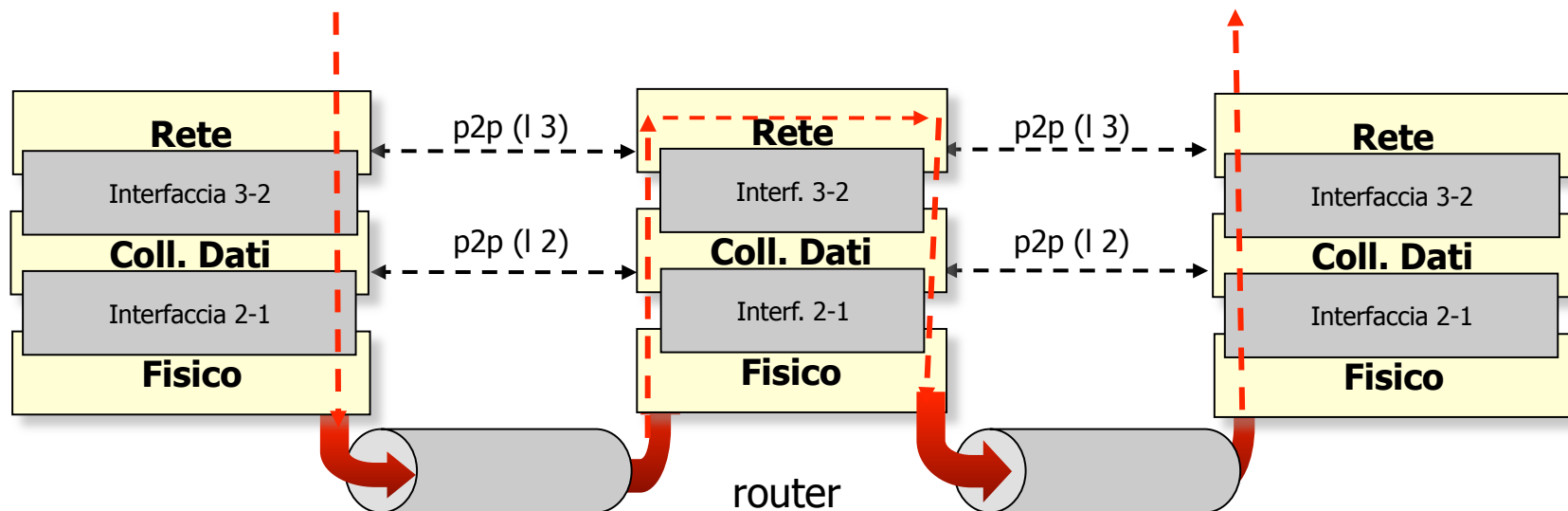


Network Layer

- Concatenates data link networks
- Packet fragmentation
- Network-wide routing
- Network-wide addressing



Network Layer

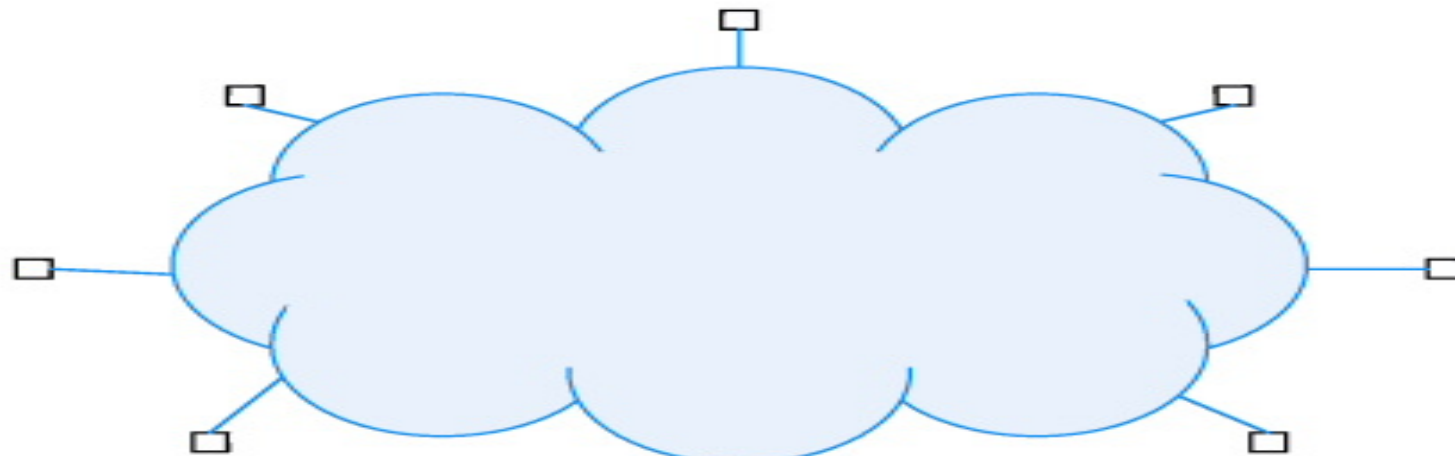
Le funzioni principali del layer “Network” sono:

- * *Indirizzamento globale*
- * *Instradamento*

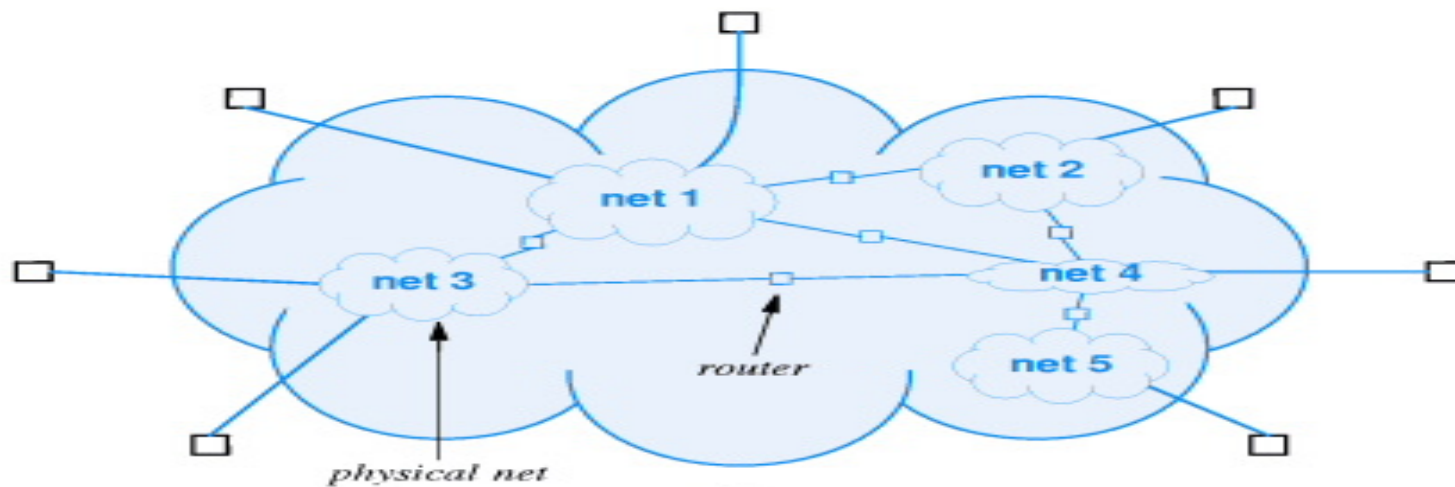
Principali Obiettivi:

- Connettere diverse tecnologie per formare **internetworking**
- Fornire uno schema di **indirizzamento universale**
- Nascondere le tecnologie di rete all’end-user
- Muovere **pacchetti** tra ogni host sulla rete

Internet: il concetto base

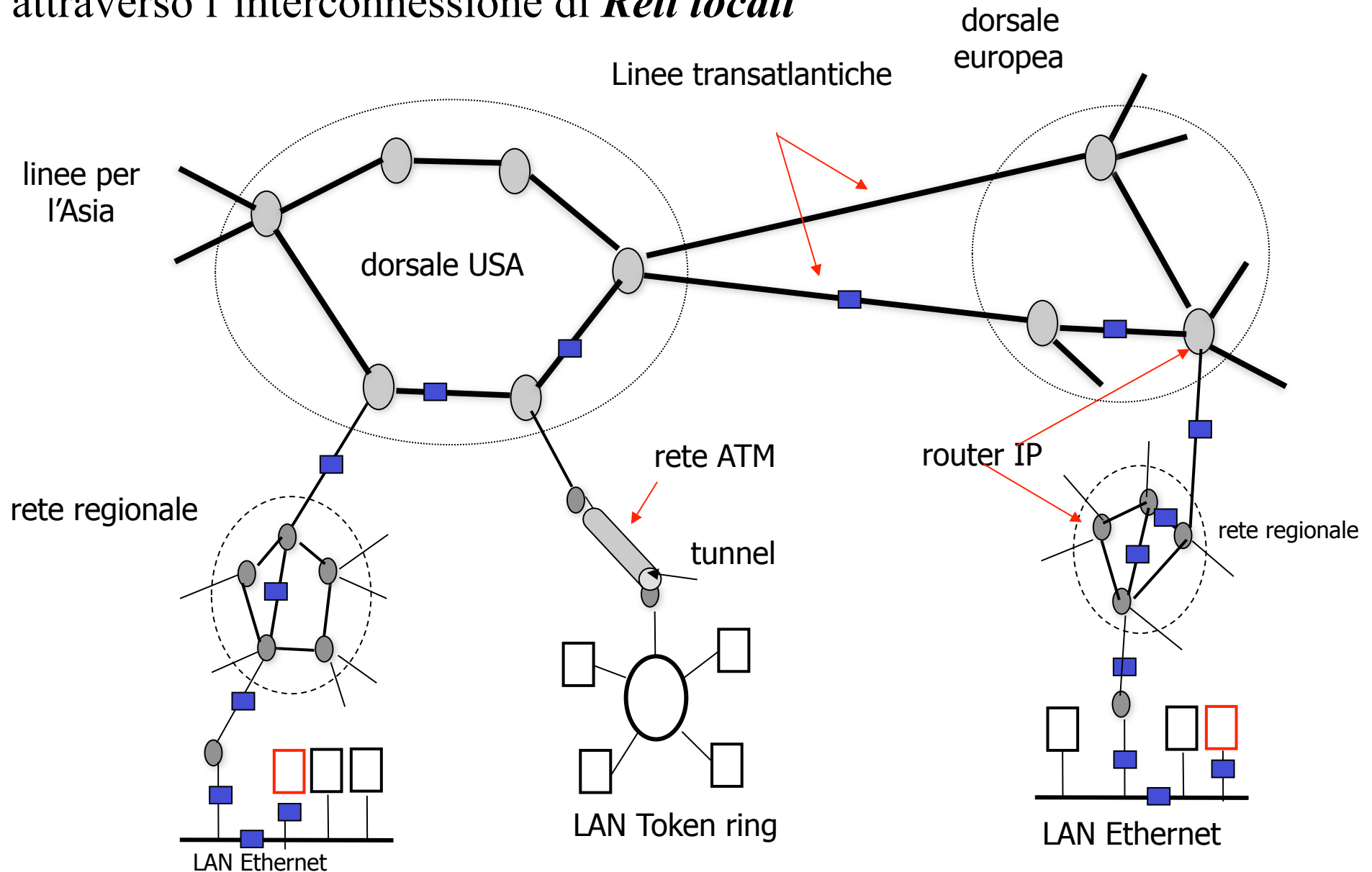


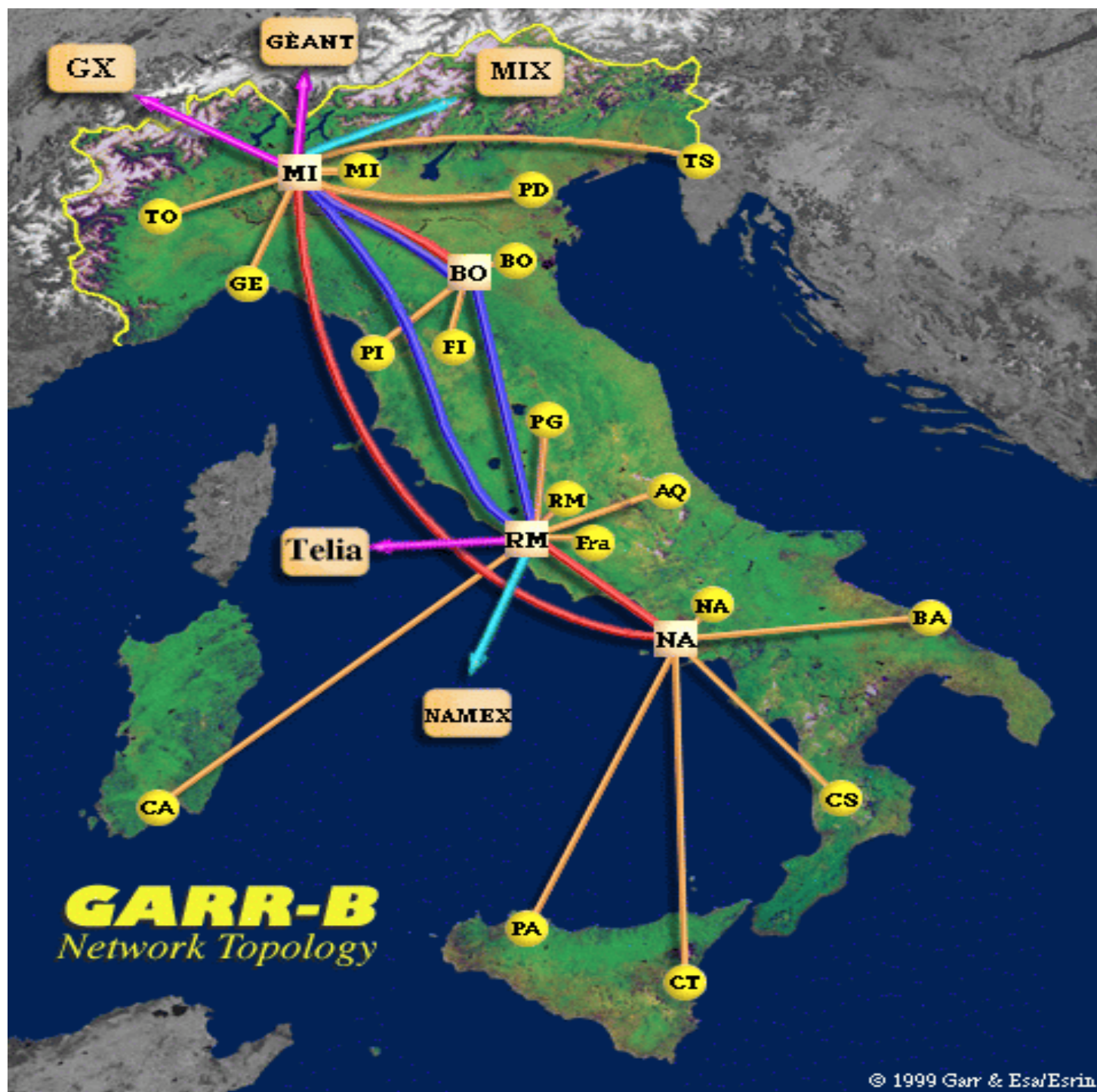
(a)



(b)

Internet (*una rete di reti*): **Backbone**, ad alta capacità, collegano router veloci appartenenti a **Reti regionali**, a loro volta costituite attraverso l'interconnessione di **Reti locali**





Ciò che tiene tutto insieme è l'**IP** (**Internet Protocol**), il protocollo di livello network dell'architettura TCP/IP.

IP: protocollo datagram (non connesso e non affidabile) che opera come segue:

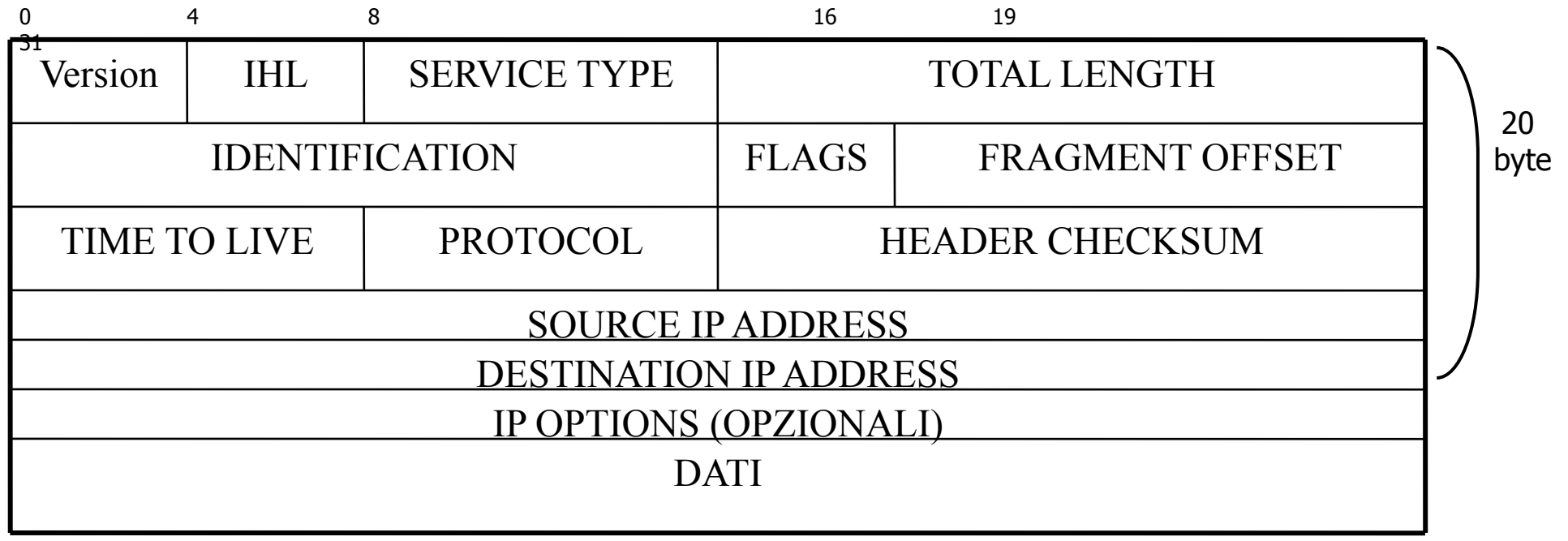
- * riceve i dati dal livello transport e li incapsula in pacchetti (massimo 64 Kbyte, normalmente circa 1.500 byte);
- * instrada i pacchetti sulla subnet, eventualmente frammentandoli lungo il viaggio;

a destinazione:

- * riassembla (se necessario) i frammenti in pacchetti;
- * estrae da questi i dati del livello transport;
- * consegna al livello transport i dati arrivati.

Internet Protocol (IP): protocollo senza connessione (best-effort) standardizzato in RFC 791, è la base di Internet:

- * I **datagrammi** contengono l'indirizzo del destinatario
- * Ogni datagramma viene spedito/gestito indipendentemente
- * La consegna non è garantita



Datagrammi IP

- * Seguono una politica di **best-effort**
- * Possono essere ritardati
- * Duplicati
- * Distribuiti fuori ordine
- * Persi

Pacchetti diversi dello stesso messaggio possono seguire percorsi diversi

Header IP

0	4	8	16	19	31
Version	IHL	SERVICE TYPE	TOTAL LENGTH		
IDENTIFICATION			FLAGS	FRAGMENT OFFSET	
TIME TO LIVE		PROTOCOL	HEADER CHECKSUM		

Version: specifica la versioni

IHL: specifica la lunghezza del preambolo in parole da 32 bit

SERVICE TYPE: (3+3+2) 3 bit per la precedenza, 3 bit per il tipo di servizio in termini di affidabilità, velocità e ritardo (può influenzare le scelte di routing), 2 inutilizzati

TOTAL LENGTH: specifica la lunghezza dell'intero datagram (header + dati al massimo $2^{16}-1=65535$ byte)

Header IP

0 31	4	8	16	19
Version	IHL	SERVICE TYPE	TOTAL LENGTH	
IDENTIFICATION			FLAGS	FRAGMENT OFFSET
TIME TO LIVE		PROTOCOL	HEADER CHECKSUM	

Frammentazione: necessaria se il datagram eccede la massima lunghezza del frame datalink utilizzato (ad es. 1500 byte per Ethernet)

IDENTIFICATION: identifica il datagram a cui appartiene il frammento

FRAGMENT OFFSET: (13 bit) identifica il numero d'ordine del frammento (massimo 8192 frammenti)

FLAGS: (1+1+1) bit, il primo è inutilizzato, il secondo (*Don't Fragment*) ordina ai router di non frammentare, il terzo indica se ci sono (*More Fragments*) o meno altri frammenti

Header IP

0	4	8	16	19	31
Version	IHL	SERVICE TYPE	TOTAL LENGTH		
IDENTIFICATION			FLAGS	FRAGMENT OFFSET	
TIME TO LIVE		PROTOCOL	HEADER CHECKSUM		

TIME TO LIVE: contatore (0-255) utilizzato per limitare la vita dei pacchetti viene decrementato ad ogni salto (hop). Quando si azzerà il pacchetto è scartato e si invia una notifica al mittente

PROTOCOL: individua il protocollo di trasporto (es. TCP o UDP)

HEADER CHECKSUM: calcolato solo sull'header, deve essere ricalcolato ad ogni hop, visto che cambia il campo **TTL**

Header IP Options

0	4	8	16	19	31
Version	IHL	SERVICE TYPE	TOTAL LENGTH		
IDENTIFICATION			FLAGS	FRAGMENT OFFSET	
TIME TO LIVE		PROTOCOL	HEADER CHECKSUM		
SOURCE IP ADDRESS					
DESTINATION IP ADDRESS					
IP OPTIONS (OPZIONALI)					

Il primo byte identifica l'opzione, il secondo ne indica la lunghezza e i restanti specificano i dati relativi all'opzione attivata.

Attualmente vi sono solo cinque opzioni: *Security*, *Strict source routing*, *Louse source routing*, *Record route*, *Timestamp*.

Header IP Options

Security: *Specifica il livello di segretezza*

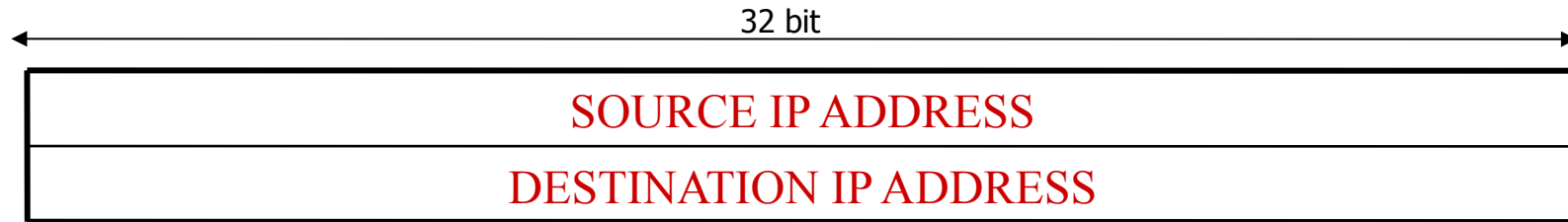
Strict source routing: *specifica il percorso dalla sorgente alla destinazione con una sequenza di indirizzi IP*

Louse source routing: *fornisce una lista di router che non devono essere saltati*

Record route: *forza i router ad aggiungere i loro IP nel campo opzione del pacchetto (max 9 router)*

Timestamp: *ogni router sul percorso oltre all'IP aggiunge anche un timestamp a 32 bit*

Indirizzamento IP



L'indirizzo **IP** a 32 bit identifica **univocamente** un dispositivo (host o router) sulla rete

Gli indirizzi IP **pubblici** sulla rete Internet sono assegnati dal **NIC** (Network Information Center)

Due dispositivi connessi alla rete Internet non possono avere lo stesso IP

Un indirizzo IP identifica una connessione alle rete

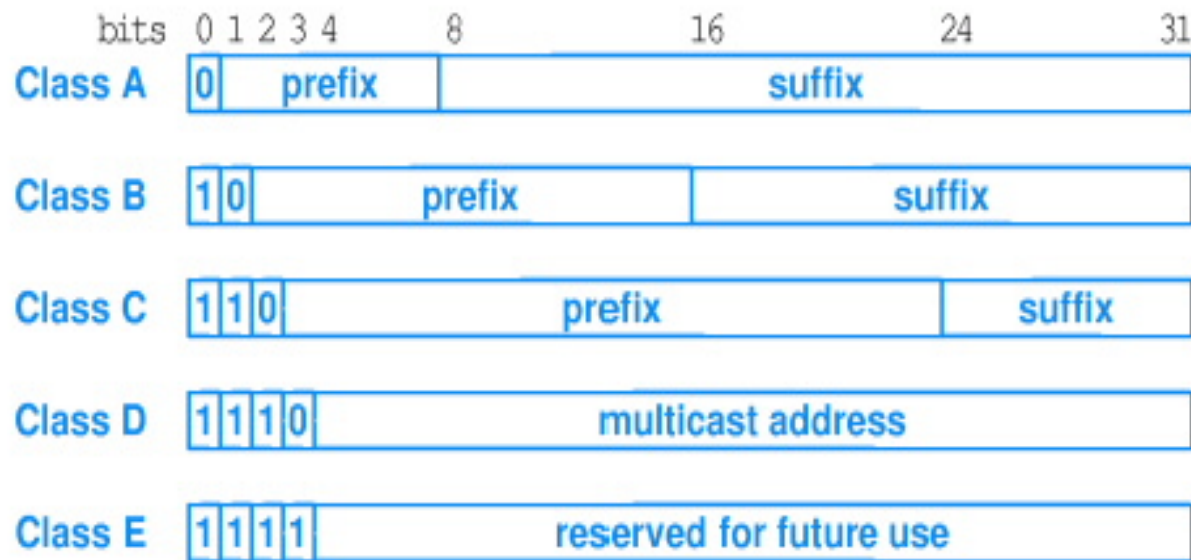
** Un host che ha più interfacce di rete ha più indirizzi IP*

Indirizzi IP: 4 byte, rappresentati con 4 numeri decimali (0-255) separati da punti (.)

Indirizzo IP binario (32 bit)				Indirizzo numerico
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

- L'indirizzo è composto da:
 - Indirizzo di rete (*net_id*)
 - Indirizzo dell'host nella rete (*host_id*)

Classi di indirizzi IP



1.0.0.0
 127.255.255.255
 128.0.0.0
 191.255.255.255
 192.0.0.0
 223.255.255.255
 224.0.0.0
 239.255.255.255
 240.0.0.0
 247.255.255.255

<i>classe</i>	<i>bit nel prefisso</i>	<i>numero massimo di reti</i>	<i>bit nel suffisso</i>	<i>numero massimo di host per rete</i>
<i>A</i>	7	128	24	16777216
<i>B</i>	14	16384	16	65536
<i>C</i>	21	2097152	8	256

Indirizzi speciali

prefisso	suffisso	indirizzo	scopo
tutti 0	tutti 0	<i>questo computer</i>	utilizzato in fase di boot
rete	tutti 0	<i>la rete</i>	identificare una rete
rete	tutti 1	<i>broadcast diretto</i>	bcast su una rete remota
tutti 1	tutti 1	<i>broadcast locale</i>	broadcast sulla propria rete
127	qualsiasi	<i>loopback</i>	Test

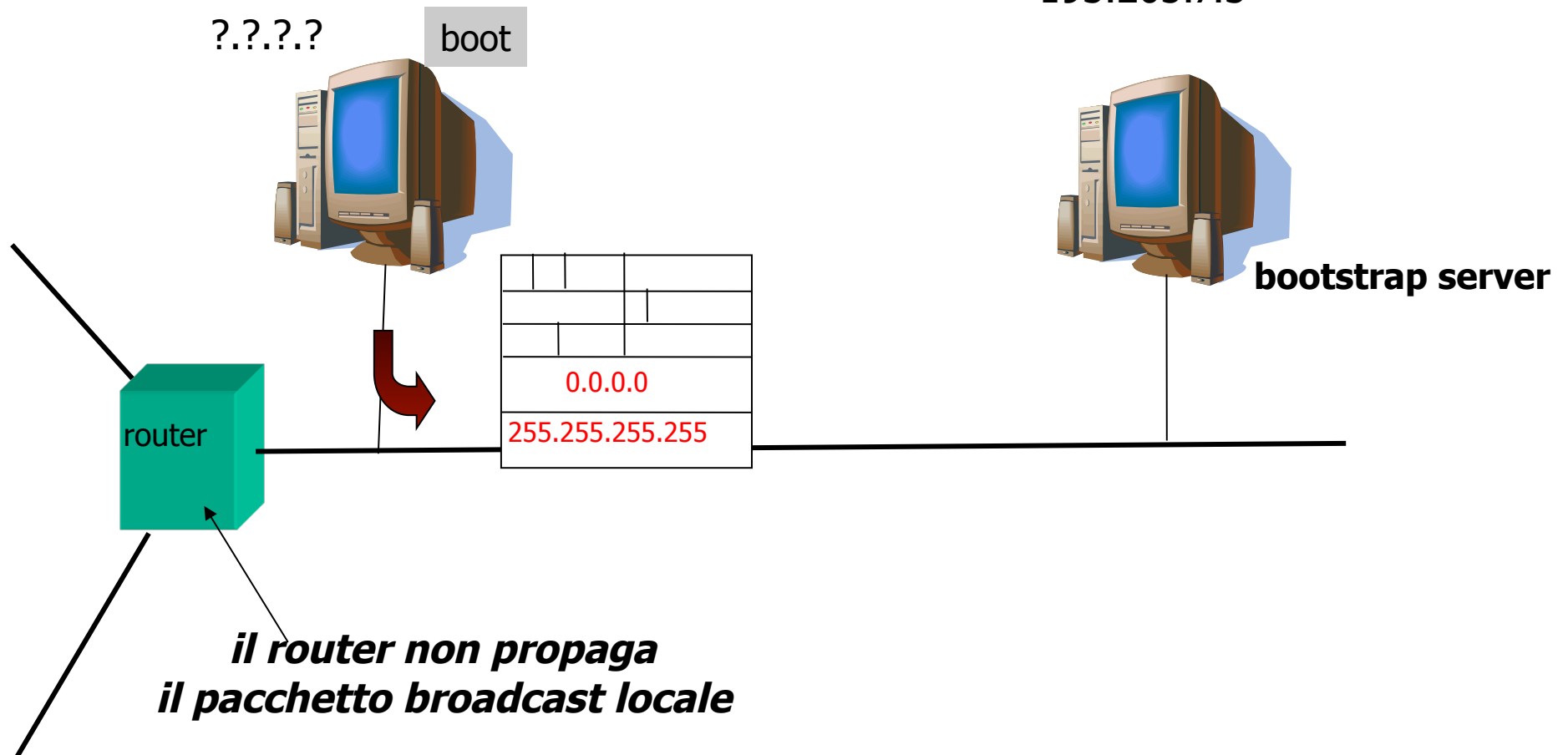
Broadcast limitato: i pacchetti sono confinati alla sottorete, il router non li propaga all'esterno

Indirizzi di loopback: il pacchetto viene smistato localmente e solo a livello rete (non viene passato al livello fisico)

Indirizzi speciali:

0.0.0.0 indica questo host su questa rete

255.255.255.255 broadcast sulla rete corrente (locale)



Indirizzi privati: indirizzi riservati per reti (private)
non connesse ad Internet

classe

rete

numero reti

A

10.0.0

1

B

Da ***172.16*** a ***172.31***

16

C

Da ***192.168.0*** a ***192.168.255***

256

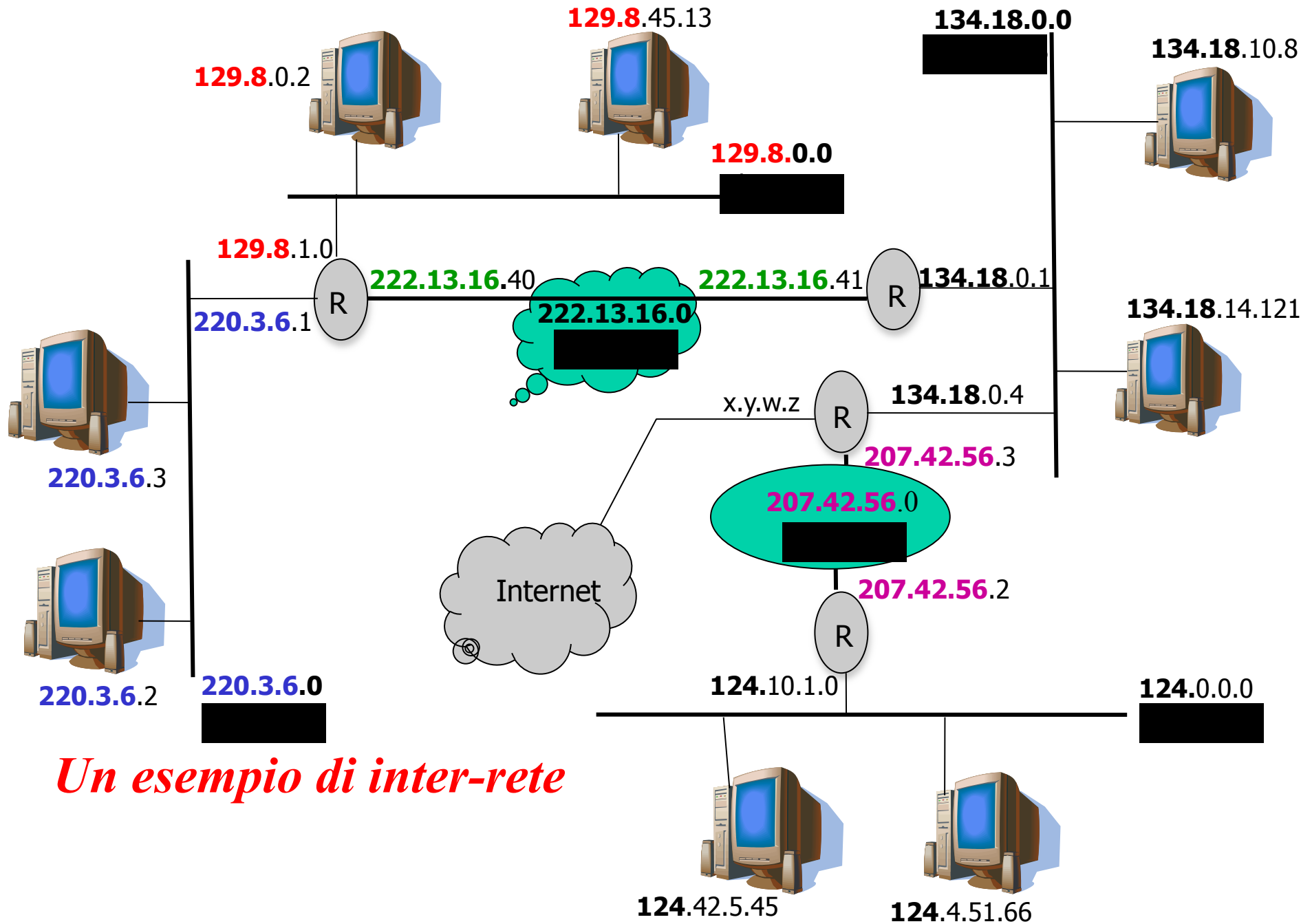
La suddivisione in Classi, e la crescita esplosiva di Internet, ha causato un rapido consumo degli indirizzi IPv4.

IPv6: indirizzi non più a 32 bit, ma a **128 bit**
(3.40×10^{38} IP address).

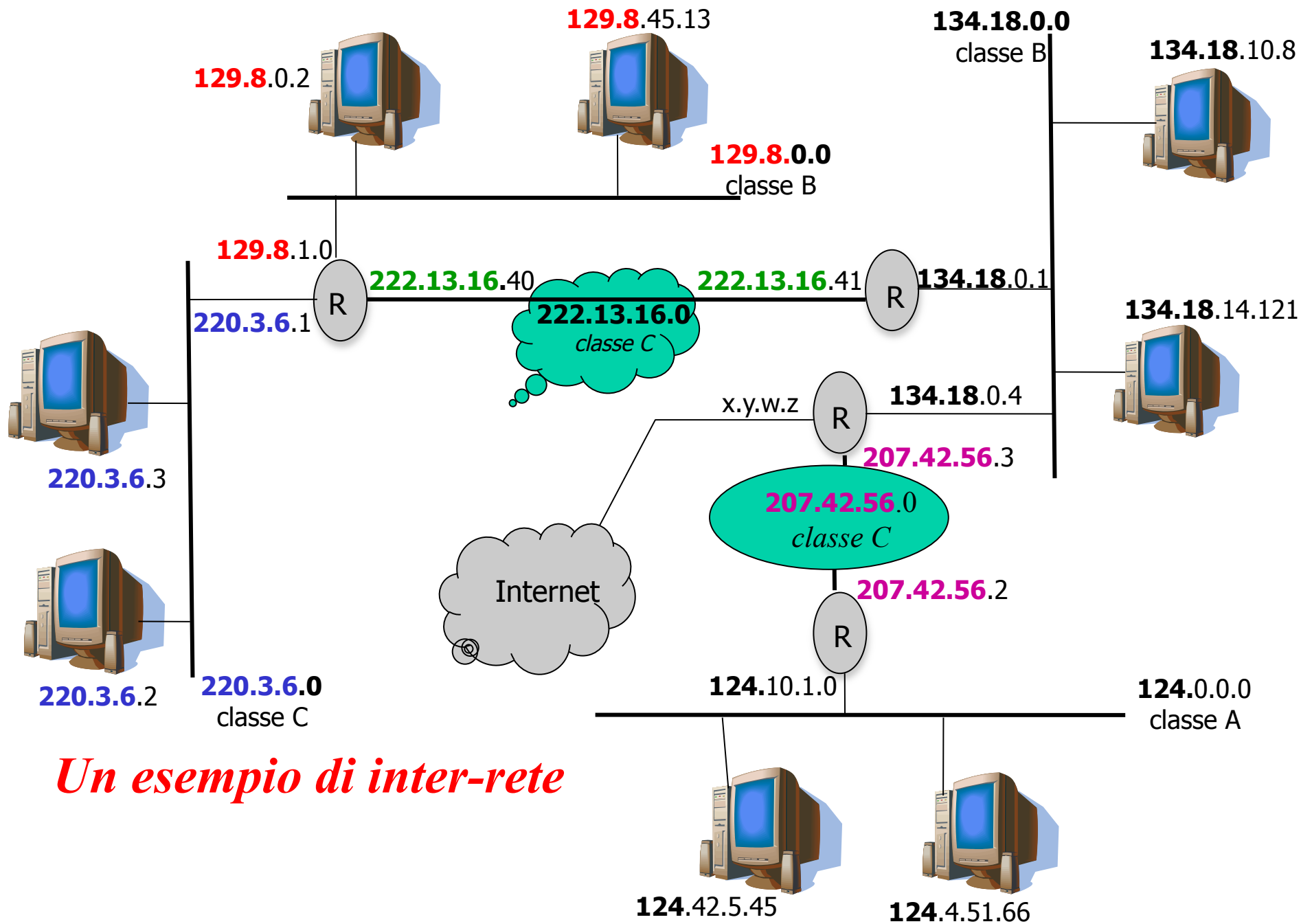
Con IPv6 sulla terra (oceani e mari compresi) esisterebbero 7×10^{23} *IP address per m²*.

145.123.136.201.255.255.255.0.0.0.21.236.18.150.255.10

917B:88C9:FFFF:FF00:0000:15EC:1296:FF0A

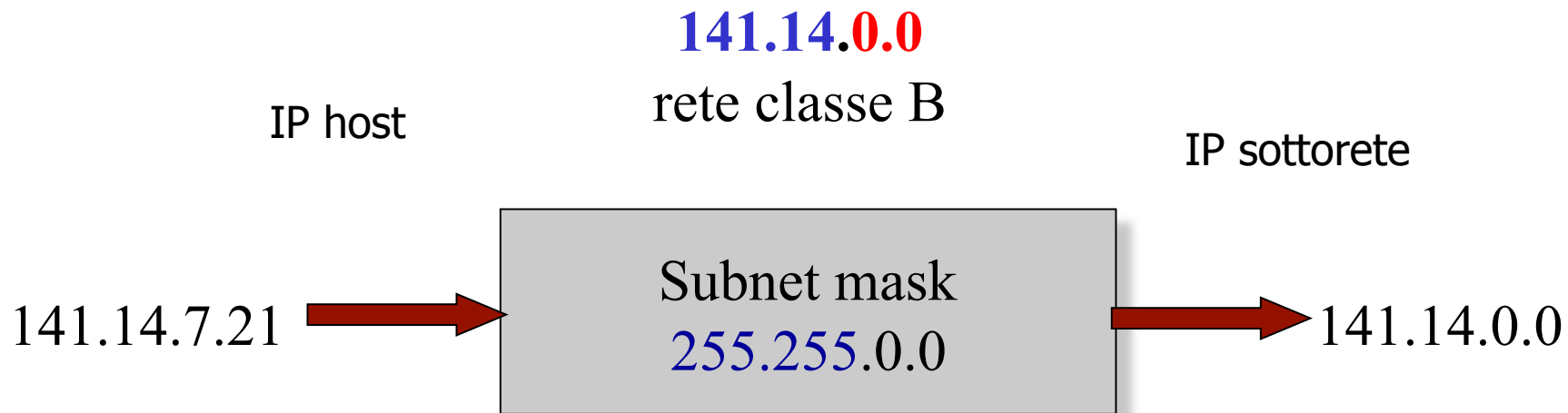


Un esempio di inter-rete



Un esempio di inter-rete

Subnet Mask: per individuare l'indirizzo di rete si utilizza una maschera con bit 1 in corrispondenza del *net_id* e 0 dell'*host_id*



11111111.11111111.00000000.00000000 mask

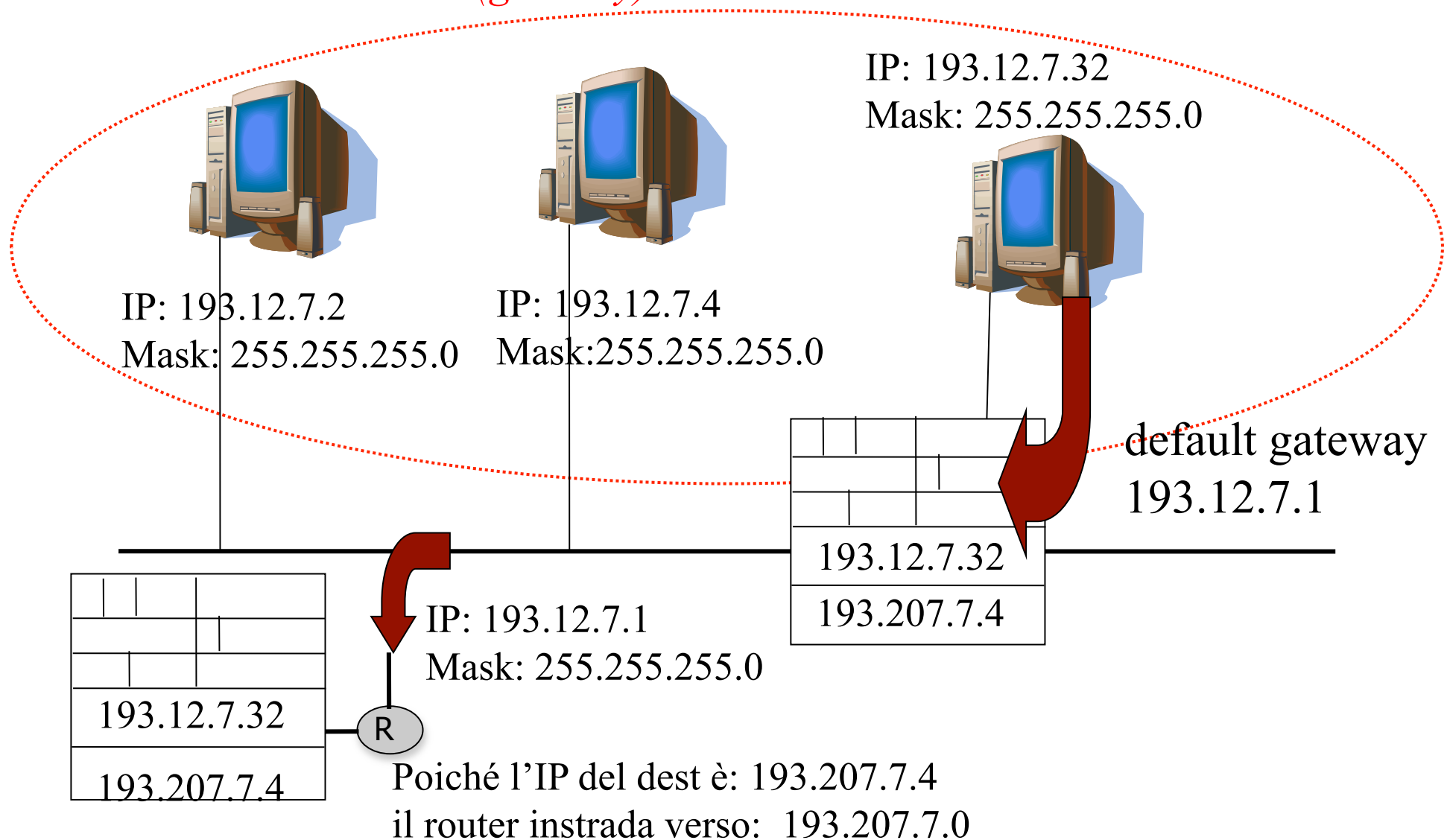
10001101.00001110.00000111.00010101 IP host

10001101.00001110.00000000.00000000 IP sottorete

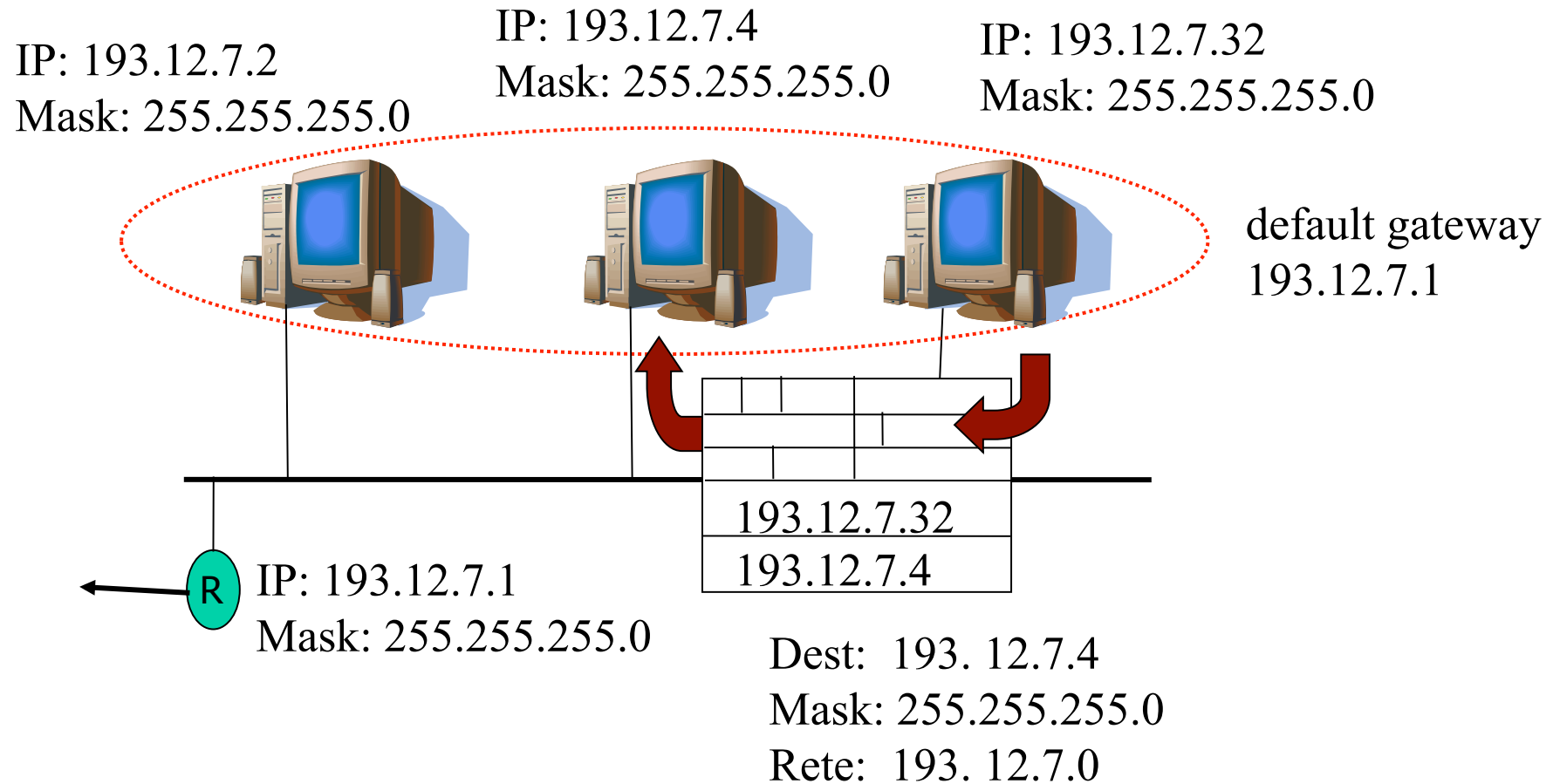
Indirizzo di rete: *AND bit a bit fra l'IP e la maschera*

Subnet Mask e instradamento: la subnet mask permette di verificare se l'IP del destinatario appartiene o meno alla rete del mittente

Se l'host destinatario non è nella stessa sottorete il pacchetto deve essere instradato verso il router (gateway)



Consegna diretta: destinatario e mittente sono sulla stessa sottorete



ARP (Address Resolution Protocol): associa all'indirizzo IP del destinatario il suo indirizzo fisico

Protocolli *ARP* e *RARP*

L'associazione *IP* \rightarrow *Indirizzo fisico* può essere:

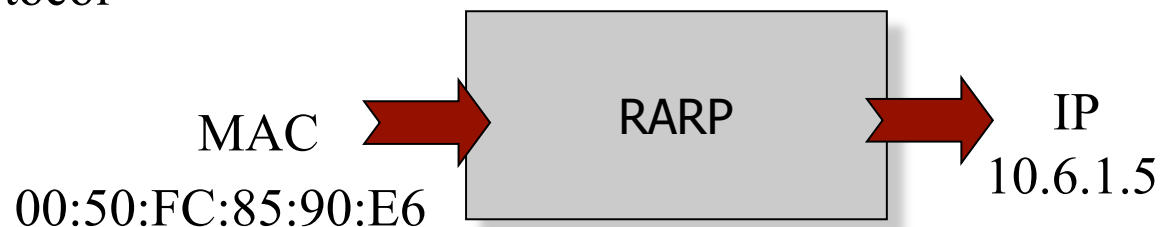
Statica: le associazioni vengono registrate in una tabella che, periodicamente aggiornata, viene memorizzata su tutti i dispositivi della sottorete (poco flessibile)

Dinamica: le associazioni si ottengono all'occorrenza attraverso esplicita richiesta inoltrate sulla rete



Address Resolution Protocol

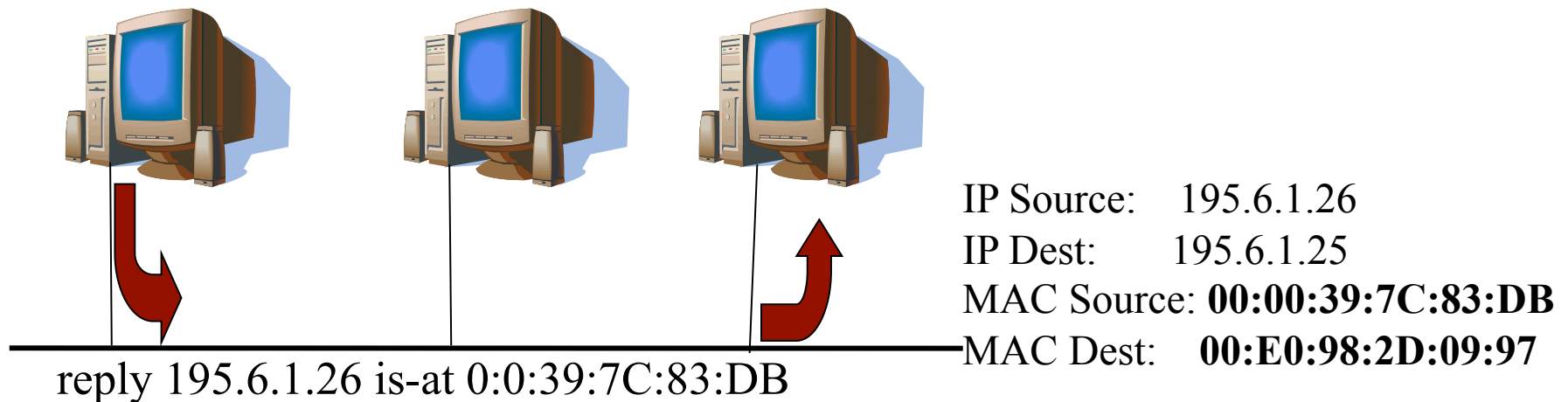
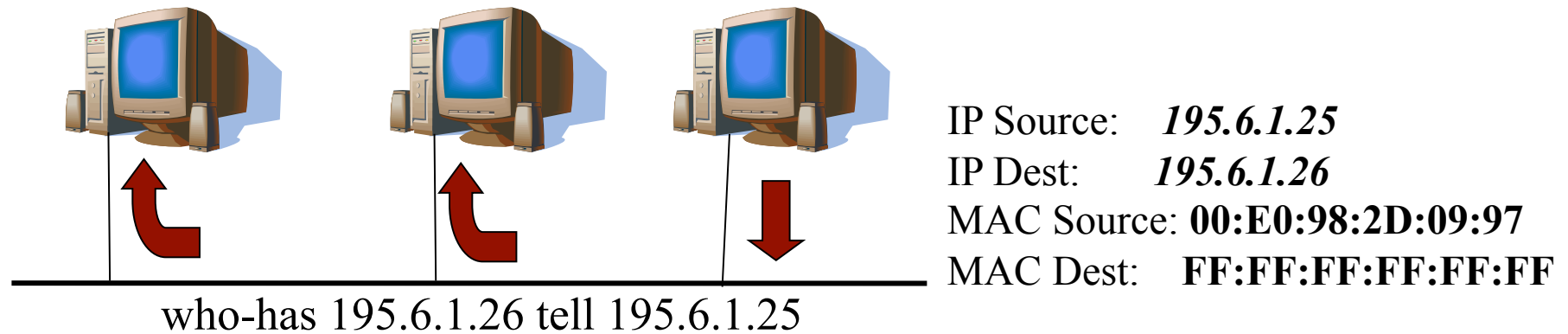
Reverse Address Resolution Protocol



Protocollo ARP

Il mittente inoltra in broadcast una richiesta ARP per l'IP di interesse.
La macchina con quell'IP risponde col suo indirizzo MAC

IP: 195.6.1.26	IP: 195.6.1.5	IP: 195.6.1.25
Mask: 255.255.255.0	Mask: 255.255.255.0	Mask: 255.255.255.0



Cache ARP: per evitare la risoluzione $IP \Rightarrow MAC$ per ogni pacchetto le coppie IP:MAC, via via ottenute, vengono registrate in una cache locale

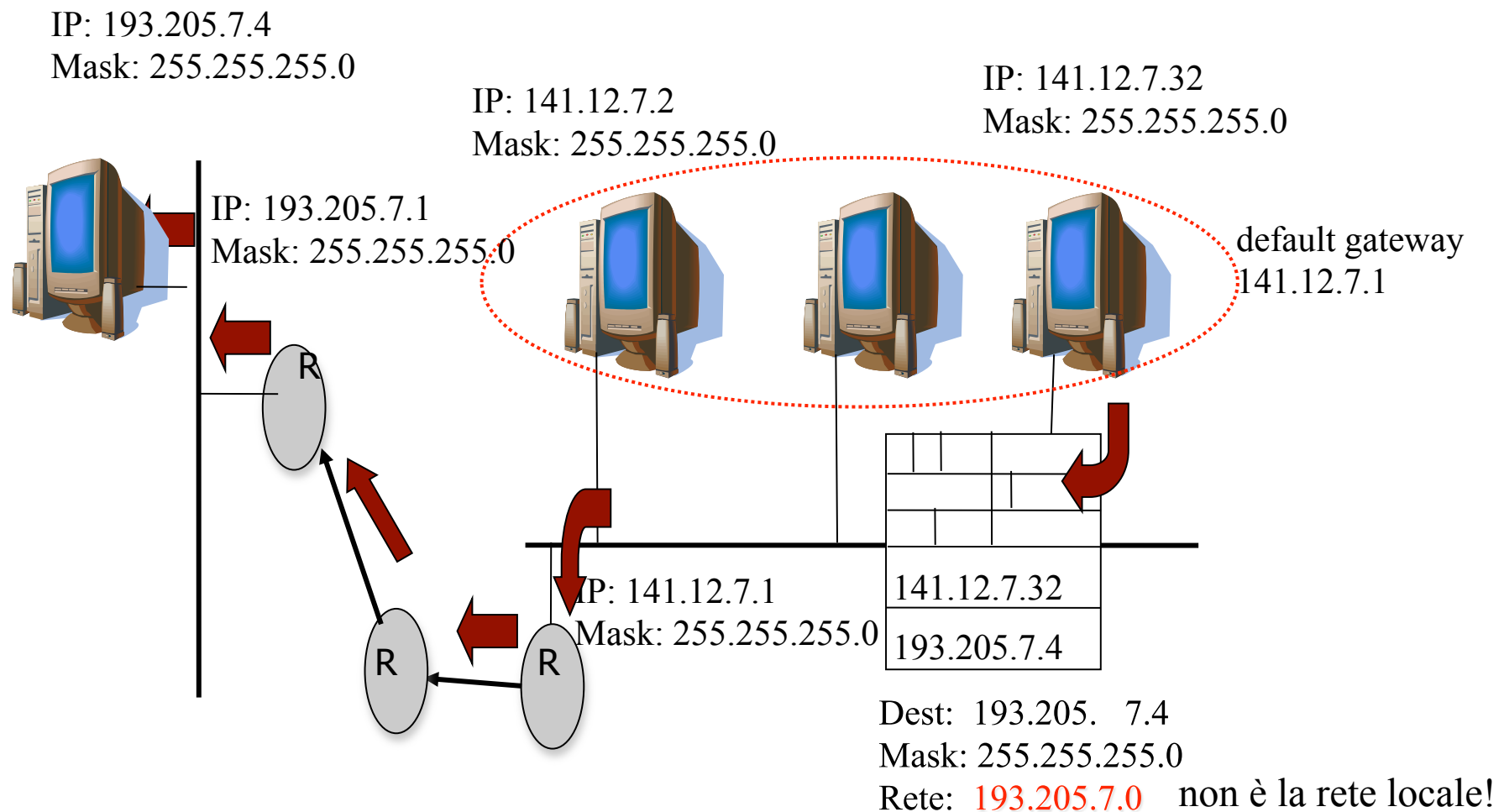
```
[host]# arp -n
```

Address	HWtype	HWaddress	Flags	Mask	Iface
193.205.7.1	ether	00:00:0C:3A:99:C6	C		eth0
193.205.7.2	ether	00:10:5A:2E:5A:CE	C		eth0

Con il comando **arp** di Unix è possibile consultare la cache

Consegna indiretta: la destinazione è un host connesso ad una rete fisica diversa da quella in cui si trova il mittente.

Il mittente, stabilito che la destinazione non è sulla sua sottorete, invia il pacchetto al gateway (router) predefinito



Scelta del gateway

Sottorete con più router: sull'host può essere definita una tabella di routing che specifica il gateway a cui inviare i pacchetti per ogni specifica destinazione.

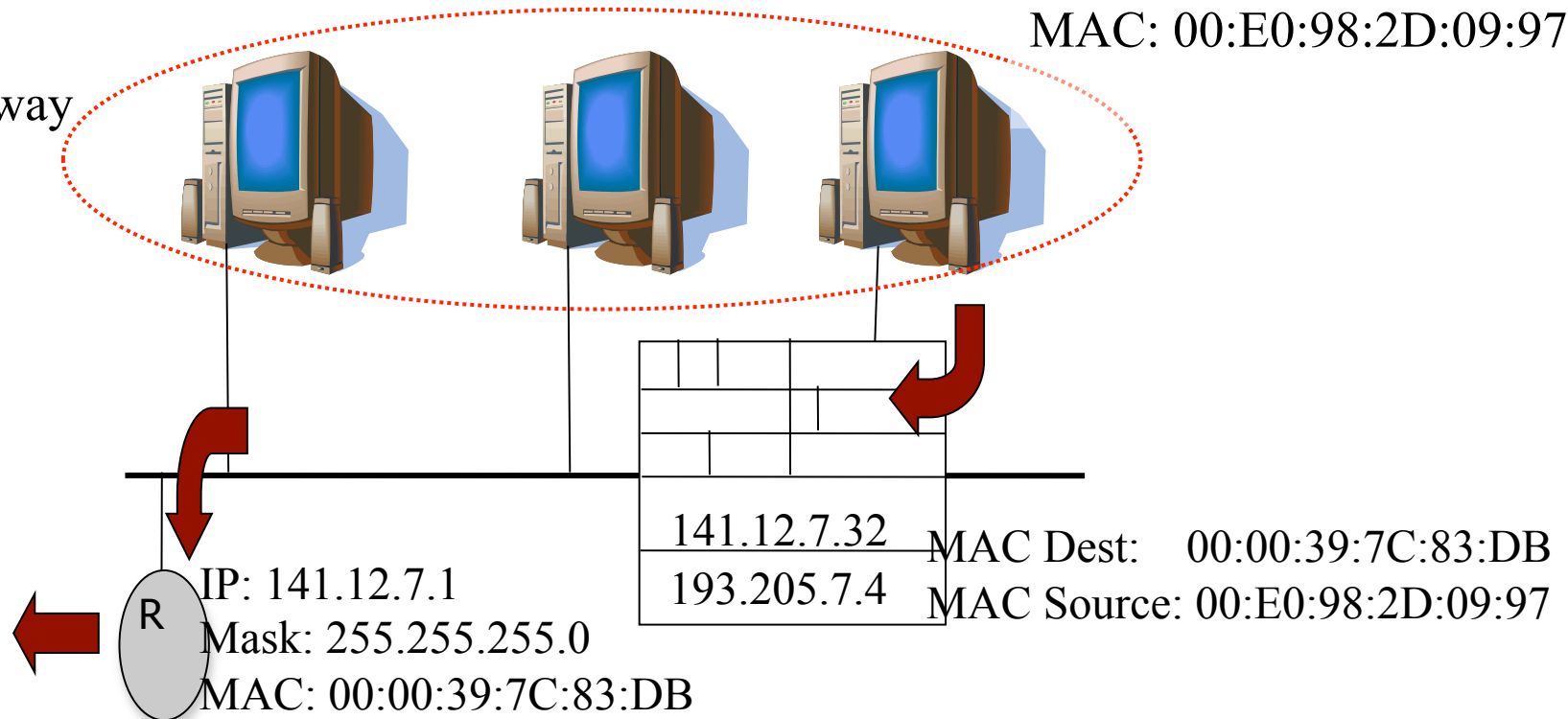
Sottorete con un solo router (gateway di default): l'host invia il pacchetto direttamente gateway.

IP: 141.12.7.32

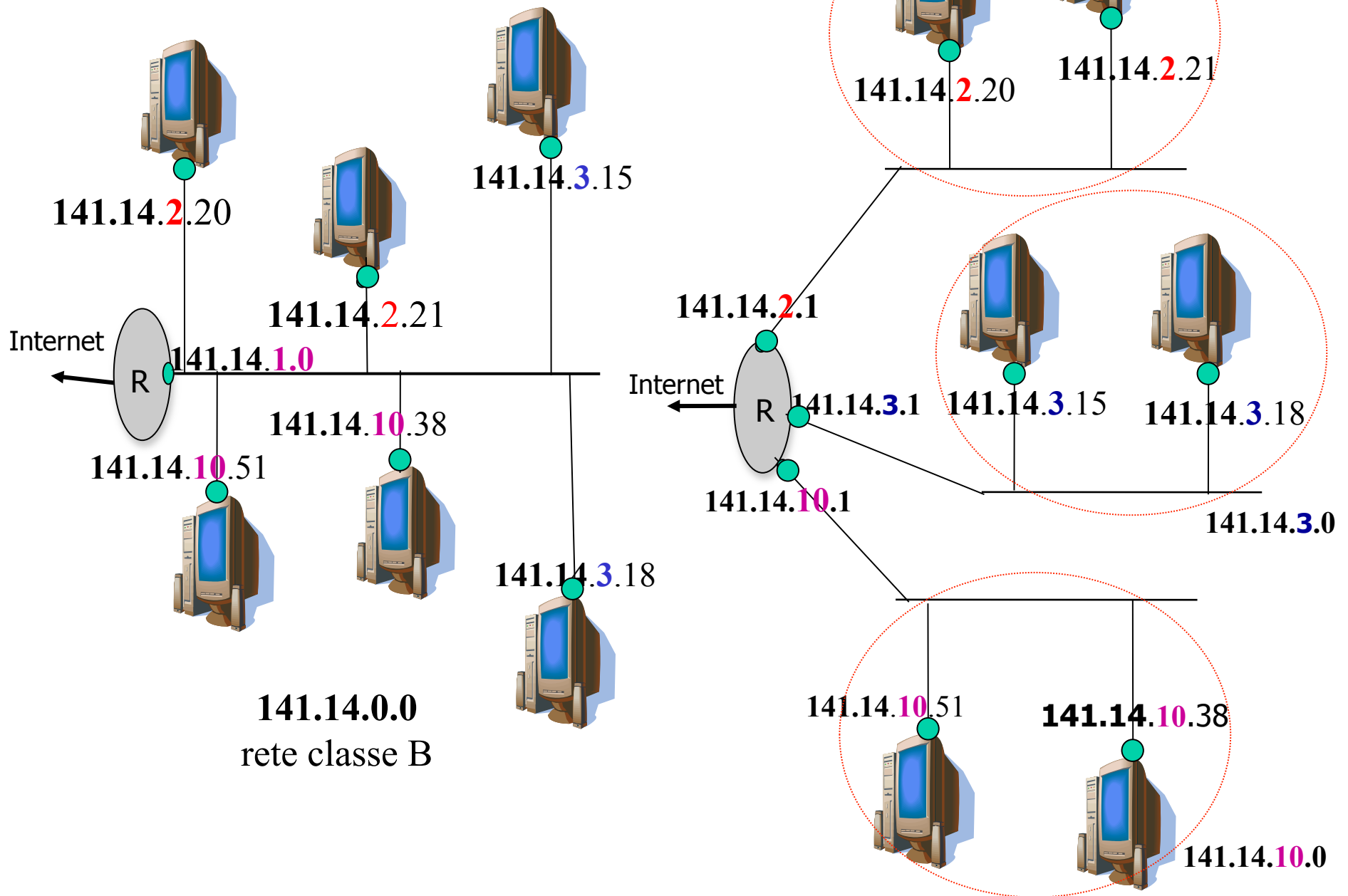
Mask: 255.255.255.0

MAC: 00:E0:98:2D:09:97

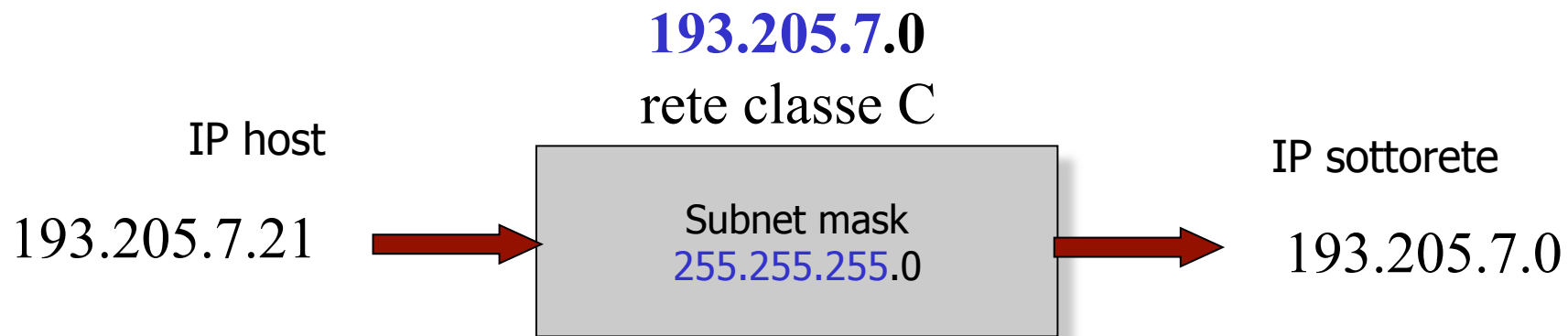
default gateway
141.12.7.1



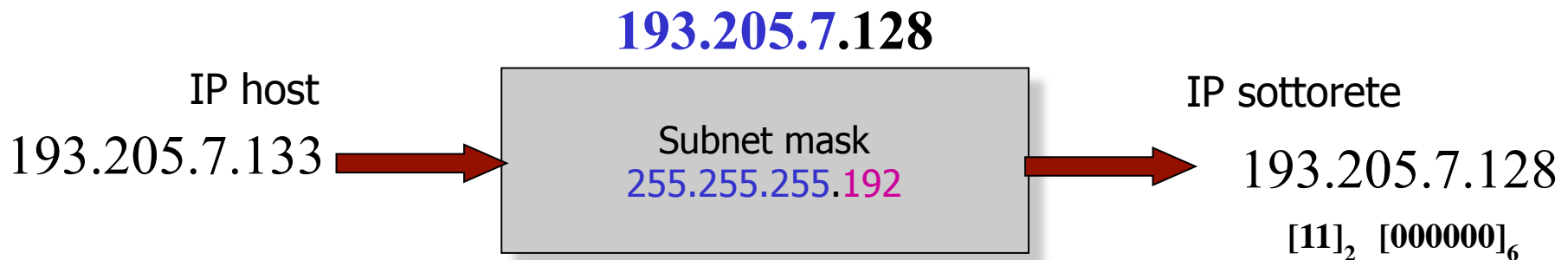
Subnetting



Subnet Mask per classe C



Una rete di classe C: **quattro (2 bit) sottoreti di 62 host (6 bit)**



11111111.11111111.11111111.11 000000 mask

11000001.11001101.00000111.10 000101 IP host

11000001.00001110.00000111.10000000 IP sottorete

$2^2 \text{ bit} = 4 \text{ reti}$

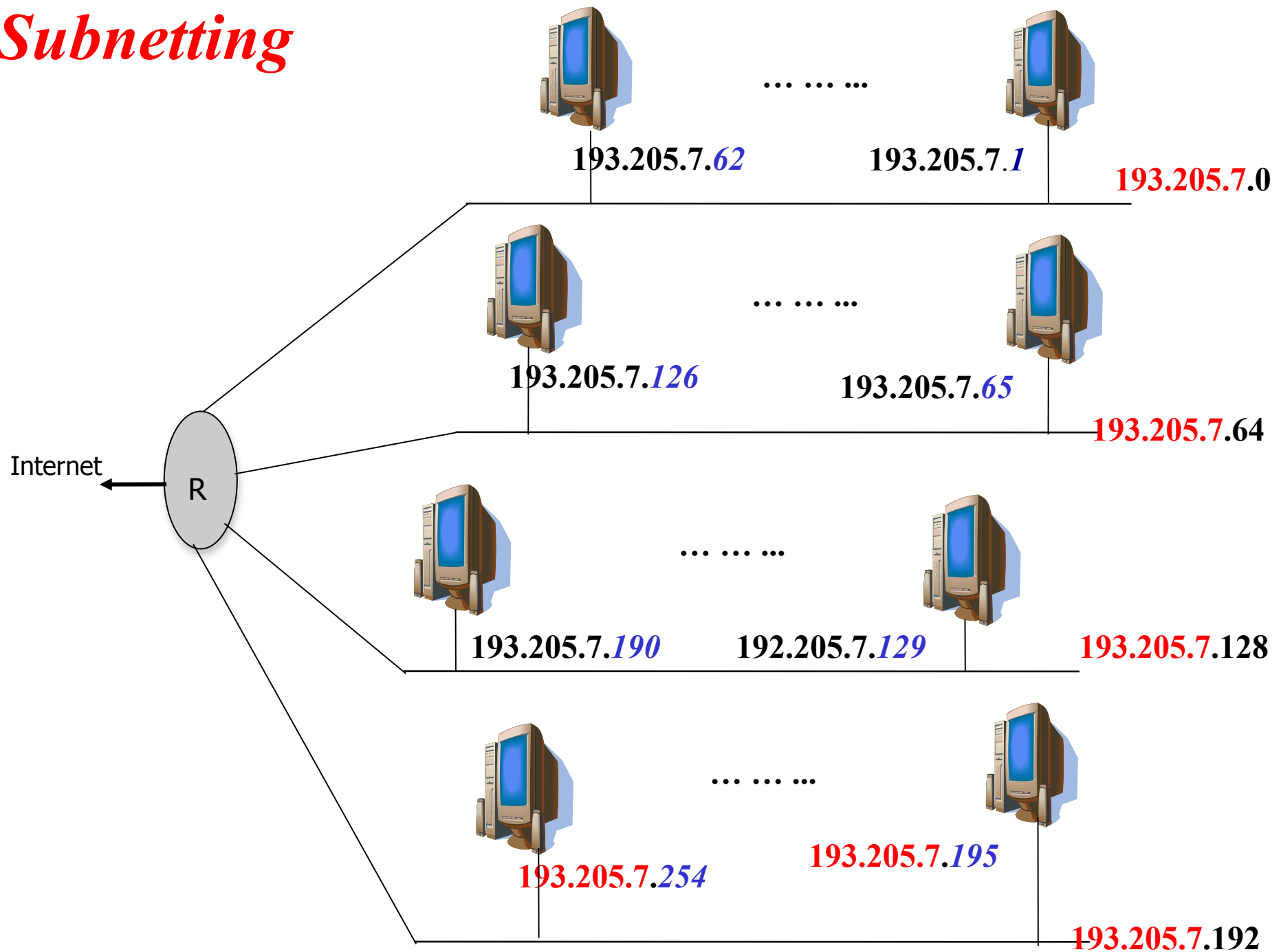
$2^6 \text{ bit} = 64 \text{ hosts}$

per ogni sottorete abbiamo:

1 sotto_rete + 1 broadcast

$64 - 2 \text{ (default use)} = 62 \text{ hosts}$

Subnetting



Per descrivere facilmente una subnet è possibile agire sulla scrittura dell'indirizzo ip

Usiamo il numero di bit della maschera di rete per determinare la subnet:

193.205.7.0 ha una **4 sottoreti** con netmask

11111111 11111111 11111111 **11000000**

8	8	8	2	26
---	---	---	---	----

Possiamo riscriverla come:

193.205.7.0/26

193.205.7.64/26

193.205.7.127/26

192.205.7.192/26

Variable Length Subnet Mask (VLSM)

E' possibile dividere una subnet in più subnet
Creiamo diverse maschere di rete per ogni subnet

Es.

192.205.7.0/26 - 64 indirizzi (62 hosts + 2 rete/bc)

Dividiamo la sottorete in 2 ulteriori sottoreti, necessitiamo di **1 bit** ulteriore – rimangono **5 bit per gli hosts** (32 indirizzi)

11111111 11111111 11111111 11100000

8	8	8	3	27
---	---	---	---	----

S1: 192.205.7.0/27 -> 0-31 [0: rete, 31: bc; 1-30 hosts]

S2: 192.205.7.32/27 -> 32-63 [32: rete, 63: bc; 33-62 hosts]

Variable Length Subnet Mask (VLSM)

Proviamo a dividere la sottorete 193.205.7.64/26 in modo da avere 3 reti:

Subnet A (SA): 28 hosts

Subnet B (SB): 12 hosts

Subnet C (SC): 12 hosts

Subnet root: 193.205.7.64/26

Netmask: 11111111.11111111.11111111.11000000

Variable Length Subnet Mask (VLSM)

Proviamo a dividere la sottorete 193.205.7.64/26 in modo da avere 3 reti:

Subnet A (SA): 28 hosts

Necessitiamo di almeno 5 bit hosts

11111111 11111111 11111111 11100000

8	8	8	3	27
---	---	---	---	----

*SA1: 193.205.7.64/27 -> 64-95 [64: rete, 95: bc; 65-94 hosts]

SA2: 193.205.7.96/27 -> 96-127 [96: rete, 127: bc; 97-126 hosts]

Variable Length Subnet Mask (VLSM)

Subnet B (SB): 12 hosts

Due strade:

1. Usiamo una sottorete differente
2. Subnet della Subnet A

Necessitiamo di almeno 4 bit hosts (1 bit subnet)

11111111 11111111 11111111 11110000

8

8

8

4

28

Subnet di SA2

SA1: 193.205.7.96/28 -> 96-111 [96: rete, 111: bc; 97-110 hosts]

SA2: 193.205.7.112/28 -> 112-127 [112: rete, 127: bc; 113-126 hosts]

Variable Length Subnet Mask (VLSM)

Subnet B e Subnet C vengono assegnate a Subnet A (SA2)

Subnet B:

SA1: 193.205.7.96/28 -> 96-111 [96: rete, 111: bc; 97-110 hosts]

Subnet C:

SA2: 193.205.7.112/28 -> 112-127 [112: rete, 127: bc; 113-126 hosts]

Variable Length Subnet Mask (VLSM)

Riassumendo...

Subnet	Address	Broadcast	Origin
S1	193.205.7.0/27	193.205.7.31/27	193.205.7.0/26
S2	193.205.7.32/27	193.205.7.63/27	193.205.7.0/26
SA1	193.205.7.64/27	193.205.7.95/27	193.205.7.64/26
SA2	193.205.7.96/27	193.205.7.127/27	193.205.7.64/26
SB	193.205.7.96/28	193.205.7.111/28	193.205.7.96/27
SC	193.205.7.112/28	193.205.7.127/28	193.205.7.96/27

Comando **ifconfig** configura la scheda di rete

- * abilita/disabilita (up/down)
- * definisce i parametri hardware (es. irq, io_addr)
- * definisce i parametri della sk di rete (indirizzo, netmask,...)
- * abilita funzionalità particolare (modo promiscuo – promisc)

```
[host]# ifconfig
```

```
eth0  Link encap:Ethernet HWaddr 00:06:29:55:15:4C
```

```
inet addr:193.205.7.146 Bcast:193.205.7.255 Mask:255.255.255.0
```

```
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
```

```
RX packets:24559513 errors:0 dropped:0 overruns:0 frame:0
```

```
TX packets:32527633 errors:0 dropped:0 overruns:0 carrier:0
```

```
collisions:0 txqueuelen:100
```

```
Interrupt:16 Base address:0x2200
```

Statistiche d'uso

Configurazione IP

Configurazione HW

Max Transfer Unit

```
lo  Link encap:Local Loopback
```

```
inet addr:127.0.0.1 Mask:255.0.0.0
```

```
UP LOOPBACK RUNNING MTU:3924 Metric:1
```

```
RX packets:3803246 errors:0 dropped:0 overruns:0 frame:0
```

```
TX packets:3803246 errors:0 dropped:0 overruns:0 carrier:0
```

```
collisions:0 txqueuelen:0
```

Route: il comando **route** permette di gestire (**del**, **add**) le tabelle di routing su un host

• **route add default gw 193.205.7.1**

Connessione diretta
(nessun gateway)

```
[host]# route -n
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags		Metric	Ref	Use	Iface
193.205.7.146	0.0.0.0	255.255.255.255	UH	0	0	0	eth0	
193.205.7.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0	
127.0.0.0	0.0.0.0	255.0.0.0	U	0	0	0	lo	
0.0.0.0	193.205.7.1	0.0.0.0	UG	0	0	0	eth0	

Default route

Use gateway

distanza dalla
destinazione
(in hops)

interfaccia
usata

ICMP (***Internet Control Message Protocol***, definito in RFC 792)

I pacchetti ICMP sono impiegati per:

Segnalazione di errore (inviati al mittente)

- ***Destination Unreachable*** – Il pacchetto non può essere consegnato
- ***Time exceeded – TTL*** ha raggiunto il valore 0 oppure non si sono ricevute tutte le parti di un pacchetto frammentato entro un tempo limite
- ***Parameter Problem*** – Errore nei parametri dell'header
-

Messaggi di richiesta

- ***Echo request/Echo reply*** – Richiesta/risposta di eco (verifica se un host risponde)
- ***Timestamp request/Timestamp reply*** – Richiesta/risposta con timestamp
-

Traceroute: mostra i nodi attraversati lungo il percorso verso una data destinazione.

Sfrutta il messaggio di notifica ICMP `TIME_EXCEEDED` inviato dai gateway quando il `TIME TO LIVE` scade.

- * Per default sono inviati 3 pacchetti “sonda” per ogni TTL e viene mostrato il tempo di round-trip per ciascuno di essi.
- * Il messaggio di notifica è atteso per un tempo massimo (5 s) altrimenti stampa “*” per il relativo pacchetto sonda.

Il numero massimo di hop è 30 (come TCP).

- * Non tutti i gateway inviano il `TIME_EXCEEDED`.

Un esempio del comando route

[host]# **traceroute www.virgilio.it**

traceroute to www.virgilio.it (212.48.10.150), 30 hops max, 38 byte packets

```
1  ringe (193.205.7.1)  2.685 ms  1.665 ms  1.662 ms
2  cuces-ing.link.unisi.it (193.205.5.101)  70.407 ms  32.951 ms  70.588 ms
3  rcuces.unisi.it (193.205.4.1)  43.928 ms  130.064 ms  125.421 ms
4  rc-unisi.fi.garr.net (193.206.136.89)  125.552 ms  119.365 ms  103.817ms
5  bo-fi.garr.net (193.206.134.93)  70.092 ms  38.814 ms  20.001 ms
6  rtg-rt-1.bo.garr.net (193.206.134.198)  19.653 ms  105.629 ms  14.991 ms
7  rm-bo-g.garr.net (193.206.134.49)  32.389 ms  60.269 ms  25.866 ms
8  cw-nap.inroma.roma.it (194.242.224.3)  45.573 ms  54.386 ms  88.611 ms
9  s3-0-r2-ROM2.cwitaly.net (195.94.140.173)  37.144 ms  27.868 ms  42.471 ms
10 194.79.199.35 (194.79.199.35)  26.232 ms  25.961 ms  33.669 ms
11 pos0-1-0-r5-MIL3.cwitaly.net (194.79.192.5)  49.560 ms  33.136 ms  81.481 ms
12 fe0-0-r3-MIL1.cwitaly.net (213.233.14.75)  164.763 ms  71.242 ms  107.814 ms
13 matrix-mi.ar1.mi1.cwitaly.net (195.250.255.102)  45.349 ms  79.267 ms  *
14 212.48.4.241 (212.48.4.241)  71.192 ms  59.393 ms  43.822 ms
15 212.48.4.174 (212.48.4.174)  59.138 ms  83.495 ms  39.530 ms
16 * * *
17 * * *
```

Destination unreachable

E' inviato da un Router che non riesce ad instradare un pacchetto, oppure dall'Host destinatario che non riesce a consegnarlo.

Oltre al codice che indica il motivo (host/rete non raggiungibile, porta TCP non aperta,...) al messaggio è allegato l'inizio del pacchetto, necessario per capire a chi notificare l'errore

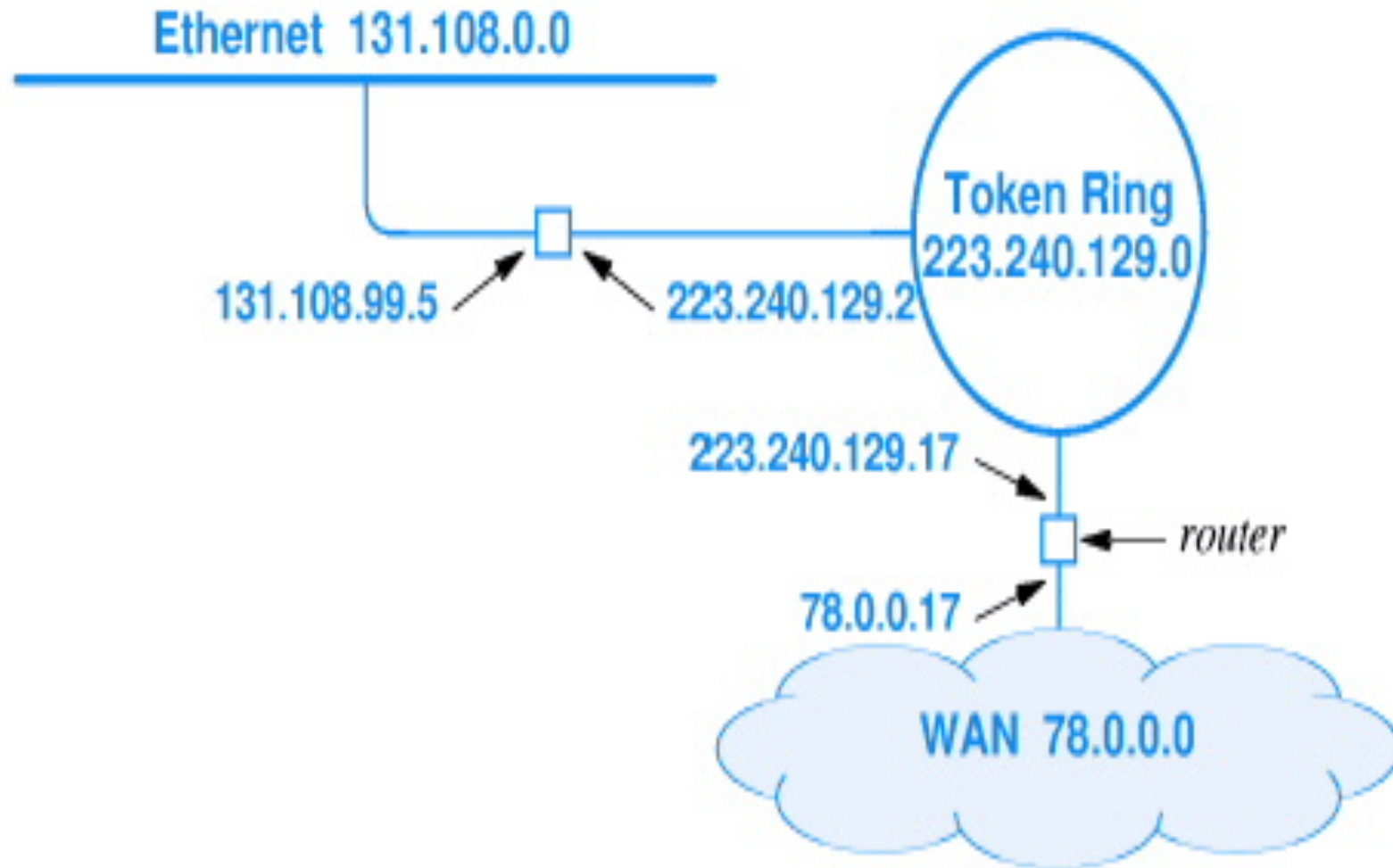
Tipo: 3	Codice: 0-15	checksum
0 (non usato)		
Header e primi 8 byte del datagram IP ricevuto		

Ping: invia pacchetti ICMP ECHO REQUEST ad un host o gateway ed attende le risposte ICMP ECHO REPLY. Il pacchetto contiene un timestamp per misurare il tempo di round-trip.

```
[host]# ping -v cuces.unisi.it
PING cuces.unisi.it (193.205.4.2): 56 data bytes
64 bytes from 193.205.4.2: icmp_seq=0 ttl=252 time=28.6 ms
64 bytes from 193.205.4.2: icmp_seq=1 ttl=252 time=45.1 ms
64 bytes from 193.205.4.2: icmp_seq=2 ttl=252 time=61.8 ms
64 bytes from 193.205.4.2: icmp_seq=3 ttl=252 time=62.8 ms
64 bytes from 193.205.4.2: icmp_seq=4 ttl=252 time=78.0 ms

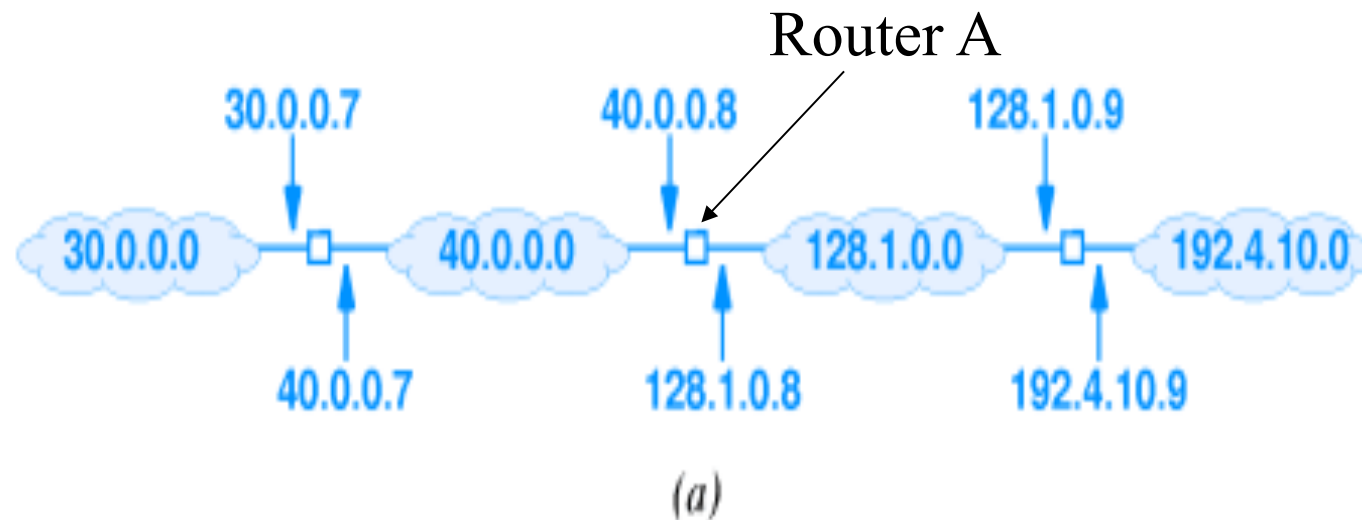
--- cuces.unisi.it ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 28.6/55.2/78.0 ms
```

Routers e Indirizzamento



Il prefisso identifica la rete

È necessario un indirizzo per la connessione del router



Tab. del Router A

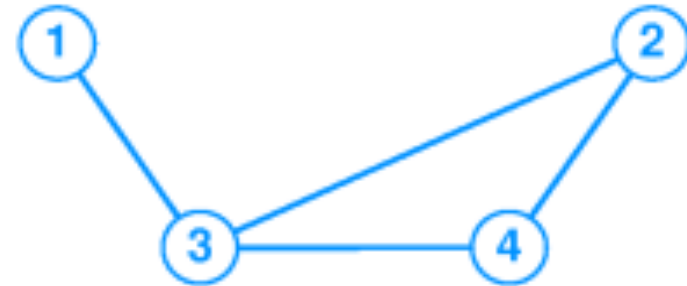
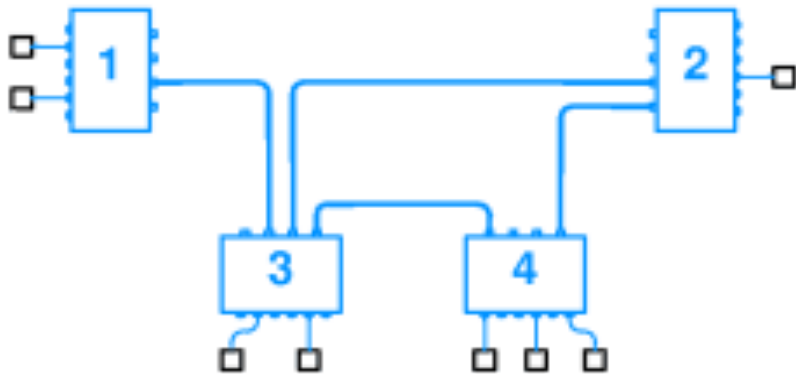
Destination	Mask	Next Hop
30.0.0.0	255.0.0.0	40.0.0.7
40.0.0.0	255.0.0.0	deliver direct
128.1.0.0	255.255.0.0	deliver direct
192.4.10.0	255.255.255.0	128.1.0.9

(b)

I router non memorizzano le informazioni riguardo gli host presenti nella varie sotto-reti.

Tabelle di Routing

Il numero di elementi nelle routing tables è proporzionale al numero delle reti nell'internetwork:



destin- ation	next hop
1	-
2	(1,3)
3	(1,3)
4	(1,3)

node 1

destin- ation	next hop
1	(2,3)
2	-
3	(2,3)
4	(2,4)

node 2

destin- ation	next hop
1	(3,1)
2	(3,2)
3	-
4	(3,4)

node 3

destin- ation	next hop
1	(4,3)
2	(4,2)
3	(4,3)
4	-

node 4

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>	

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

Default Routing: semplifica notevolmente la tabella

Routing/Forwarding

Dato un datagramma:

- * Estrae il campo destinazione DA (Dest. address)*
- * Cerca DA nella routing table*
- * Trova il prossimo “hop address”: HA*
- * Spedisce il datagramma a HA*

Nell'header del datagramma l'indirizzo del destinatario è sempre quello della destinazione ultima, e no quello del “next hop”.