: indirizzo IP dell'interfaccia router 3d
- Algoritmo hot potato:
 - Selezione rotta con costo minore per NEXT-HOP
 - Se ci sono risultati multipli, quella con AS-PATH più breve



Incidente Facebook 4 ottobre 2021

- Tutti i servizi Facebook (WhatsApp, Instagram, ...) down per 2-6h nel mondo
- Prime voci indicano che l'AS di Facebook è irraggiungibile
- Cos'è successo?

Our engineering teams have learned that configuration changes on the backbone routers that coordinate network traffic between our data centers caused issues that interrupted this communication. This disruption to network traffic had a cascading effect on the way our data centers communicate, bringing our services to a halt.

<https://engineering.fb.com/2021/10/04/networking-traffic/outage/>

One of the jobs performed by our smaller facilities is to respond to DNS queries. DNS is the address book of the internet, enabling the simple web names we type into browsers to be translated into specific server IP addresses. Those translation queries are answered by our authoritative name servers that occupy well known IP addresses themselves, which in turn are advertised to the rest of the internet via another protocol called the border gateway protocol (BGP).

<https://engineering.fb.com/2021/10/05/networking-traffic/outage-details/>

traceroute -A / traceroute -a / mtr -z

```
Ubuntu 20.04 LTS
fabio@magnus ~ $ traceroute -A facebook.com
traceroute to facebook.com (31.13.86.36), 30 hops max, 60 byte packets
 1 magnus (172.21.160.1) [*] 0.159 ms 0.138 ms 0.128 ms
 2 skyhub4.Home (192.168.0.1) [*] 1.392 ms 1.182 ms 1.376 ms
 3 51.182.160.1 (51.182.160.1) [AS210278] 3.390 ms 3.579 ms 3.571 ms
 4 10.148.255.179 (10.148.255.179) [*] 4.586 ms 10.148.255.175 (10.148.255.175) [*] 4.389 ms 10.148.255.179 (10.148.255.179) [*] 3.815 ms
 5 * * *
 6 ae18.pr02.mxp1.tfbnw.net (157.240.71.244) [AS32934] 20.623 ms 21.173 ms ae89.pr03.mxp1.tfbnw.net (157.240.71.242) [AS32934] 21.124 ms
 7 po131.asw03.mxp1.tfbnw.net (129.134.106.80) [AS33182] 20.933 ms po121.asw04.mxp1.tfbnw.net (129.134.106.142) [AS33182] 19.385 ms po121.asw01.mxp1.tfbnw.net (129.134.51.186) [AS33182] 19.177 ms
 8 po224.psw01.mxp1.tfbnw.net (129.134.106.27) [AS33182] 19.364 ms po242.psw04.mxp1.tfbnw.net (129.134.107.71) [AS33182] 19.925 ms po223.psw02.mxp1.tfbnw.net (129.134.106.37) [AS33182] 21.425 ms
 9 157.240.38.143 (157.240.38.143) [AS32934] 20.713 ms 173.252.67.197 (173.252.67.197) [AS32934] 20.700 ms 157.240.38.95 (157.240.38.95) [AS32934] 20.114 ms
10 edge-star-mini-shv-01-mxp1.facebook.com (31.13.86.36) [AS32934] 20.084 ms 20.072 ms 21.153 ms
fabio@magnus ~ $
```


- AS 210278: Sky Italia
- AS 33182: HostDime.com Inc.
- AS 32934: Facebook in.

AS lookup

Autonomous System – AS32934 – Facebook Inc.

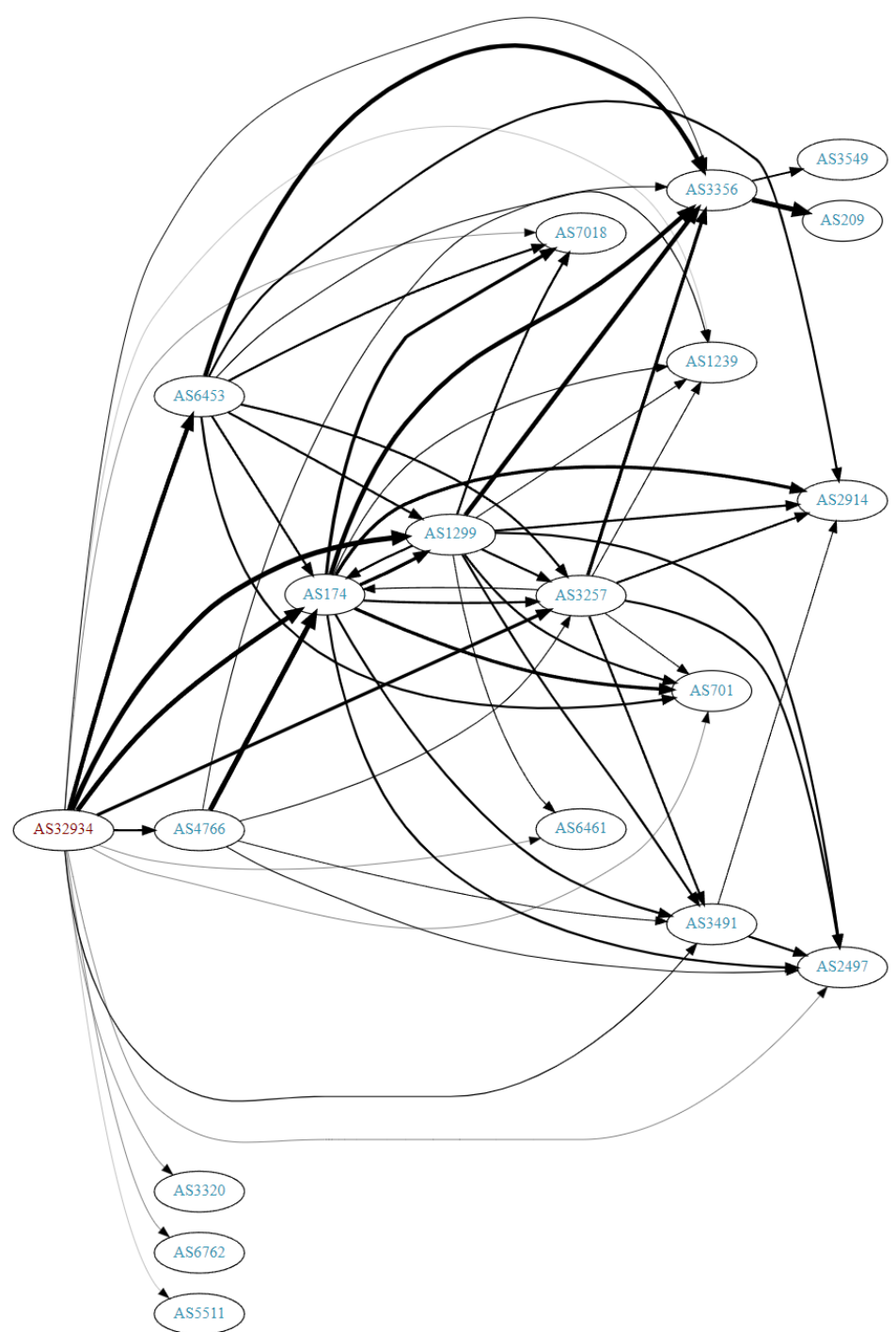


Lookup Result

ASN	AS32934
Organisation	Facebook Inc.
AS Name	FACEBOOK
IPv4 addresses announced	143872
IPv4 prefixes announced	177
Bogon prefixes announced	0
IPv4 space rank	#1,419 out of 73,325
Registry	ARIN
Registered Country	 US
Registration Date	2004-08-24
Registration Change	2012-02-24

BGPview

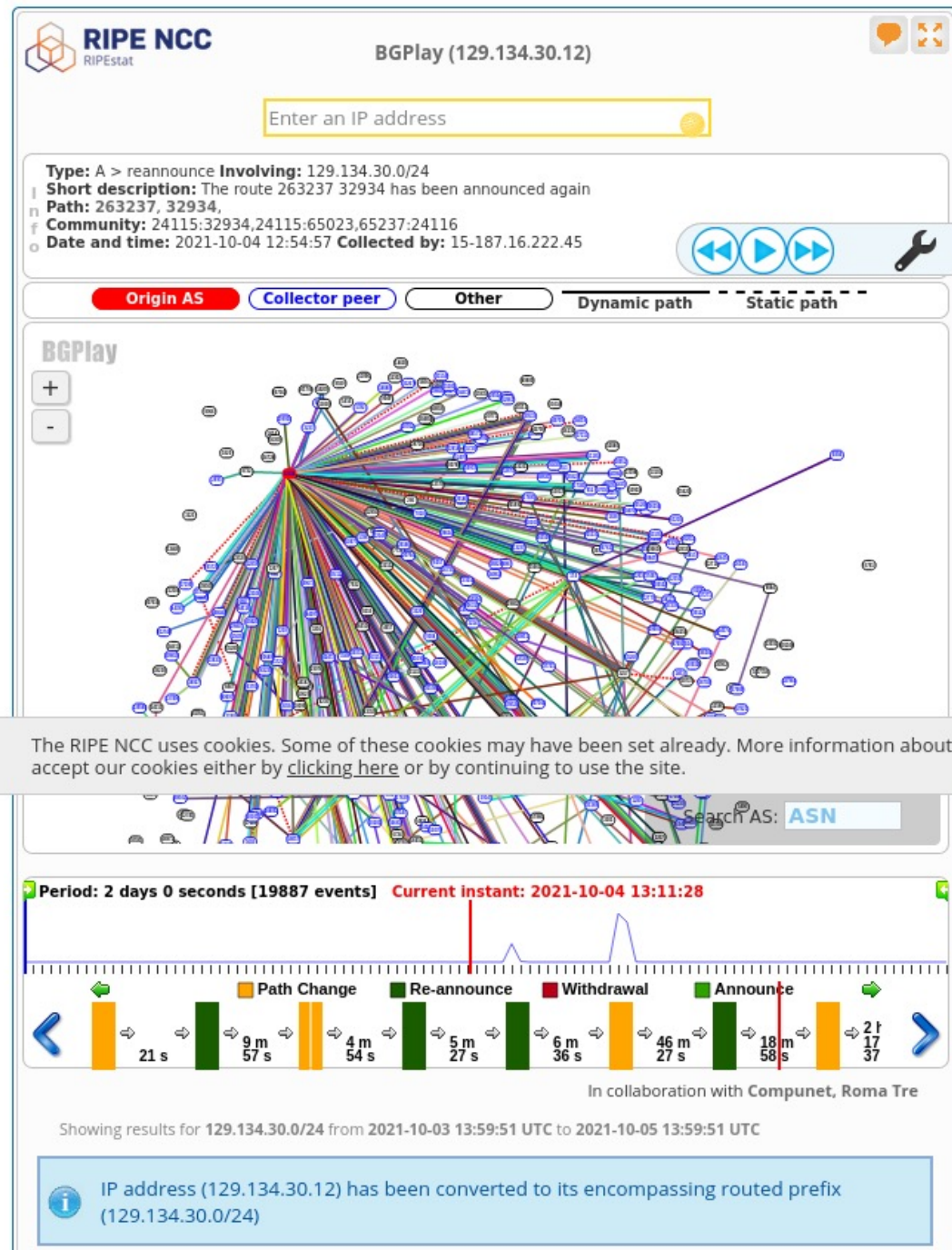
Connessioni dell'AS
Facebook con altri
AS



BGP routing

Verso IP 129.134.30.12
h. ~13.00 UTC

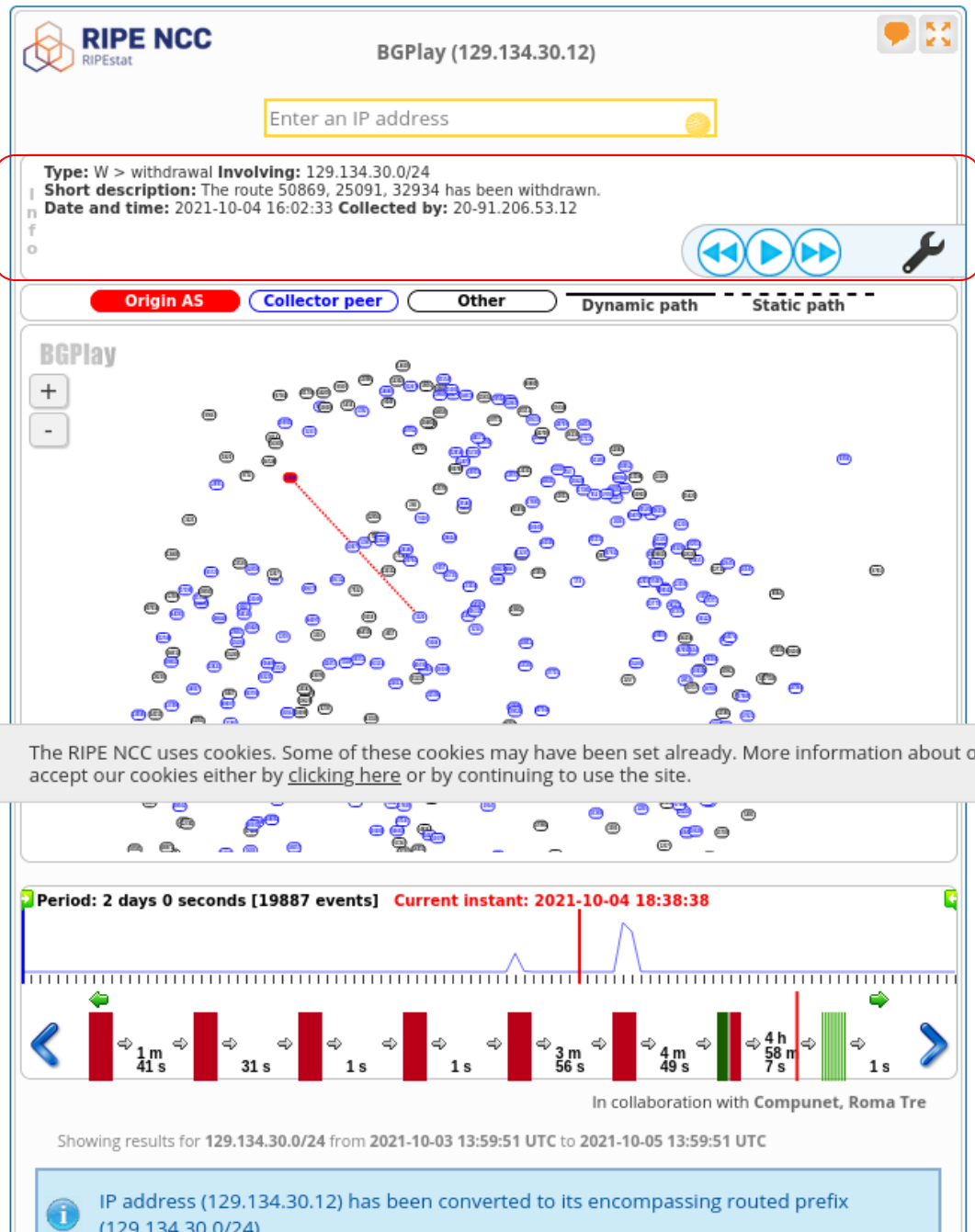
Stato: normal



BGP routing

Verso IP 129.134.30.12
h. ~18.00 UTC

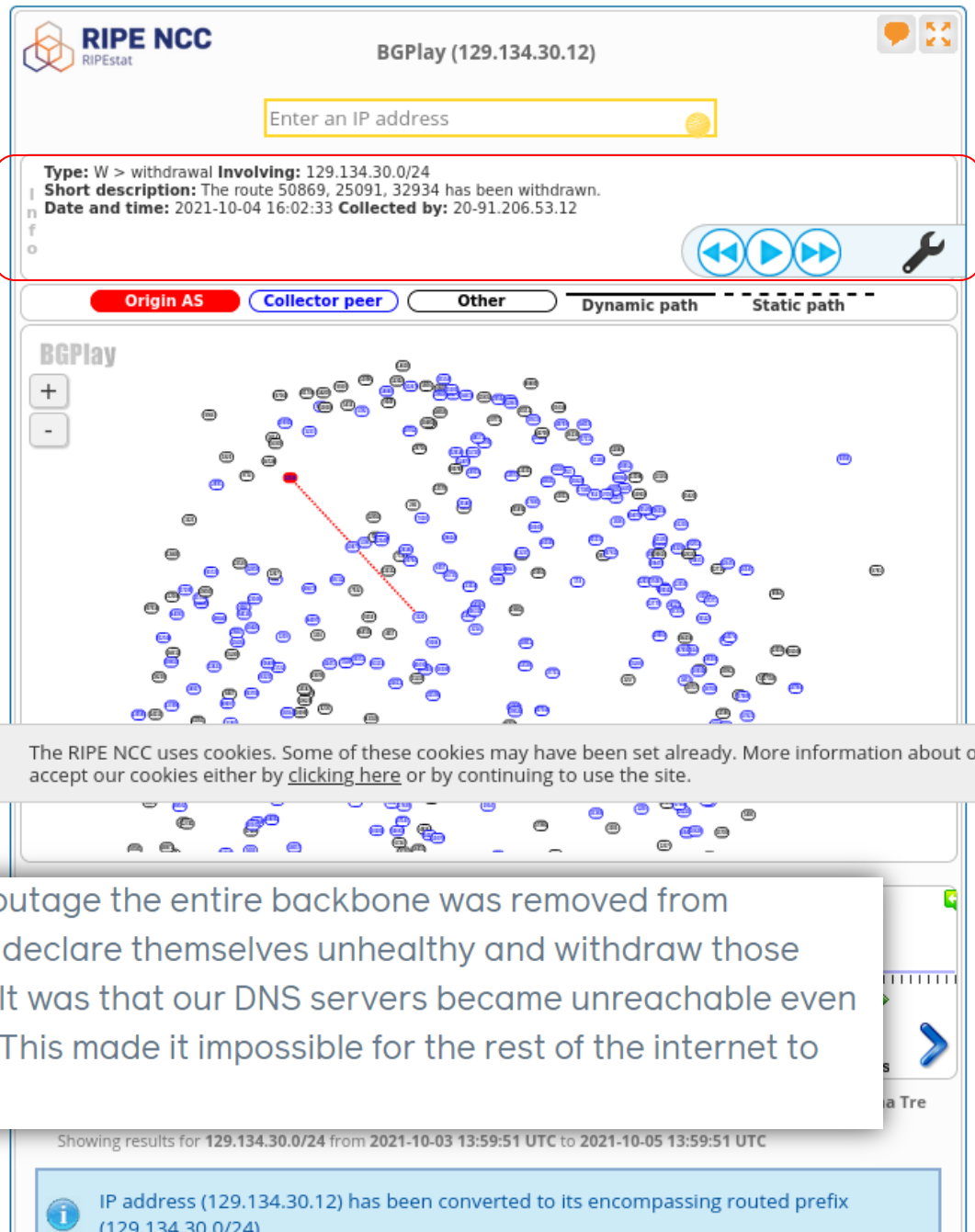
Stato: outage



BGP routing

Verso IP 129.134.30.12
h. ~18.00 UTC

Stato: outage

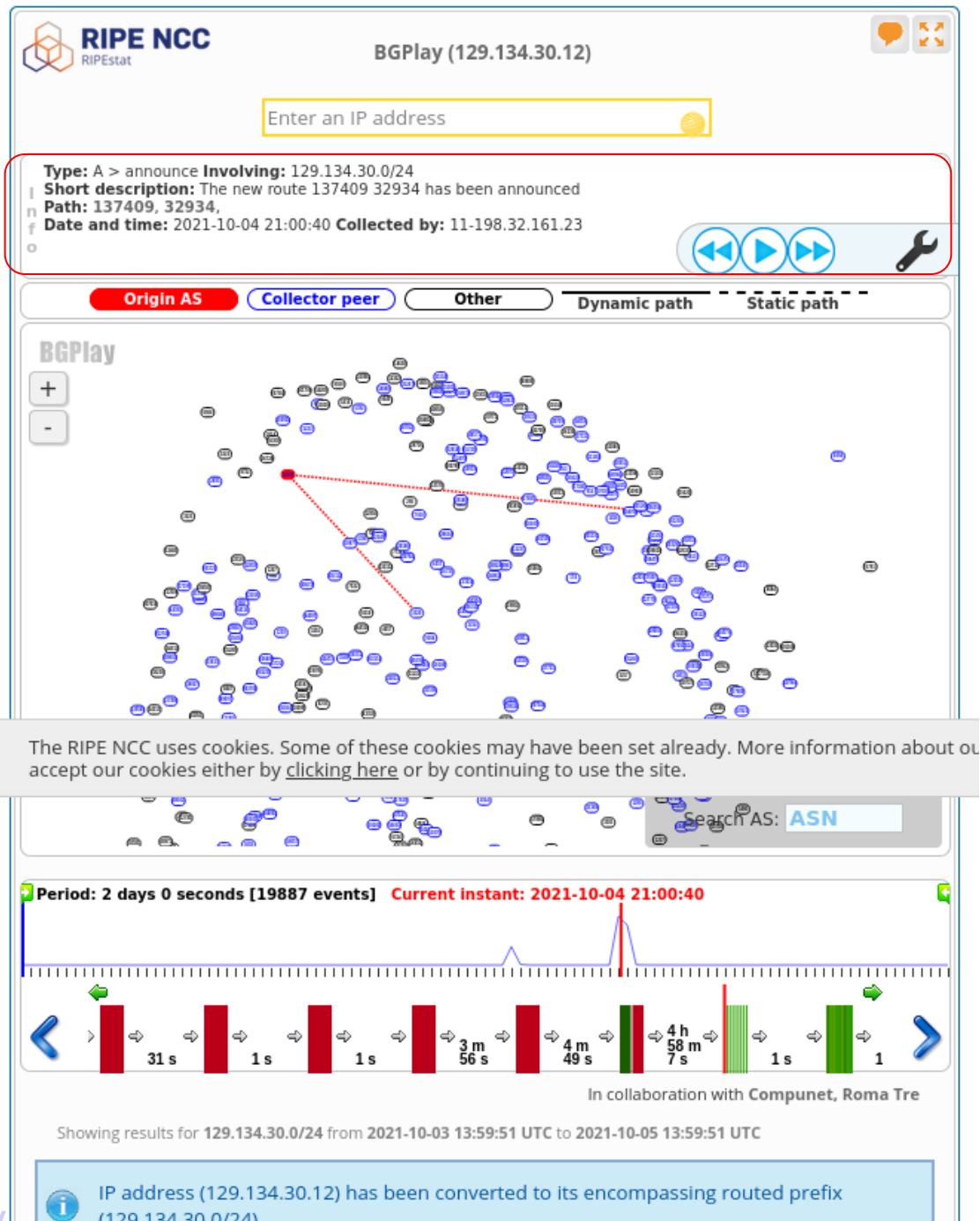


network connection. In the recent outage the entire backbone was removed from operation, making these locations declare themselves unhealthy and withdraw those BGP advertisements. The end result was that our DNS servers became unreachable even though they were still operational. This made it impossible for the rest of the internet to find our servers.

BGP routing

Verso IP 129.134.30.12
h. ~21.00 UTC

Stato: resuming



DHCP

Dynamic Host Configuration Protocol

- permette a un host di ottenere un indirizzo IP in modo automatico

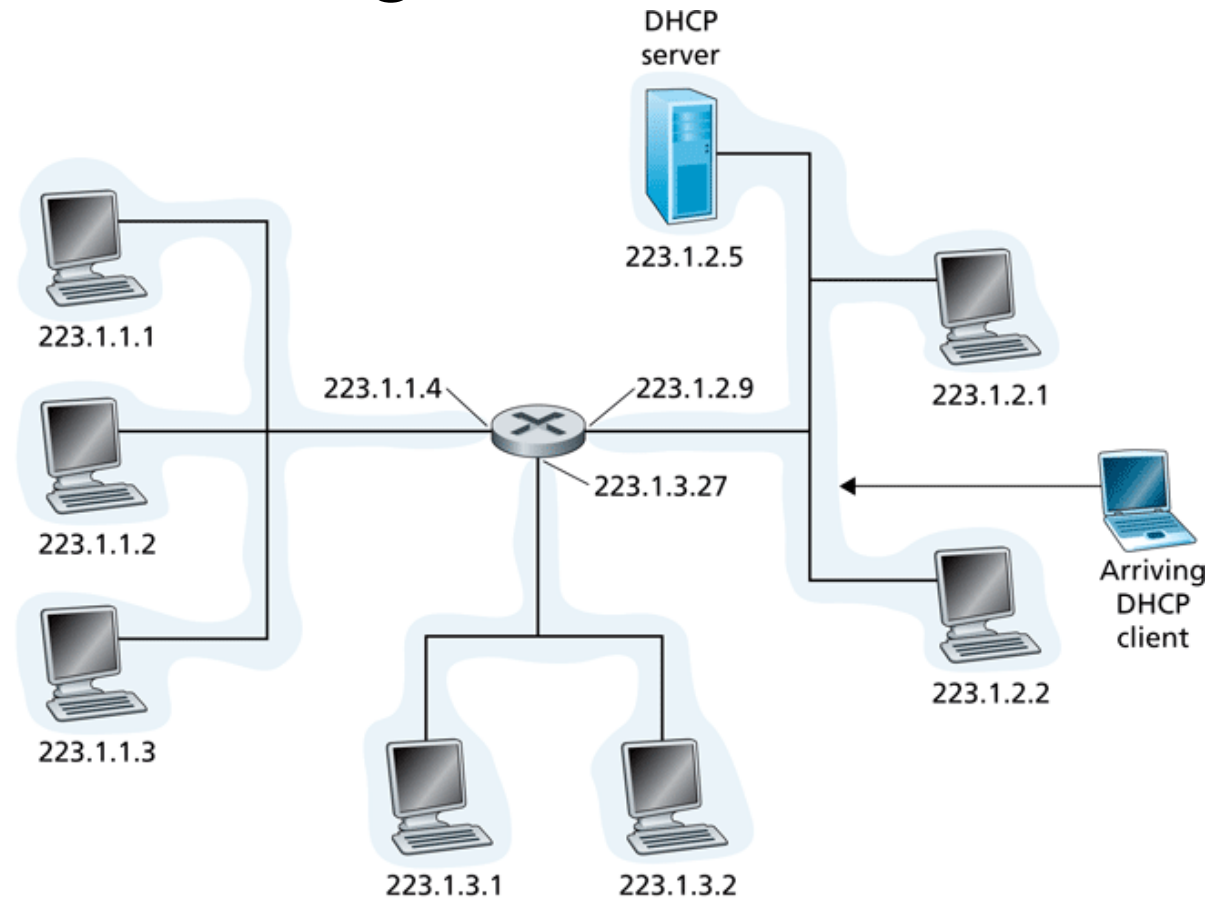
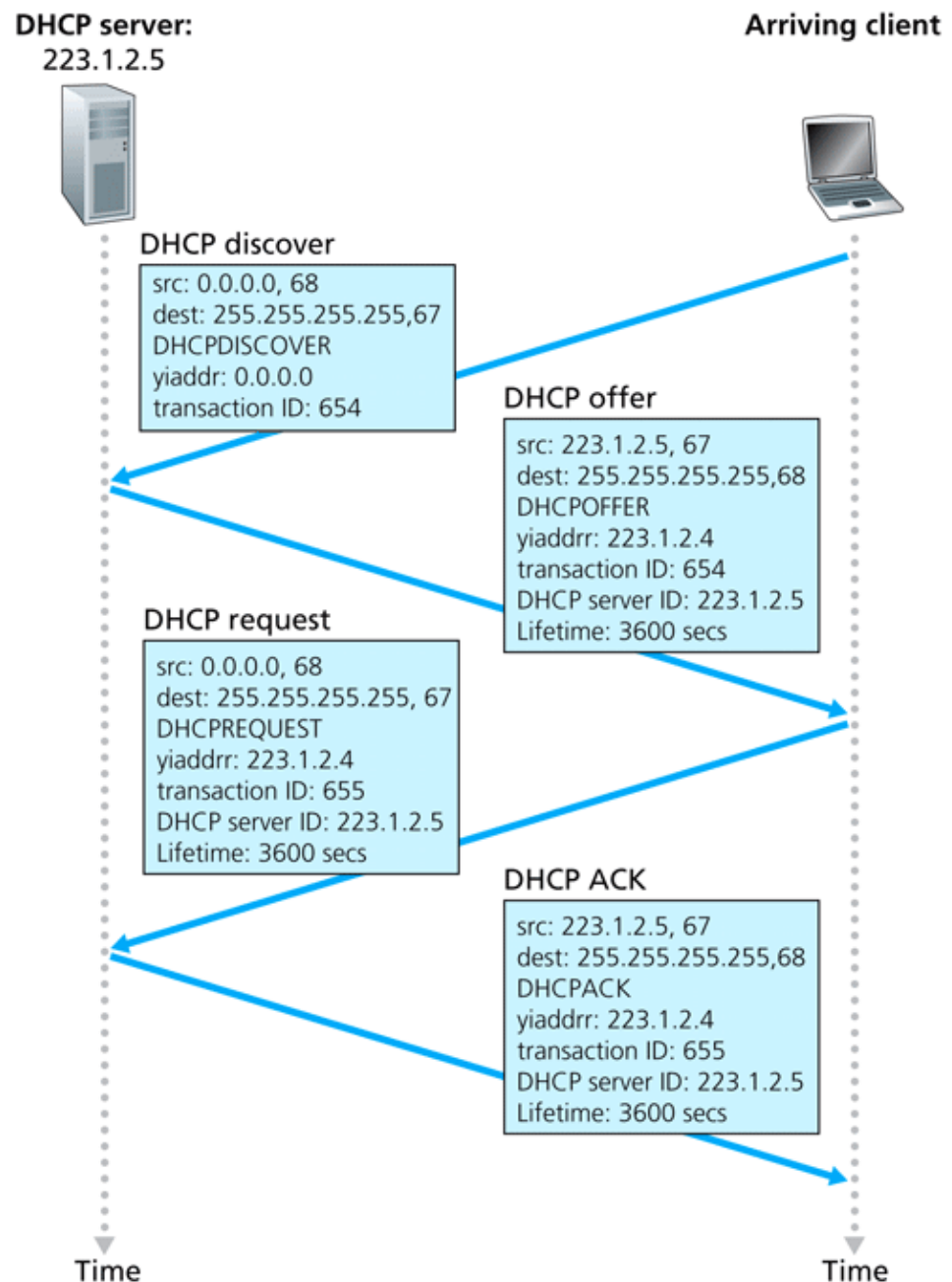


Figure 4.20 ♦ DHCP client-server scenario

Scenario DHCP



Prof. Fabio Calefato

Figure 4.21 ♦ DHCP client-server interaction

Indirizzi IP privati

- Utilizzabili per inter-reti private, ovvero reti private che usano l'Internet suite ma non sono interconnesse ad Internet

Class	Network Address Range
A	10.0.0.0 - 10.255.255.255 (10.0.0.0/8)
B	172.16.0.0 – 172.31.255.255 (172.16.0.0/12)
C	192.168.0.0-192.168.255.255 (192.168.0.0/16)

Traduzione degli indirizzi di rete (NAT)

Una rete con router NAT appare con un unico indirizzo IP pubblico

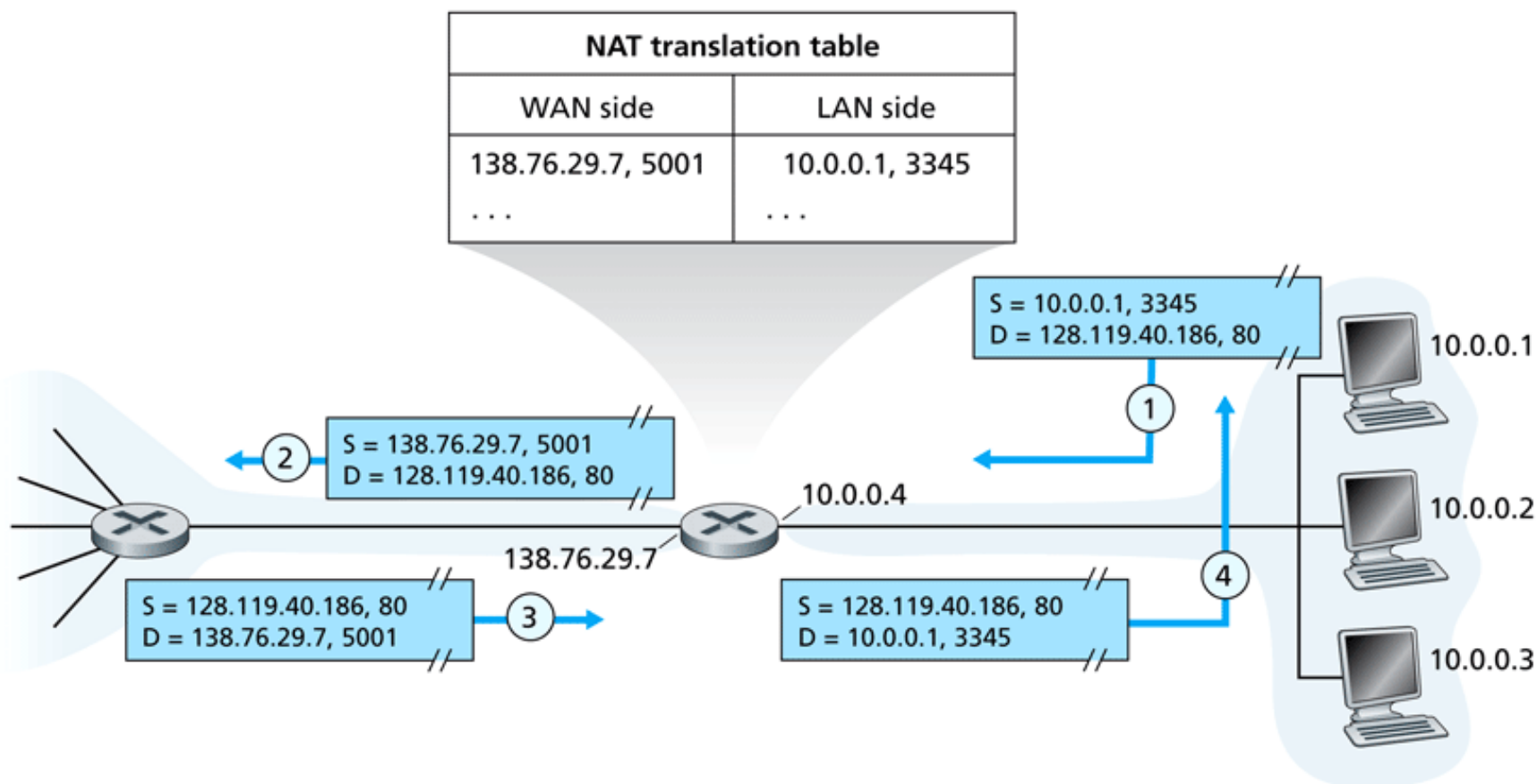


Figure 4.22 ♦ Network address translation

Advanced Port Forwarding Rules

Application Rules

MAC Filtering

ACL filter

Traffic Control

Firewall & DMZ

Advanced Wireless Settings

Advanced Network Settings

Routing

Logout

ADVANCED PORT FORWARDING RULES

The Advanced Port Forwarding option allows you to define a single public port on your router for redirection to an internal LAN IP Address and Private LAN port if required. This feature is useful for hosting online service such as FTP or Web Servers.

Save Settings

Don't Save Settings

24--ADVANCED PORT FORWARDING RULES

Remaining number of rules that can be created : 24

			Port	Traffic Type
<input checked="" type="checkbox"/>	<div>Name</div> <div>FTP</div>	<div><<</div> <div>Application Name</div>	<div>Public Port</div> <div>21 ~ 21</div>	<div>TCP</div>
	<div>IP Address</div> <div>192.168.0.100</div>	<div><<</div> <div>Computer Name</div>	<div>Private Port</div> <div>21 ~ 21</div>	
<input type="checkbox"/>	<div>Name</div> <div></div>	<div><<</div> <div>Application Name</div>	<div>Public Port</div> <div> ~ </div>	<div>Any</div>
	<div>IP Address</div> <div></div>	<div><<</div> <div>Computer Name</div>	<div>Private Port</div> <div> ~ </div>	
<input type="checkbox"/>	<div>Name</div> <div></div>	<div><<</div> <div>Application Name</div>	<div>Public Port</div> <div> ~ </div>	<div>Any</div>
	<div>IP Address</div> <div></div>		<div>Private Port</div> <div></div>	

Helpful Hints...

Click the **Application Name** drop down menu for a list of predefined server types. If you select one of the predefined server types, click the arrow button next to the drop down menu to fill out the corresponding field.

[More...](#)



TP-Link Wireless N Router WR840N

Model No. TL-WR840N

Status

Quick Setup

Network

Wireless

Guest Network

DHCP

Forwarding

- Virtual Server

- Port Triggering

- DMZ

- UPnP

Security

Parental Controls

Access Control

Advanced Routing

Bandwidth Control

IP & MAC Binding

Dynamic DNS

Virtual Server

<input type="checkbox"/>	Service Port	IP Address	Internal Port	Protocol	Status	Edit
<input type="checkbox"/>	455	192.168.0.100	455	TCP or UDP	Enabled	Edit
<input type="checkbox"/>	456	192.168.0.102	456	TCP or UDP	Enabled	Edit

Add New

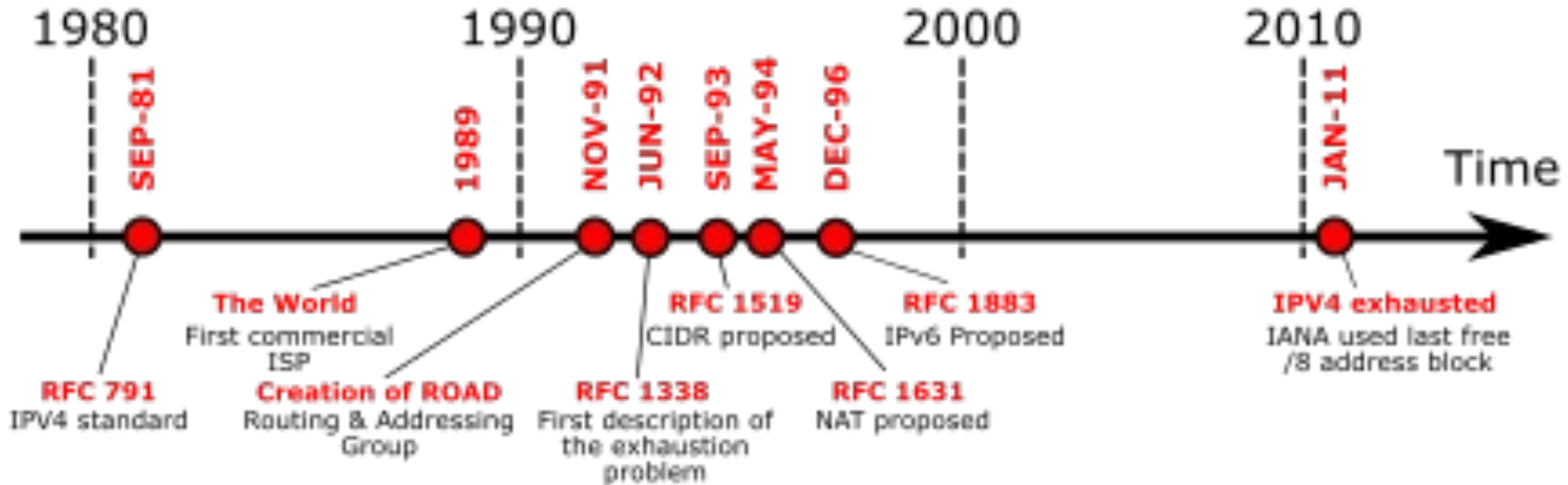
Enable Selected

Disable Selected

Delete Selected

Refresh

Timeline esaurimento indirizzi IPv4



https://en.wikipedia.org/wiki/IPv4_address_exhaustion

IPv6

- 1992: IETF crea un gruppo per risolvere l'emergenza degli indirizzi IP in esaurimento
 - » <http://www.ipv6forum.com/>
- Altre motivazioni
 - Efficienza (header a lunghezza fissa, no checksum)
 - Supporto alla sicurezza
 - Supporto alla QoS
 - Supporto alla mobilità
 - ...
- Oggi: IPv6 è implementato in tutti i router e host attualmente prodotti

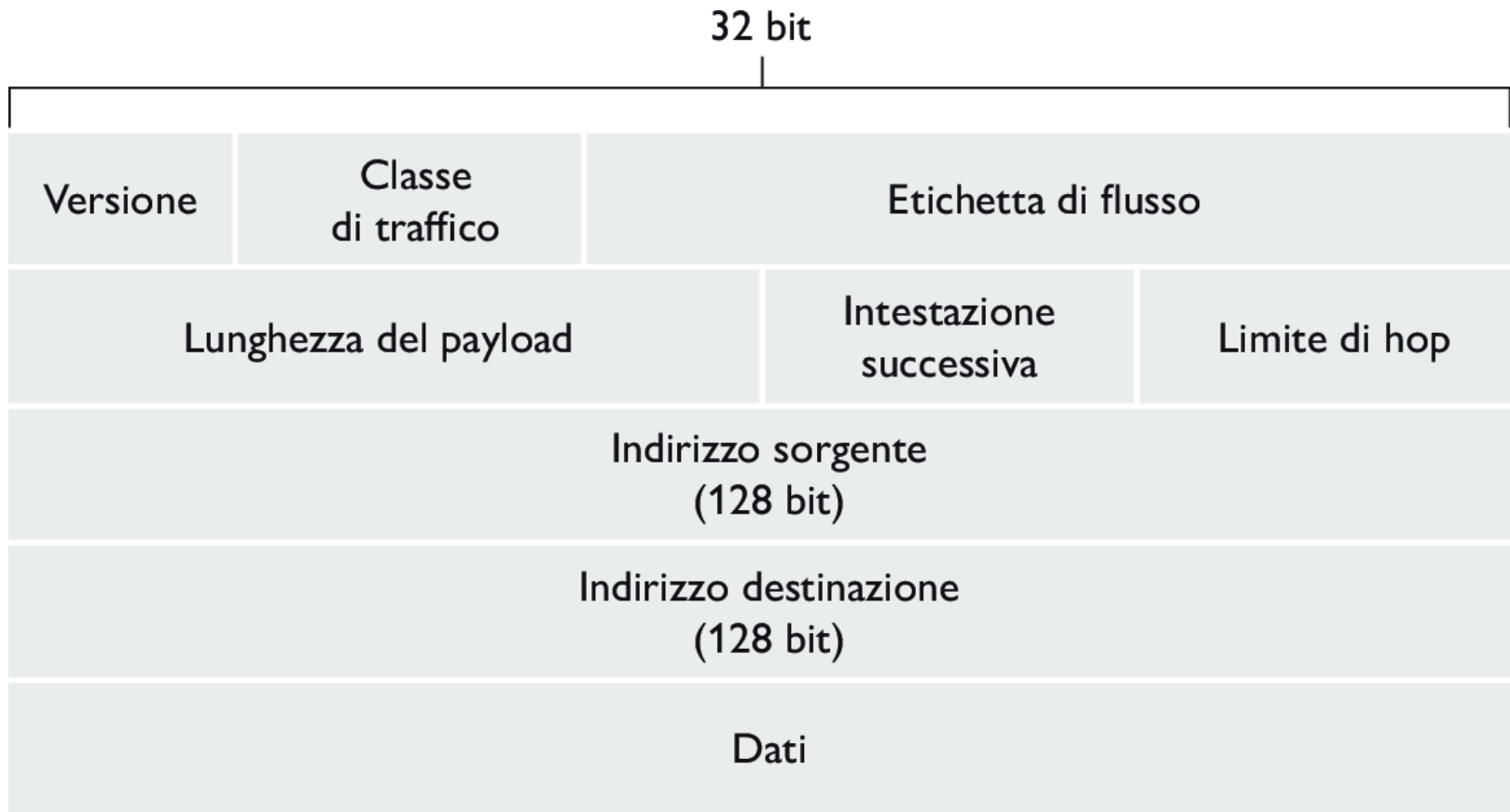
Indirizzi IPv6

- 128 bit (16 byte)
- Notazione esadecimale
 - 8 blocchi da 16 bit ognuno
 - 2001:0000:1234:0000:0000:00d0:abcd:0532
- Scrittura semplificata
 - zero opzionali a sinistra di ogni campo
 - 2001:0:1234:0:0:d0:abcd:532
 - Campi successivi di zero rappresentati da :: (solo una volta)
 - 2001:0:1234::d0:abcd:532
 - 0:0:0:0:0:0:0:1 => ::1 (indirizzo di loopback)

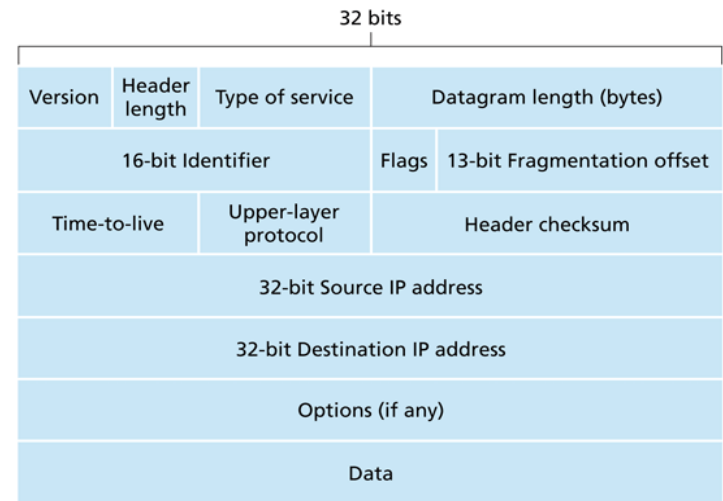
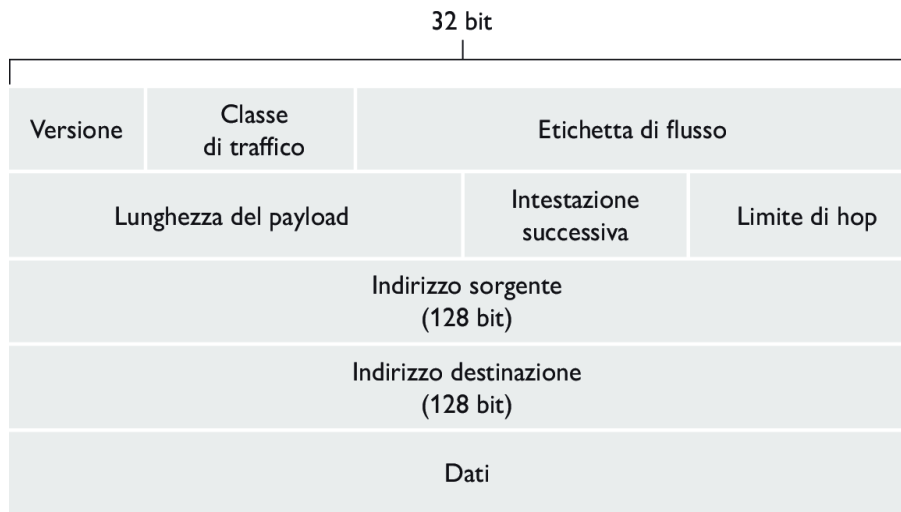
Indirizzi IPv6

- Nessuna maschera di sottorete
- Subnet Prefix (64 bit) + Host Identifier (64 bit)
 - I primi 64 bit indicano il netid
 - I primi 48 bit usati per le tabelle di routing
 - I successivi 16 bit per definire sottoreti
 - Gli ultimi 64 bit indicano l'hostid

Formato datagramma IPv6



IPv6 vs. IPv4



- Indirizzamento esteso
- Intestazione ottimizzata a 40 byte e a lunghezza fissa
 - Rimossi alcuni campi (Options)
- Frammentazione e riassetblaggio non consentiti nei router intermedi
 - Solo in quelli sorgente e destinazione
- Checksum affidato al livello di trasporto e di collegamento
 - Rimosso perchè ridondante a livello di rete

Passaggio da IPv4 a IPv6: doppia pila (dual stack)

- Nodi capaci di inviare e ricevere sia datagrammi IPv4 che IPv6
- Doppio indirizzamento gestito dal DNS
 - type AAAA

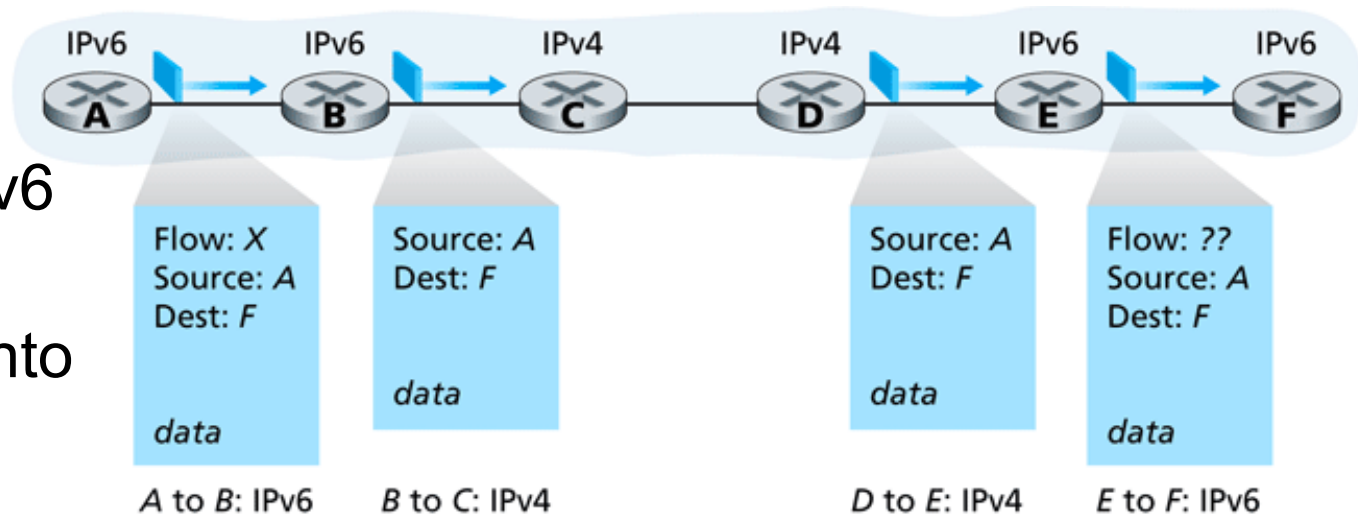
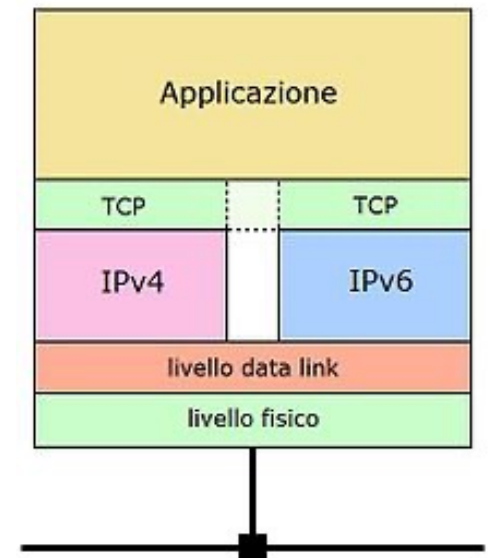


Figure 4.25 ♦ A dual-stack approach



Passaggio da IPv4 a IPv6: tunneling

- Datagram IPv6 incapsulati in datagram IPv4
- Protocol numbers
 - 1 ICMP
 - 6 TCP
 - 17 UDP
 - 41 IPv6

