

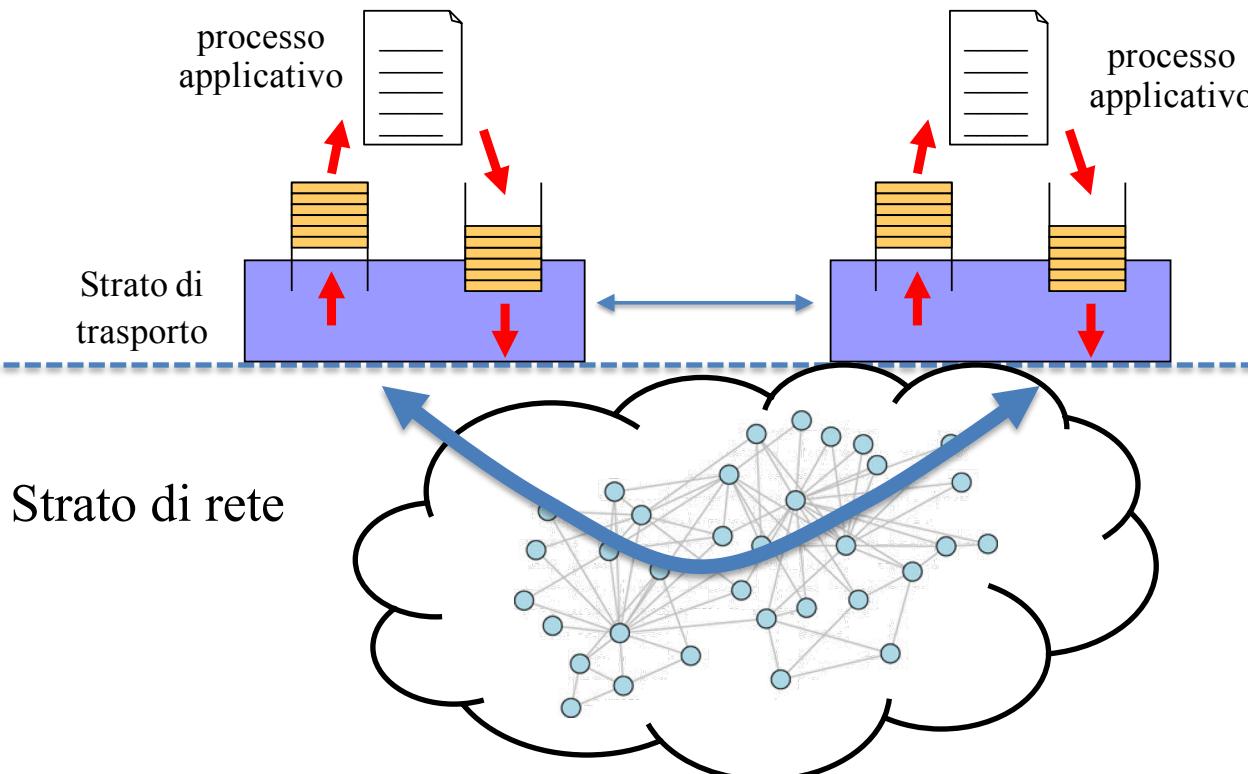


## **4 - Livello Rete (parte A)**

**Introduzione, indirizzamento, inoltro**

# Strato di rete e strato di trasporto

- Lo strato di **trasporto** realizza la comunicazione tra due **processi**
- Lo strato di **rete** si incarica di trasferire i dati tra gli **host** che ospitano i due processi comunicanti



# Architettura di protocolli TCP/IP

Application layer

Transport layer

Network layer

(Data-link layer)  
(non specificato in  
Internet)

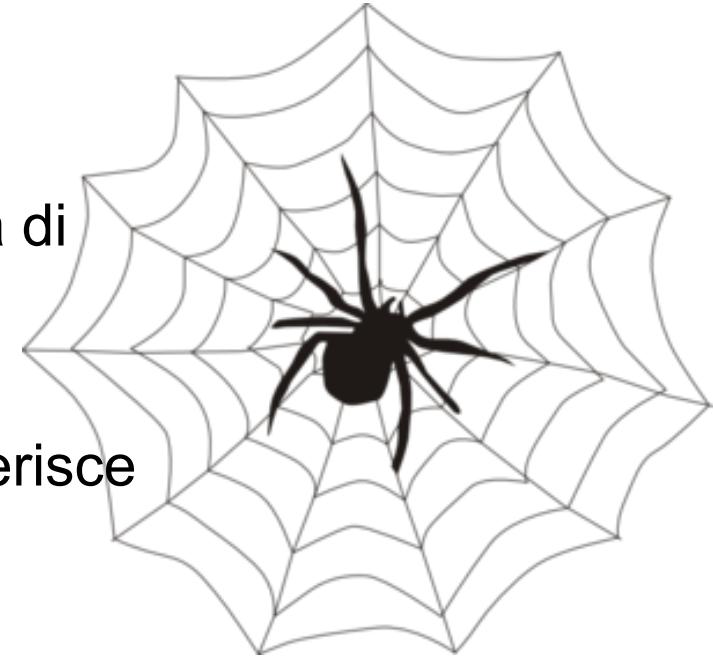
|                |      |     |      |     |      |          |
|----------------|------|-----|------|-----|------|----------|
| Telnet         | HTTP | FTP | SMTP | BGP | SNMP | RIP      |
| TCP            |      |     |      |     | UDP  |          |
| ICMP           | OSPF |     | IP   |     |      | ARP RARP |
| Network access |      |     |      |     |      |          |

- **Piano Dati (*Data Plane*)**: protocolli per trasferire i dati d'utente
  - IP
- **Piano di Controllo (*Control Plane*)**: protocolli di segnalazione per supportare il trasferimento dati
  - ICMP, ARP, RARP, OSPF, RIP



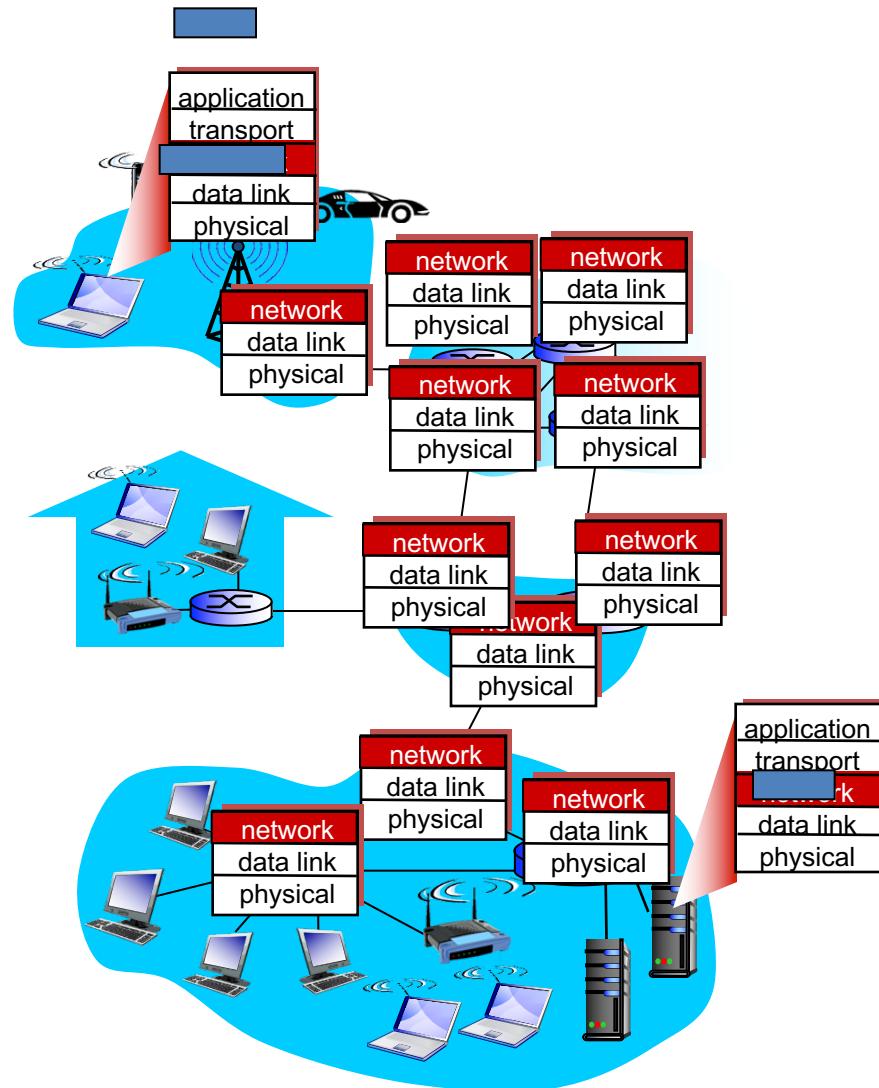
# Funzioni fondamentali dello strato di rete

- **Indirizzamento**
  - Identificazione univoca dell'interfaccia di rete di un *host/router*
- **Inoltro / Forwarding:**
  - Funzione locale con cui il router trasferisce i pacchetti dall'ingresso all'uscita
- **Instradamento / Routing:**
  - Processo che determina i percorsi dei pacchetti dalla sorgente alla destinazione
  - Processo globale svolto dagli **Algoritmi di Routing**
    - Processo centralizzato vs distribuito, statico vs dinamico, manuale vs dinamico



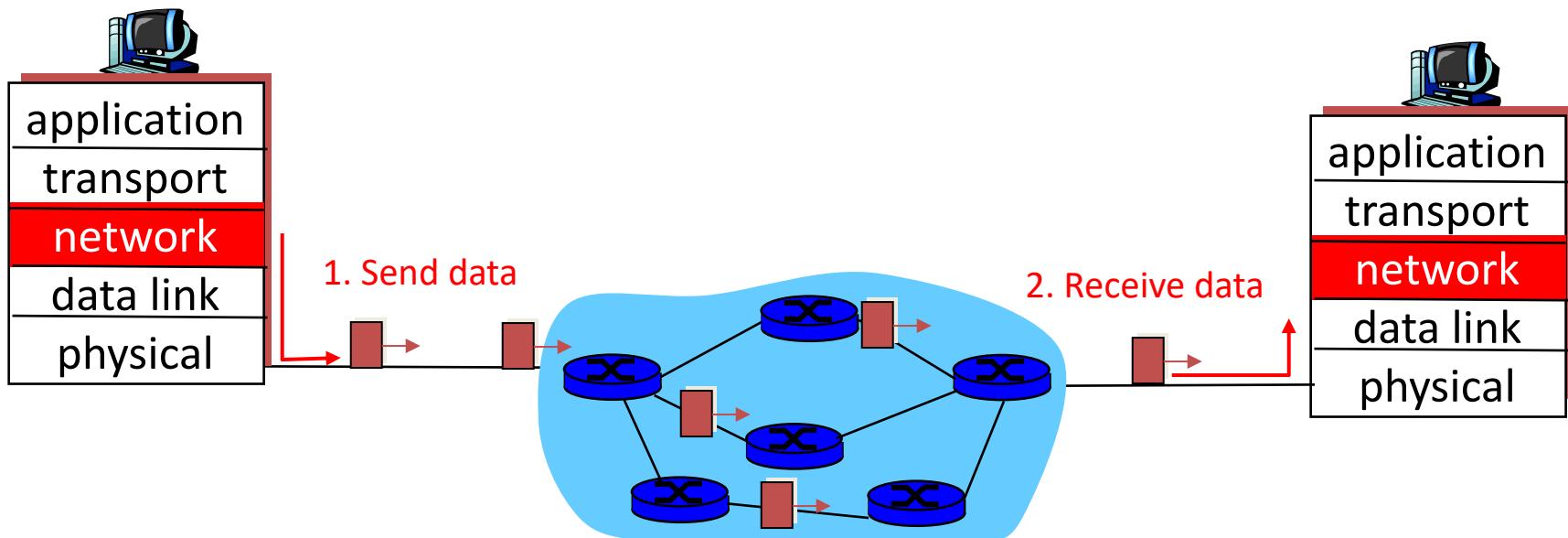
# Data Plane a livello di rete

- I segmenti dello strato di trasporto vengono trasferiti dallo strato di rete dall'*host* sorgente all'*host* destinazione
  - Lato sorgente, vengono incapsulati in datagram
  - I *datagram* vengono inoltrati *hop-by-hop* fino a destinazione
  - I *router* esaminano i campi dell'*header* di ciascun *datagram IP* che li attraversa
  - Lato destinazione, i segmenti vengono consegnati allo strato di trasporto
- I protocolli dello strato di rete sono implementati in ogni host e in ogni router***

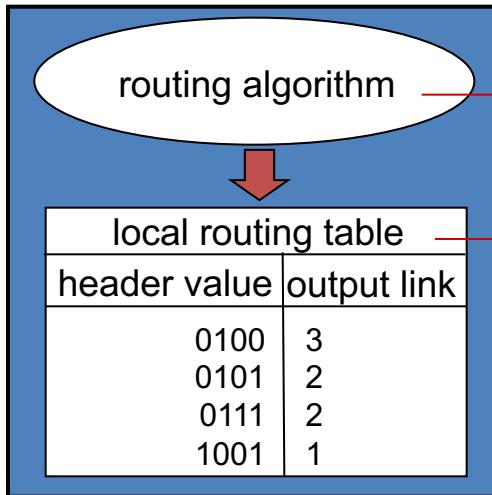


# Data Plane a livello di rete: caratteristiche

- Servizio di trasferimento dell'informazione
  - *Best Effort* e senza connessione
- Ogni router che riceve un datagramma legge *l'header* e decide come/dove inoltrare il datagramma sulla base di:
  - Un “indirizzo di destinazione” presente *nell'header* del datagramma
  - Una tabella di instradamento presente in ogni nodo
- I pacchetti possono (potenzialmente) percorrere strade diverse tra sorgente e destinazione



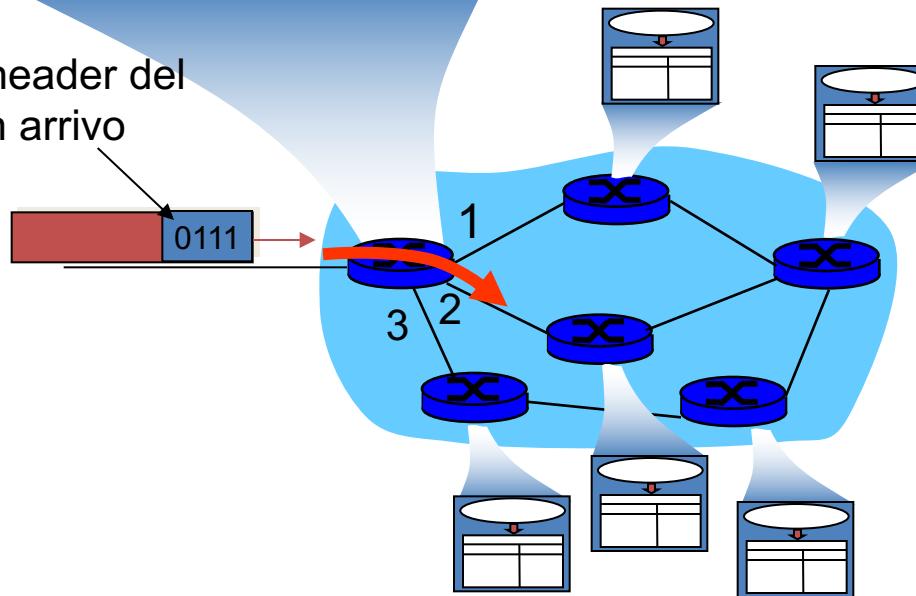
# Funzioni base svolte dal router (richiamo)



Gli algoritmi di routing determinano il percorso end-to-end attraverso la rete

La tabella di routing governa l'attraversamento (local forwarding) di questo router

Valore nell'header del pacchetto in arrivo



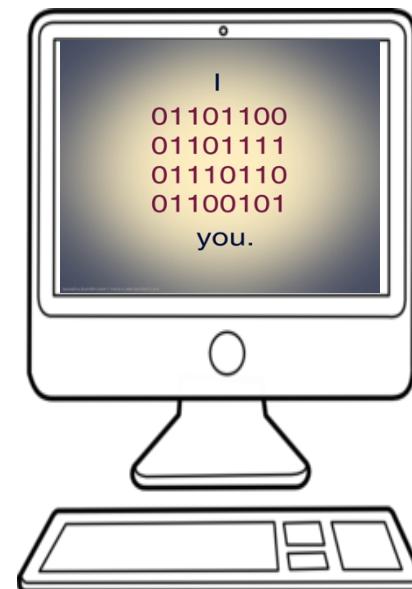
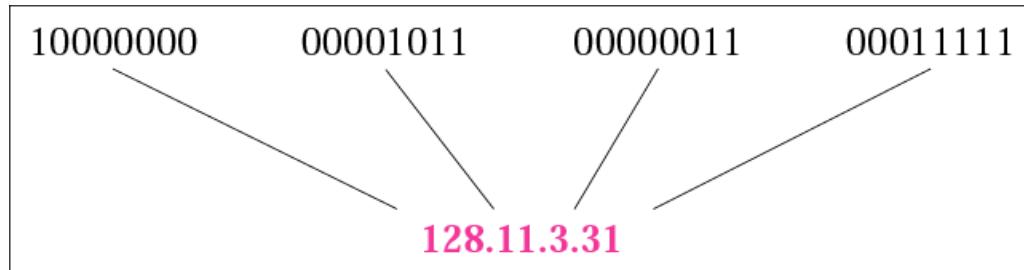
# Agenda

- Introduzione agli indirizzi IPv4
- Indirizzamento IP
  - Indirizzamento CIDR (*Classless Inter Domain Routing*)
  - *Subnetting*
- Inoltro dei pacchetti IP
  - Aggregazione di indirizzi (*supernetting*)



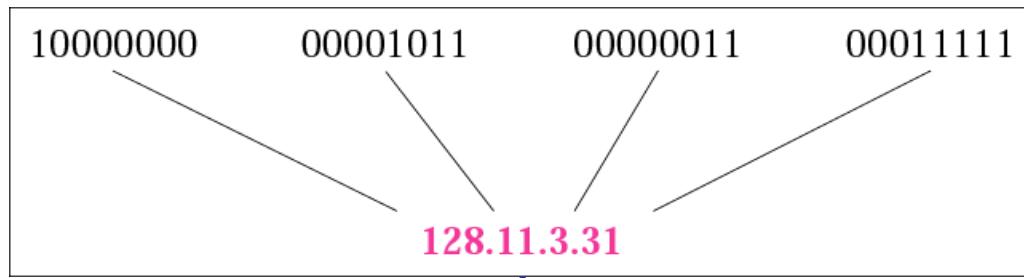
# Indirizzo IP (IPv4)

- È un numero binario di 32 bit
- “A beneficio degli esseri umani”, viene scritto nella forma x.y.z.w.  
Ciascuno dei x,y,z,w
  - rappresenta 8 bit
  - può assumere tutti i valori da 00000000 a 11111111 (0 – 255 in base 10)
- Sempre “a beneficio degli esseri umani” si usa scriverlo nella forma decimale (notazione decimale puntata)

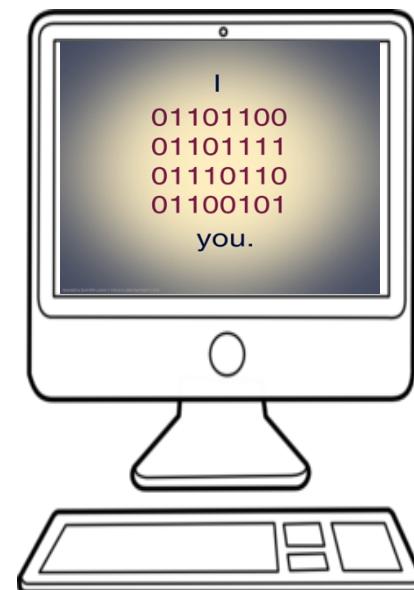


# Indirizzo IP (IPv4)

- È associato in modo univoco ad un'interfaccia di rete di un *host* o di un router
  - Non è associato direttamente a un *host* o a un router perché questi possono avere più interfacce di rete
- Indirizzo IP deve avere valenza e univocità universali (in tutto Internet)
  - Il *routing* in IP è basato sull'indirizzo dell'*host* destinazione
- Ogni gestore di rete ha a disposizione un blocco di indirizzi che distribuisce alle interfacce dei singoli apparati



“Dotted decimal notation”



# IP Network, Network prefix (NetID)

- Un blocco elementare di indirizzi IP è assegnato alle interfacce di una **rete IP**
- Gli indirizzi del blocco devono avere identici i primi  $n$  bit
- Questi primi  $n$  bit si chiamano **prefisso** (o **identificativo**) di rete (**network prefix** o **NetID**)
- L'indirizzo IP è dunque diviso in due campi (o livelli)
  - I primi  $n$  bit (*prefix* o *NetID*) identificano la rete
  - I rimanenti bit (**HostID**) sono usati per identificare un host specifico (un'interfaccia) nell'ambito della rete
- Il valore di  $n$  dipende dal tipo di rete



# Agenda

- Introduzione agli indirizzi IPv4
- Indirizzamento IP
  - Indirizzamento CIDR (*Classless Inter Domain Routing*)
  - Subnetting
- Inoltro dei pacchetti IP
  - Aggregazione di indirizzi (*supernetting*)



# Classless Inter-Domain Routing (CIDR)

- CIDR sviluppato negli anni '90 e introdotto in via definitiva nel settembre 1993
  - RFC 1517, 1518, 1519, 1520
  - x.y.z.w/n significa che viene allocato a una rete un insieme di indirizzi **contigui** con
    - Primo indirizzo: x.y.z.w e  $2^{32-n}$  indirizzi in totale
- Esempio: 134.76.96.0/19 sono 8192 indirizzi IP adiacenti da 134.76.96.0 a 134.76.127.255
- Ha razionalizzato l'assegnazione di indirizzi del poco flessibile sistema *classful* (si veda oltre)

$$0110000 = 96$$

$$2^{32-19} = 2^{13}$$



# Netmask

11111111 11111111 11111111 00000000

- La *netmask* è un numero binario di 32 bit associato ad una rete IP
  - Inizia con  $n$  bit a 1(dal più significativo), con  $n$  pari alla lunghezza del NetID
  - I restanti  $32 - n$  bit sono a 0
  - Indica quali bit di un indirizzo IP sono assegnati al NetID
  - Viene indicata con *dotted decimal notation*
- Esempio:
  - Indirizzo IP 193.17.31.45 e Netmask: 255.255.255.0
  - La rete a cui appartiene l'indirizzo è 193.17.31.0



# NetID: notazioni alternative equivalenti

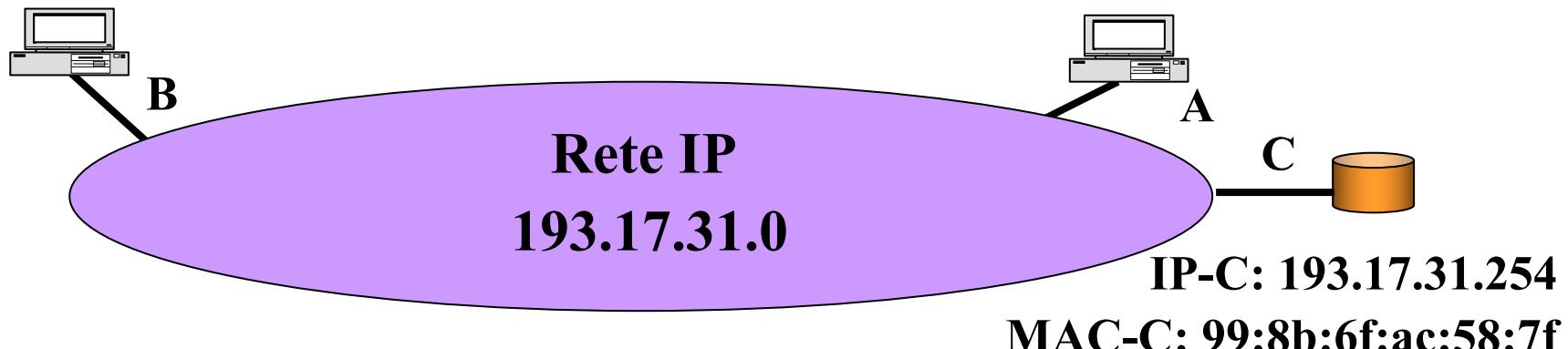
- Esempio: NetID 130.86.0.0
  - Rappresenta tutti i  $2^{16}$  indirizzi IP che iniziano con il pattern di bit 10000010 01010110
- Questo si può scrivere come
  - Coppia: 130.86.0.0 + Netmask 255.255.0.0
  - 130.86.0.0/16
  - 130.86.\*.\*
  - Intervallo (range) [130.86.0.0, 130.86.255.255]



# Rete IP $\leftrightarrow$ rete fisica

IP-B: 193.17.31.55/24

MAC-B: 05:98:76:6c:4a:7b

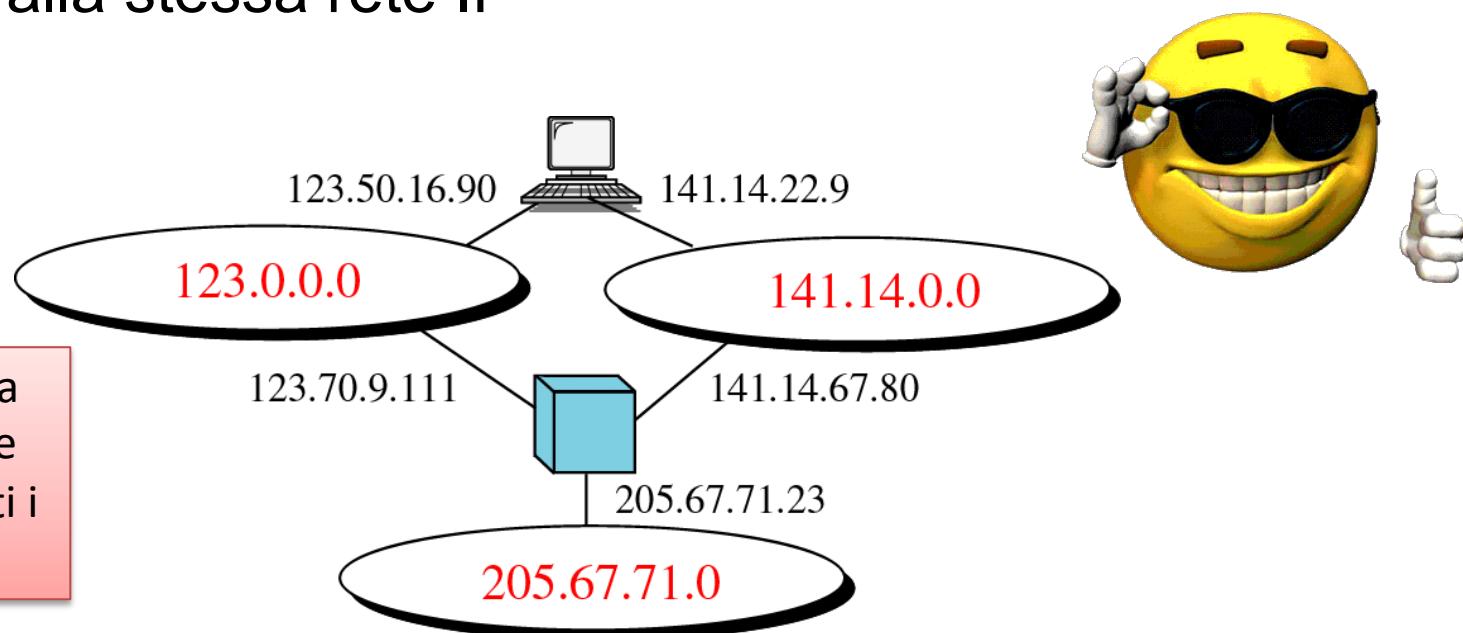


- Una rete IP, identificata da un *NetID*, è una insieme di interfacce (*host + router*) fisicamente interconnesse, tipicamente con *switch* e hub
- E' necessario che vi sia almeno un router con un'interfaccia collegata alla rete IP per comunicare con altre reti IP



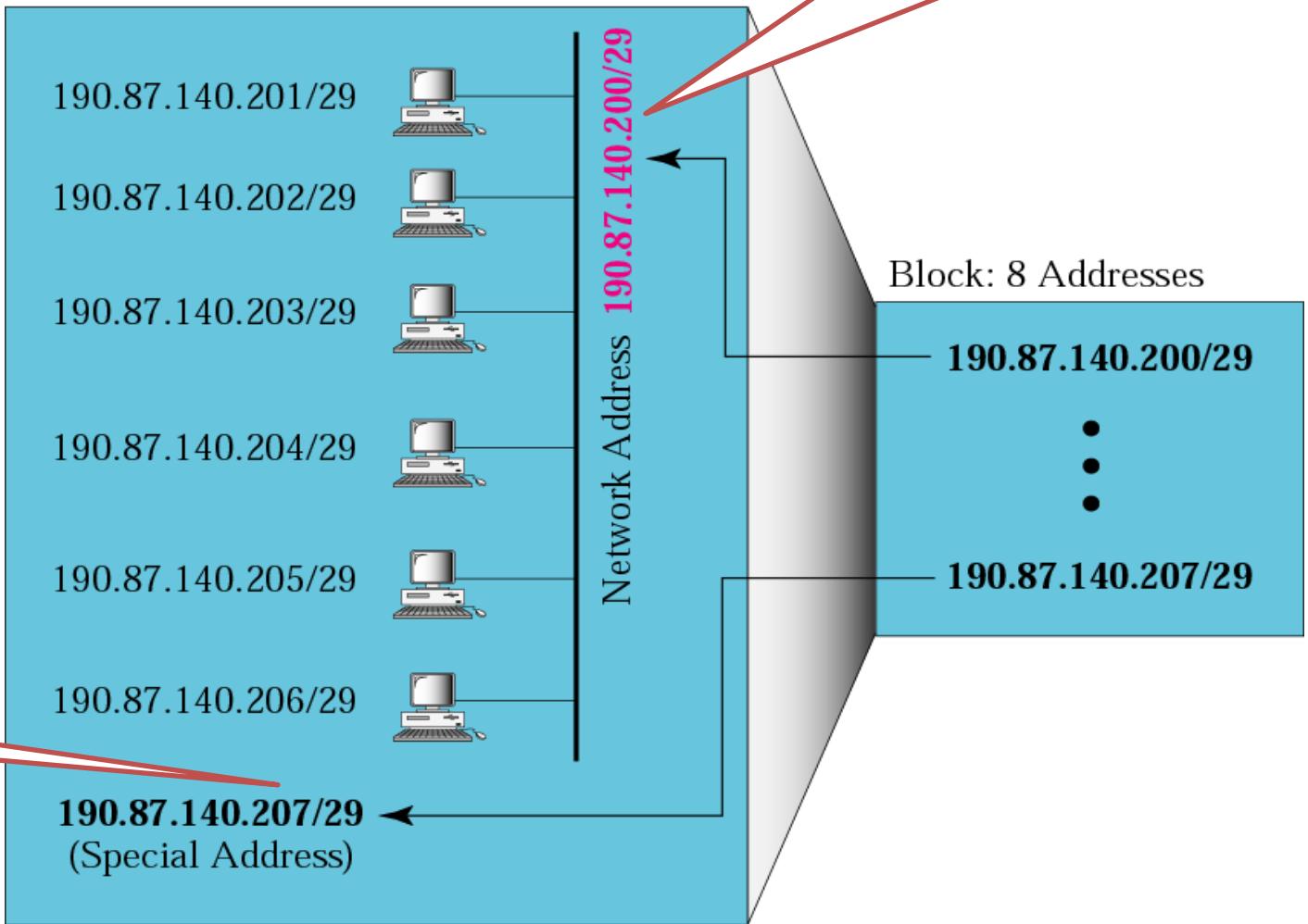
# Esempio

- Notare che anche un *host* può avere più interfacce di rete (*dual homing*)
- Ogni interfaccia di un router (e di un *host*) deve appartenere a reti IP diverse
- Non è possibile assegnare due interfacce dello stesso apparato alla stessa rete IP



# CIDR: esempio

Network Organization



# Indirizzi privati

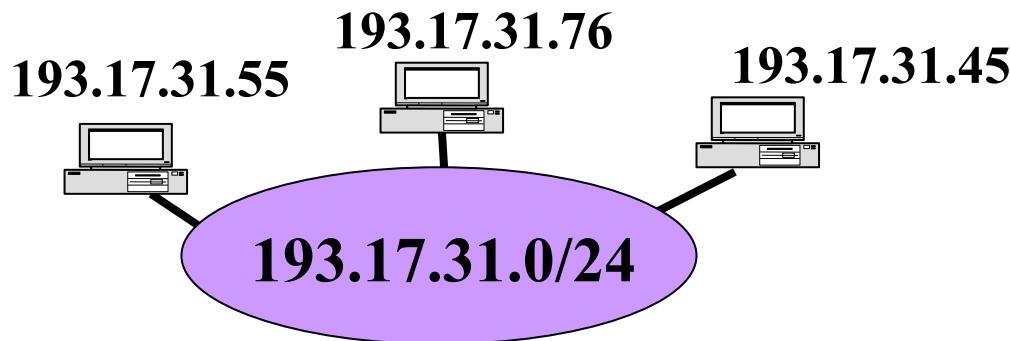
- Utilizzabili da chiunque ma solo in ambito privato. In Internet possono essere riutilizzati più volte → Non sono univoci
  - Tre blocchi (usati anche in CIDR)
    - I: 10.0.0.0 - 10.255.255.255
    - II: 172.16.0.0 -172.31.255.255
    - III:192.168.0.0 -192.168.255.255
- Un router non deve mai inoltrare un pacchetto con destinazione un indirizzo IP privato verso una propria interfaccia di uscita che abbia un indirizzo IP pubblico



# Indirizzi Speciali

## Indirizzo di rete

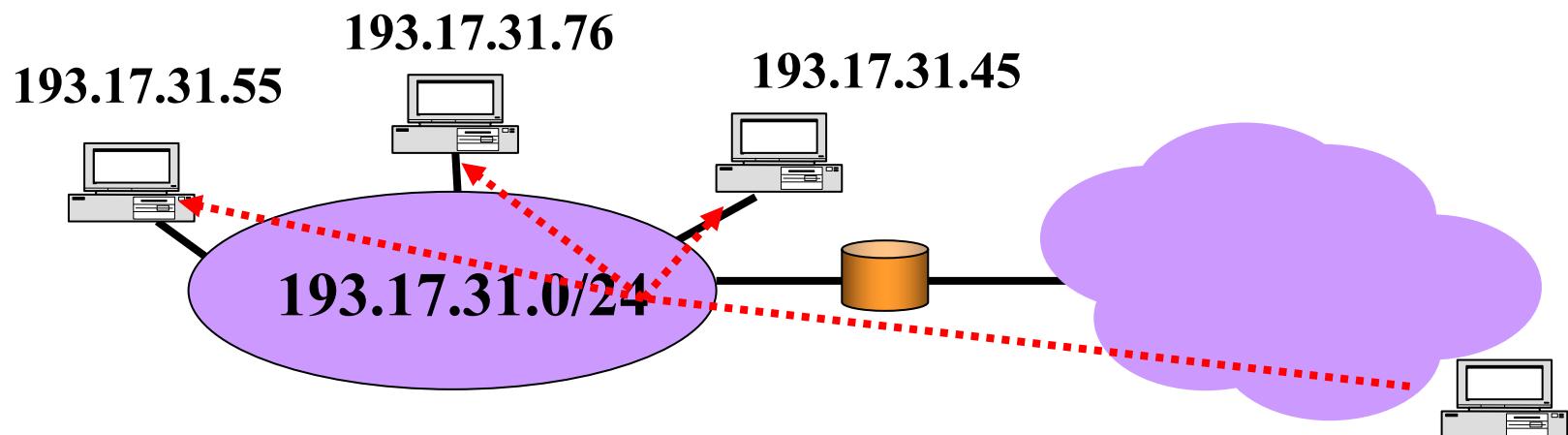
- L'indirizzo con il campo *HostID* posto a 0 serve ad indicare la rete il cui indirizzo è contenuto nel campo *NetID* (usato solo nelle tabelle di instradamento)
- esempio:
  - rete /16: 131.175.0.0
  - rete /24: 193.17.31.0



# Indirizzi Speciali

## Indirizzo broadcast diretto (Directed Broadcast):

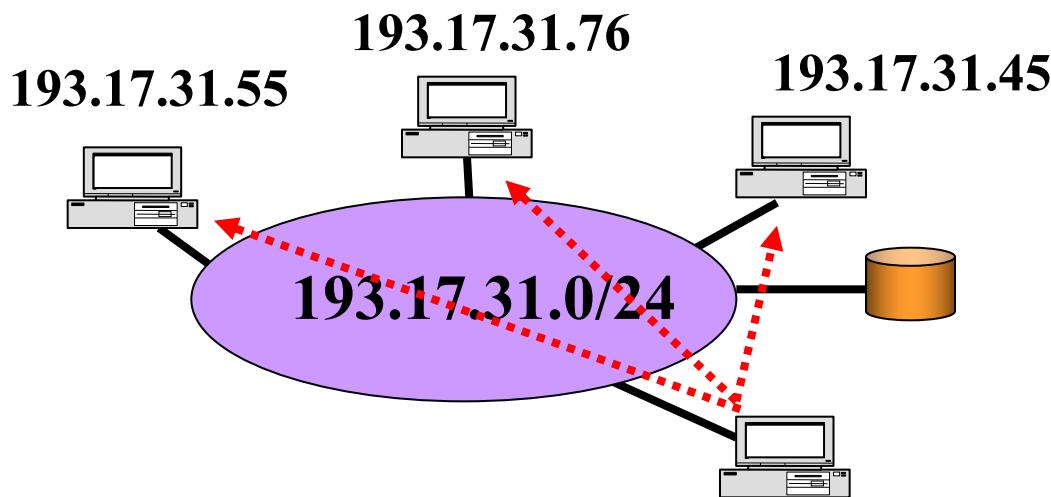
- Un indirizzo con il campo HostID di soli 1 assume il significato di indirizzo broadcast della rete indicata nel campo NetID (usato nel campo destinatario di un pacchetto IP)
- I router di transito lo trattano come un normale pacchetto (inoltrano al next hop)
- Il router della rete di destinazione esegue il broadcast (a livello 2) solo se è abilitato a farlo
- esempio: 193.17.31.255



# Indirizzi Speciali

## Indirizzo broadcast limitato (Limited Broadcast):

- Un indirizzo di soli 1 assume il significato di indirizzo broadcast nella stessa rete di chi invia il pacchetto. Il pacchetto non può oltrepassare il router (usato nel campo destinatario del pacchetto IP) 255.255.255.255



# Indirizzi Speciali

- Quando il campo *NetID* è posto a zero, l'indirizzo indica l'host il cui indirizzo è contenuto nel campo *host* sulla stessa rete del mittente.
  - usato come campo destinazione, filtrato dai router
  - esempio: 0.0.21.173 (in una rete /16)
- Se anche il campo *host* è posto a zero l'indirizzo indica il mittente stesso del pacchetto
  - Usato come campo sorgente quando l'host non conosce il proprio indirizzo, ad es. in fase di bootstrap
  - esempio: 0.0.0.0
- Infine, l'indirizzo con il primo ottetto pari a 127 e gli altri campi qualsivoglia indica il *loopback* sullo stesso *host*
  - (usato nei sistemi operativi per testare le funzionalità di rete).
  - esempio: 127.x.y.z



# Procedure di assegnamento degli indirizzi



ICANN, Los Angeles, USA

## ***Internet Assigned Numbers Authority (IANA)***

- Coordina e pianifica l'assegnazione degli indirizzi IP su base mondiale (sia per IPv4 che per IPv6)
- E' un Dipartimento di ICANN (Internet Corporation for Assigned Names and Numbers, organizzazione no-profit americana), istituito nel 1998
- Amministra anche i DNS root

IANA assegna blocchi di indirizzi a 5 Regional Internet Registries (RIRs)

- Tipicamente alloca indirizzi delle dimensioni di /8 ai RIR che li richiedono



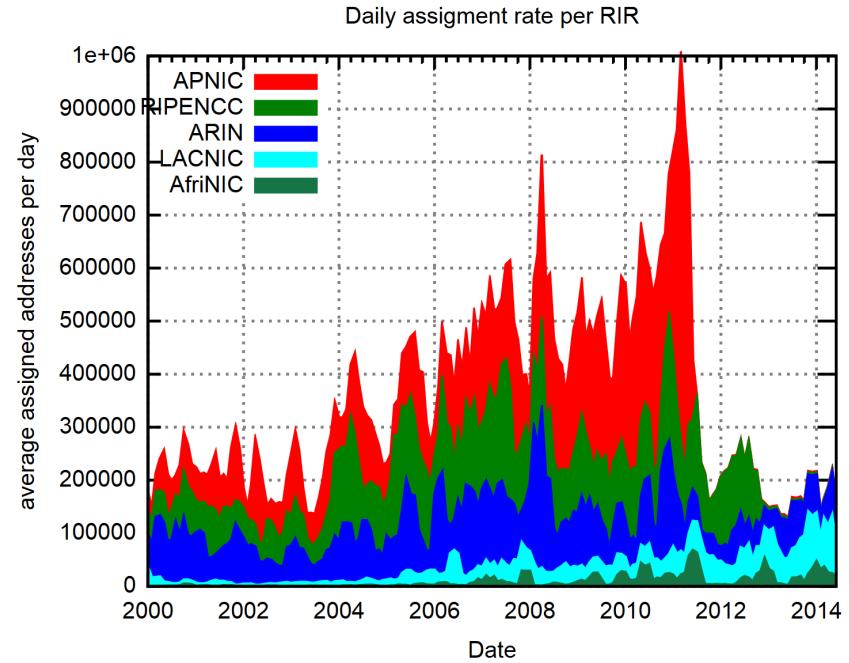
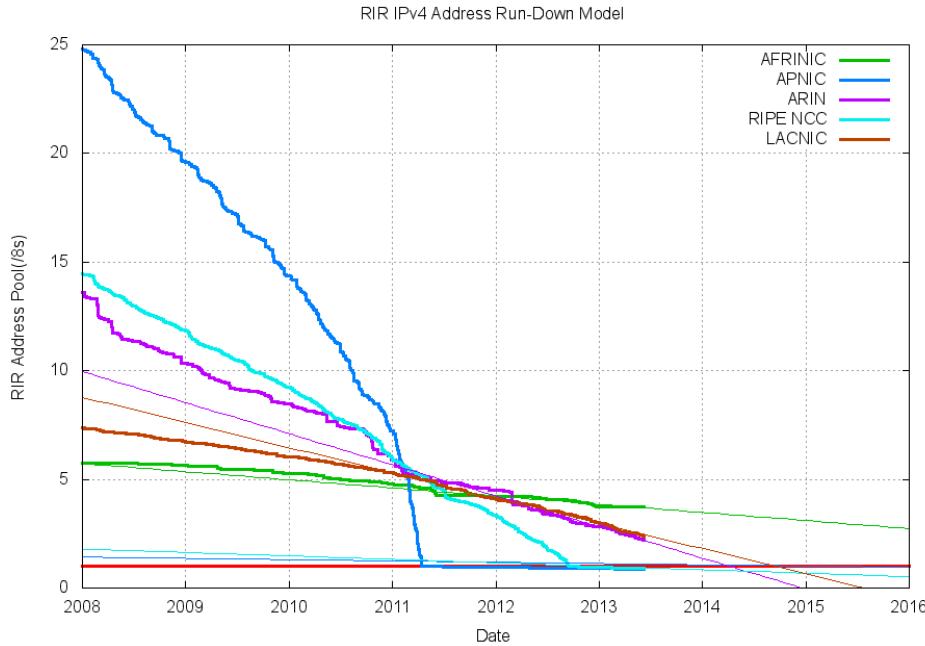
# Procedure di assegnamento degli indirizzi

- I RIR sono:
  - African Network Information Centre (AfriNIC): Africa
  - American Registry for Internet Numbers (ARIN): Stati Uniti, Canada, molti stati caraibici e Antartide.
  - Asia-Pacific Network Information Centre (APNIC): Asia, Australia, Nuova Zelanda e stati limitrofi
  - Latin America and Caribbean Network Information Centre (LACNIC): America Latina e stati caraibici rimanenti
  - Réseaux IP Européens Network Coordination Centre (RIPE NCC): Europa, Russia, Medio Oriente e Asia Centrale
- Ciascun RIR assegna blocchi al Local Internet Registries (LIRs), che sono ISP o istituzioni che a loro volta possono assegnare prefissi di rete ai propri clienti

Lista completa: <http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml>



# Esaурimento degli indirizzi IPv4

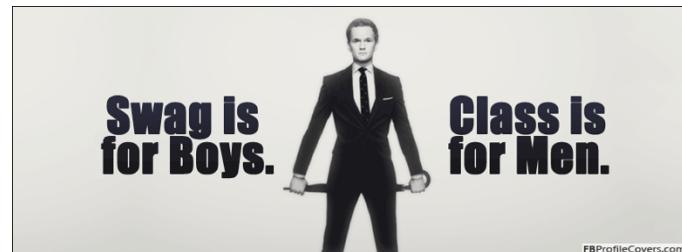


- Tutti gli indirizzi IPv4 sono ormai ESAURITI!!
  - IANA ha assegnato tutti gli indirizzi ai RIR nel 2011
  - Quasi ogni RIR ha terminato gli indirizzi disponibili
- L'utilizzo di soli 32 bit per gli indirizzi IPv4 si è rivelata una scelta sbagliata
  - Nessuno poteva aspettarsi negli anni '70-80 l'attuale sviluppo di Internet
- L'unica possibilità di assegnare nuovi indirizzi è passare a IPv6 (indirizzi lunghi 128 bit →  $2^{128}$  indirizzi)

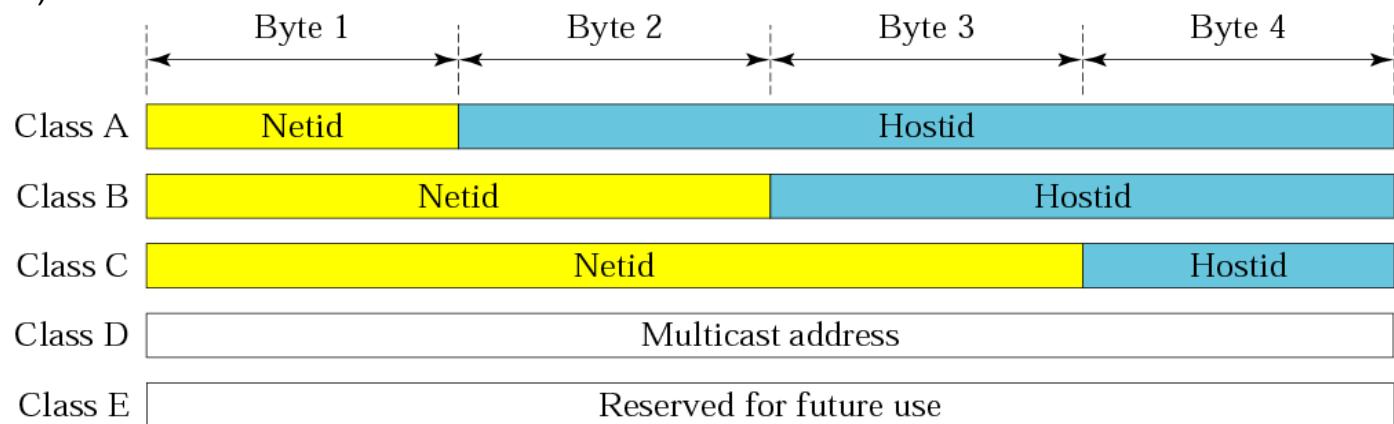


# Indirizzamento classful

- Gli indirizzi IPv4 erano divisi in 5 classi:
  - Classe A: poche reti grandi:
    - NetID 7 bit ( $x=1,\dots,126$ ), HostID 24 bit
  - Classe B: reti medio-grandi:
    - NetID 14 bit ( $x=128,\dots,191$ ), HostID 16 bit
  - Classe C: tante reti piccole:
    - NetID 21 bit ( $x=192,\dots,223$ ), HostID 8 bit
  - Classe D: indirizzi multicast:
    - ( $x=224,\dots,239$ )
  - Classe E: riservato a uso futuro
    - ( $x=240,\dots,255$ )



|         | First byte | Second byte | Third byte | Fourth byte |
|---------|------------|-------------|------------|-------------|
| Class A | 0          |             |            |             |
| Class B | 10         |             |            |             |
| Class C | 110        |             |            |             |
| Class D | 1110       |             |            |             |
| Class E | 1111       |             |            |             |



# Classful vs. CIDR

- Nel sistema classful i router deducono la lunghezza del prefisso dai primi bit del primo byte:
  - Non occorre la netmask per ciascuna entry della tabella di routing. In CIDR è fondamentale
  - Non occorre che i protocolli di routing supportino la netmask nei messaggi. In CIDR è necessario
- Problemi principali delle classi
  - Le reti di classe A e B sono troppo grandi
    - Reti di classe B con meno di 100 host effettivi sono frequenti
  - Le reti di classe B sono troppo poche
  - Le reti di classe C sono troppo piccole
- Eliminando completamente la distinzione tra reti di classe A, B e C, CIDR risolve il problema



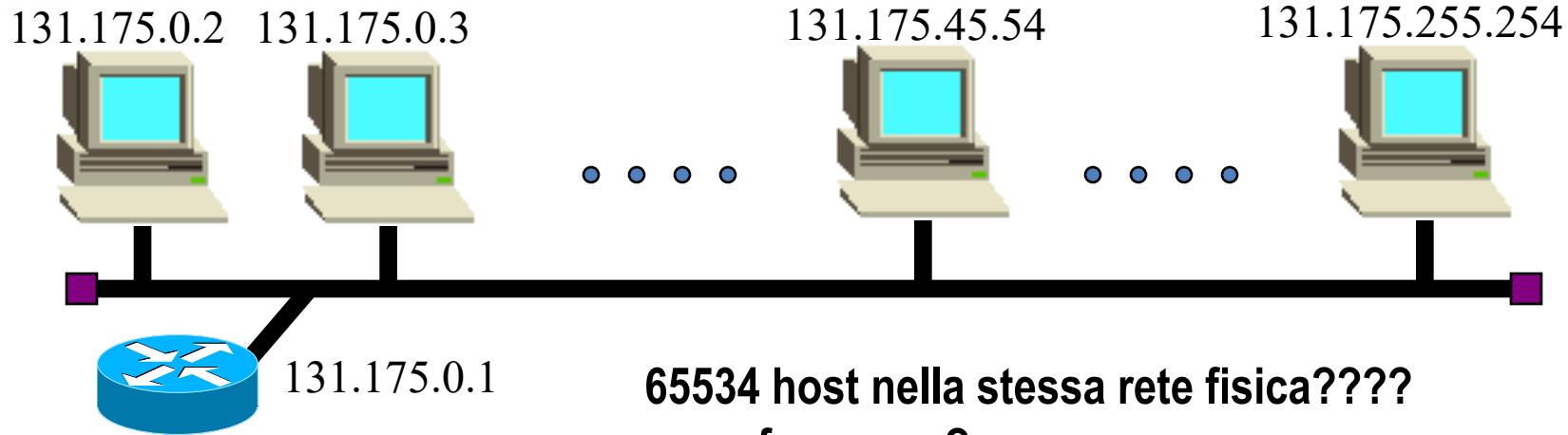
# Agenda

- Introduzione agli indirizzi IPv4
- Indirizzamento IP
  - Indirizzamento CIDR (*Classless Inter Domain Routing*)
  - **Subnetting**
- Inoltro dei pacchetti IP
  - Aggregazione di indirizzi (*supernetting*)



# Problema: reti IP troppo grandi

- Le reti IP possono essere troppo grandi
  - Es. Le reti di /16 (classe B) e /8 (classe A) sono poco adatte a corrispondere ad uniche reti di livello 2



Rete /16:

Da: 131.175.0.1

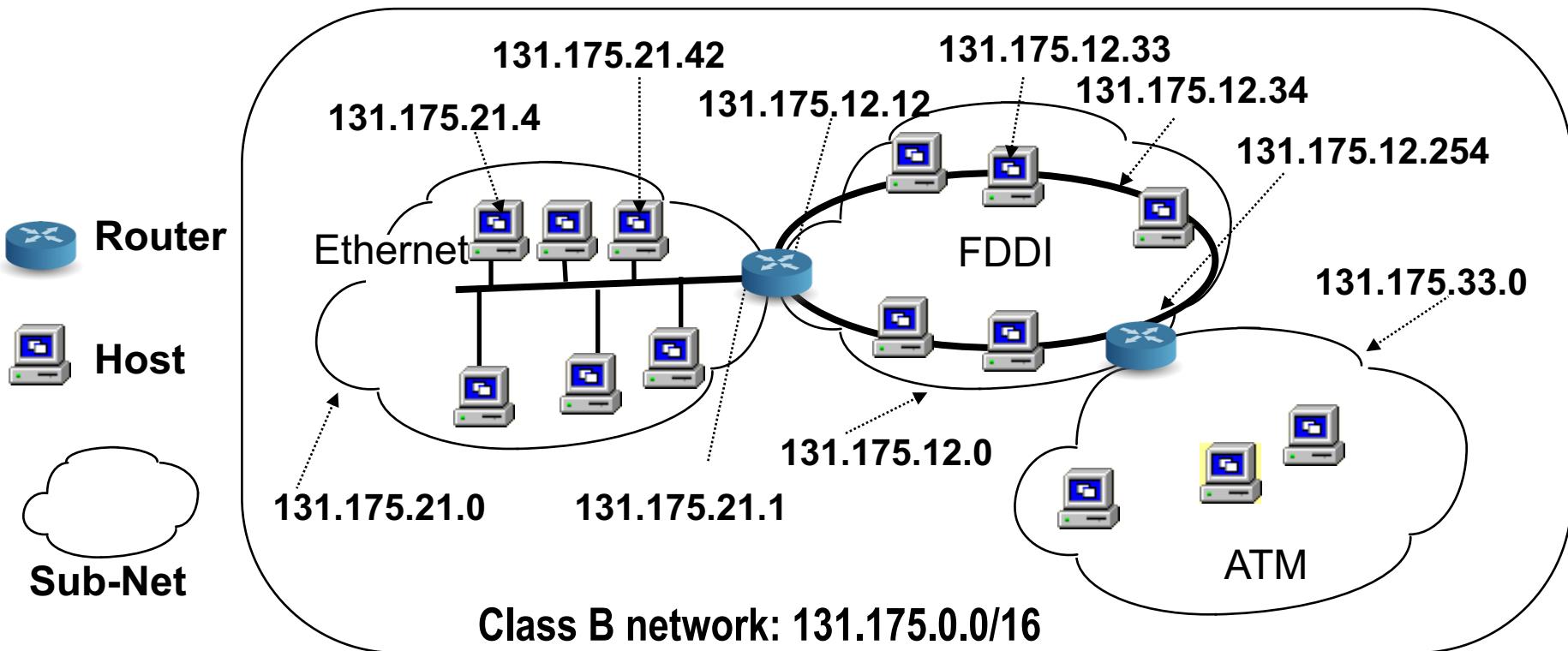
A: 131.175.255.254

**65534 host nella stessa rete fisica????**  
- performance?  
- management?



# Soluzione: introdurre una gerarchia nelle reti IP

- Suddividere la rete in molte sottoreti (*subnet*)
- Ciascuna *subnet* = una rete fisica

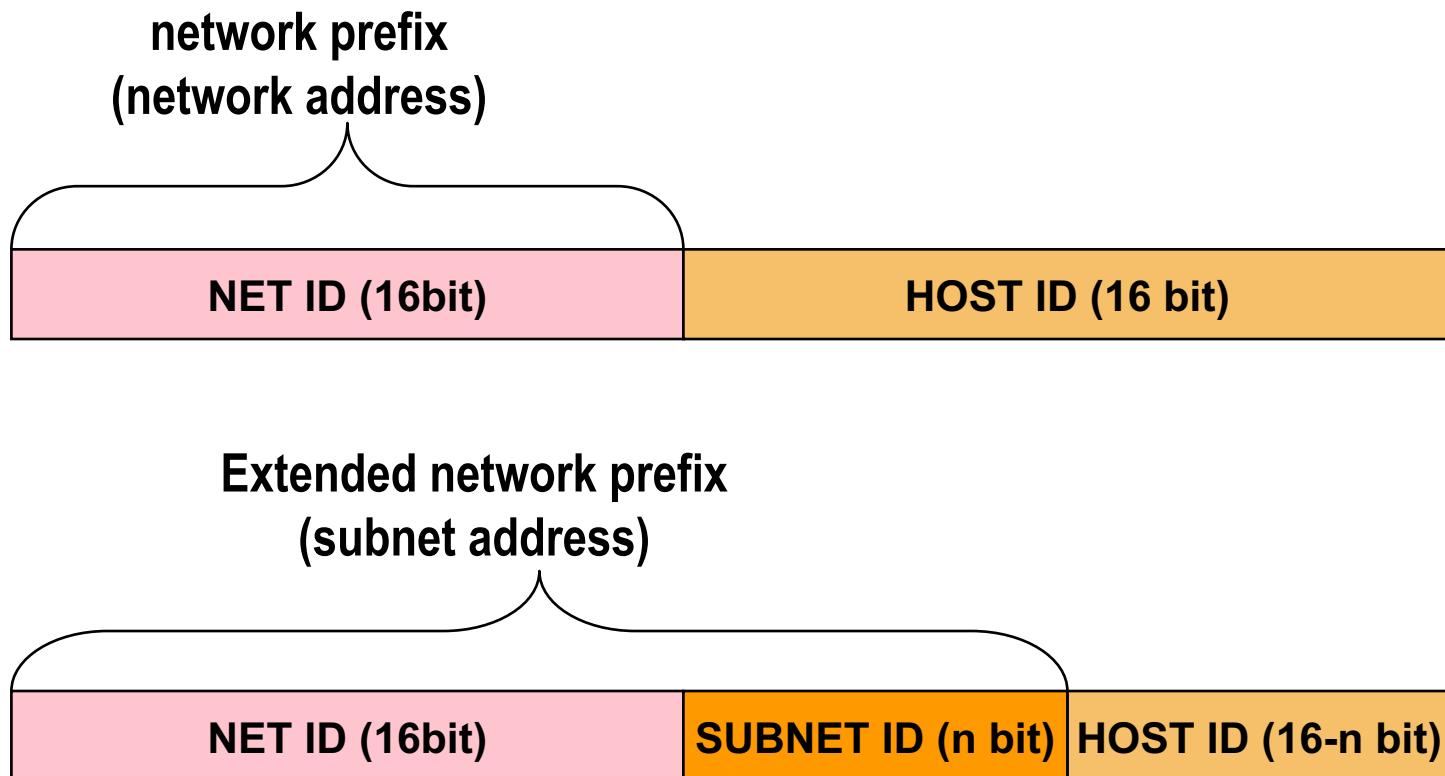


Si potrebbe usare il terzo byte per identificare la subnet: 131.175.X.0



# *Extended network prefix, subnet ID*

- Esempio con rete /16



# Analogia con le reti telefoniche

- Il concetto di suddivisione gerarchica degli indirizzi è ripreso dal mondo delle reti telefoniche.

## Mondo IP

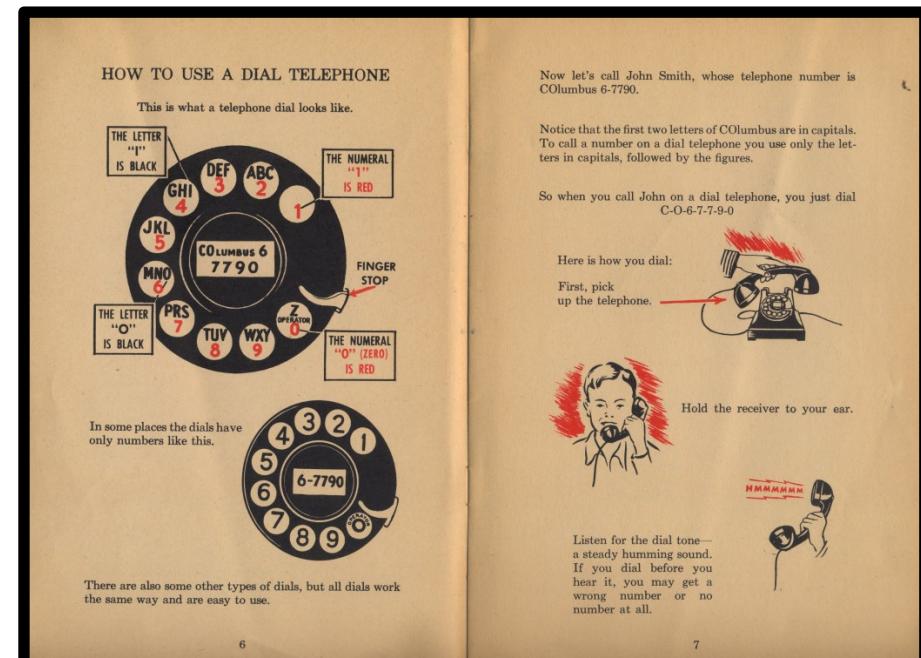
141 . 14 . 192 . 192

Site              Subnet              host

## Reti Telefoniche

(310)            824 - 2296

Area Code      Exchange      Connection



# Esempio di Subnetting-1

- Indirizzo originario della rete: 128.234.0.0/16  
(classe B, *Netmask* di 16 bit)
- Si vogliono creare almeno 1000 piccole sottoreti
- Occorrono 10 bit per il campo *subnet* in grado di indirizzare  $2^{10}=1024$  *subnet*
- La *netmask* dovrà dunque avere  $16+10=26$  bit  
(255.255.255.192)



# Esempio di Subnetting-1

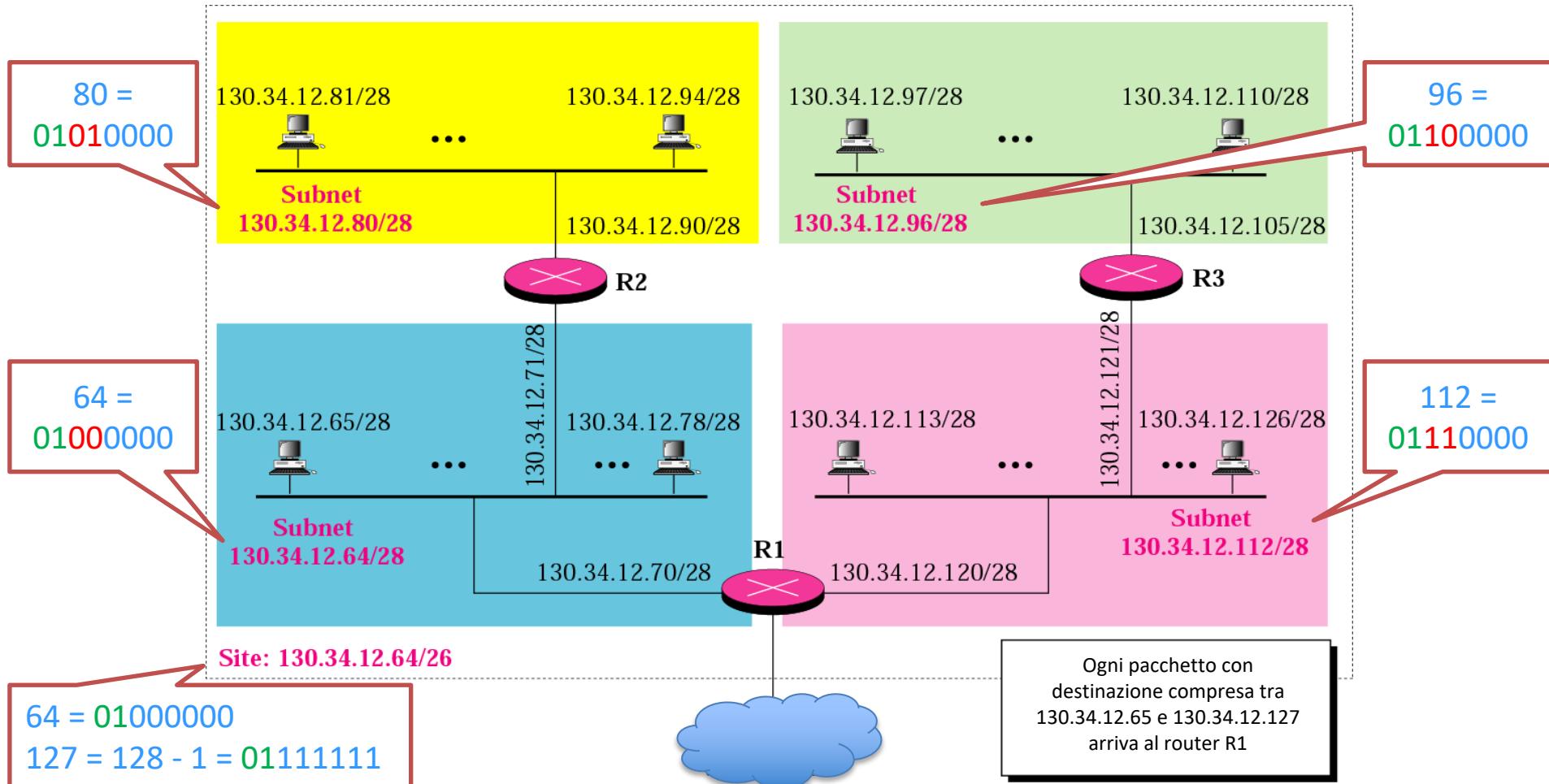
- Rimangono 6 bit per il campo host, quindi ogni sottorete supporta  $(2^6 - 2)$  host

| 255             | 192             | subnet mask        |  |
|-----------------|-----------------|--------------------|--|
| 1 1 1 1 1 1 1 1 | 1 1 0 0 0 0 0 0 | subnet mask        |  |
| 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 | 128.234.0.0/26     |  |
| 0 0 0 0 0 0 0 0 | 0 1 0 0 0 0 0 0 | 128.234.0.64/26    |  |
| 0 0 0 0 0 0 0 0 | 1 0 0 0 0 0 0 0 | 128.234.0.128/26   |  |
| 0 0 0 0 0 0 0 0 | 1 1 0 0 0 0 0 0 | 128.234.0.192/26   |  |
| ...             |                 |                    |  |
| 1 1 1 1 1 1 1 1 | 1 0 0 0 0 0 0 0 | 128.234.255.128/26 |  |
| 1 1 1 1 1 1 1 1 | 1 1 0 0 0 0 0 0 | 128.234.255.192/26 |  |



# Esempio di subnetting-2

- E' assegnato il blocco 130.34.12.64/26 servono 4 subnet



# Quando il *Subnetting* non basta..

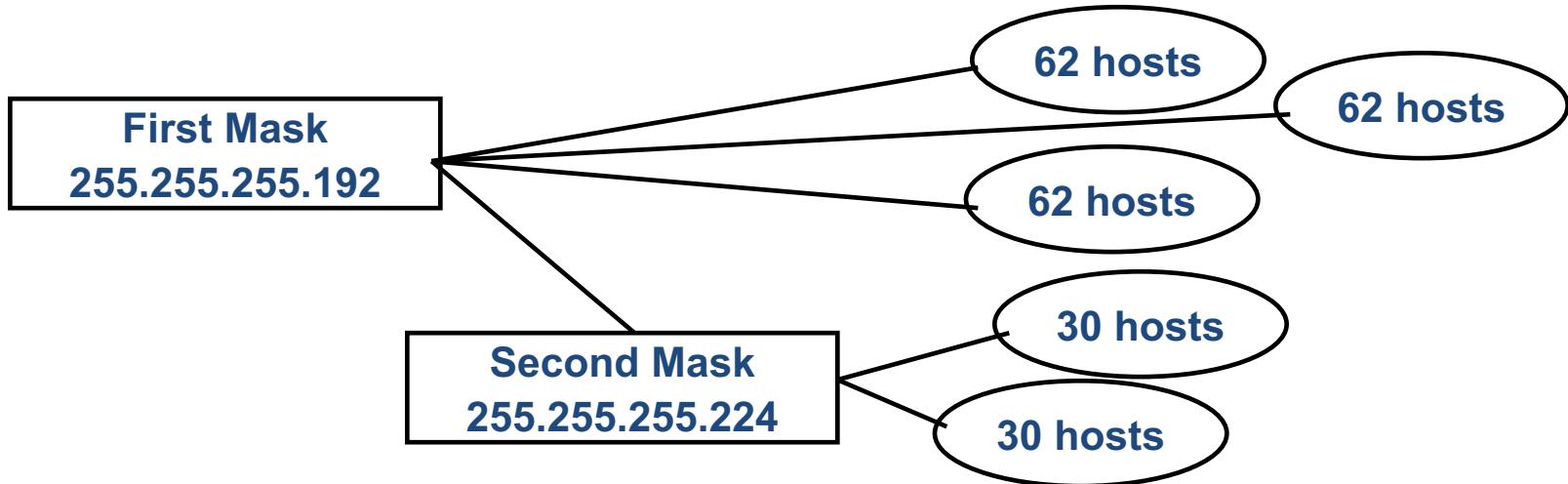
- Esempio: un'organizzazione possiede un indirizzo in classe C ed ha la necessità di creare 3 sottoreti con 60 host e altre due sottoreti con 30 host.
- Il Subnetting classico fallisce:
  - Subnet con 2 bit: 4 reti con 64 host ciascuna
  - Subnet con 3 bit: 8 reti con 32 host ciascuna
- Come fare ??



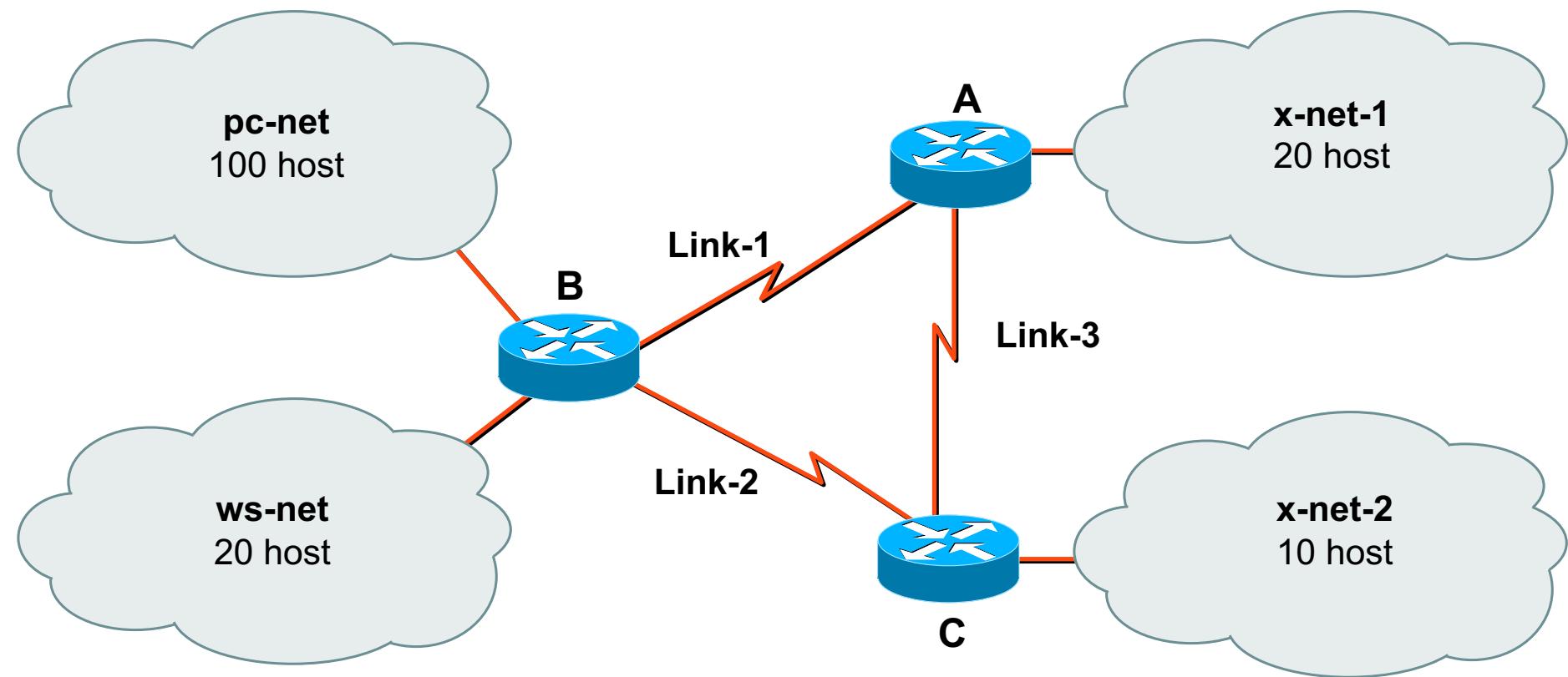
# Netmask Variabile



- *Variable Length Subnet Mask (VLSM)*
- Applicare due *netmask* in cascata
  - La prima *netmask* 255.255.255.192 con 26 “1” definisce 4 subnet con 62 indirizzi per gli *host*.
  - Si applica ad una delle *subnet* la *netmask* 255.255.255.224 che divide la sottorete in due ulteriori *subnet* con 30 indirizzi per *host*



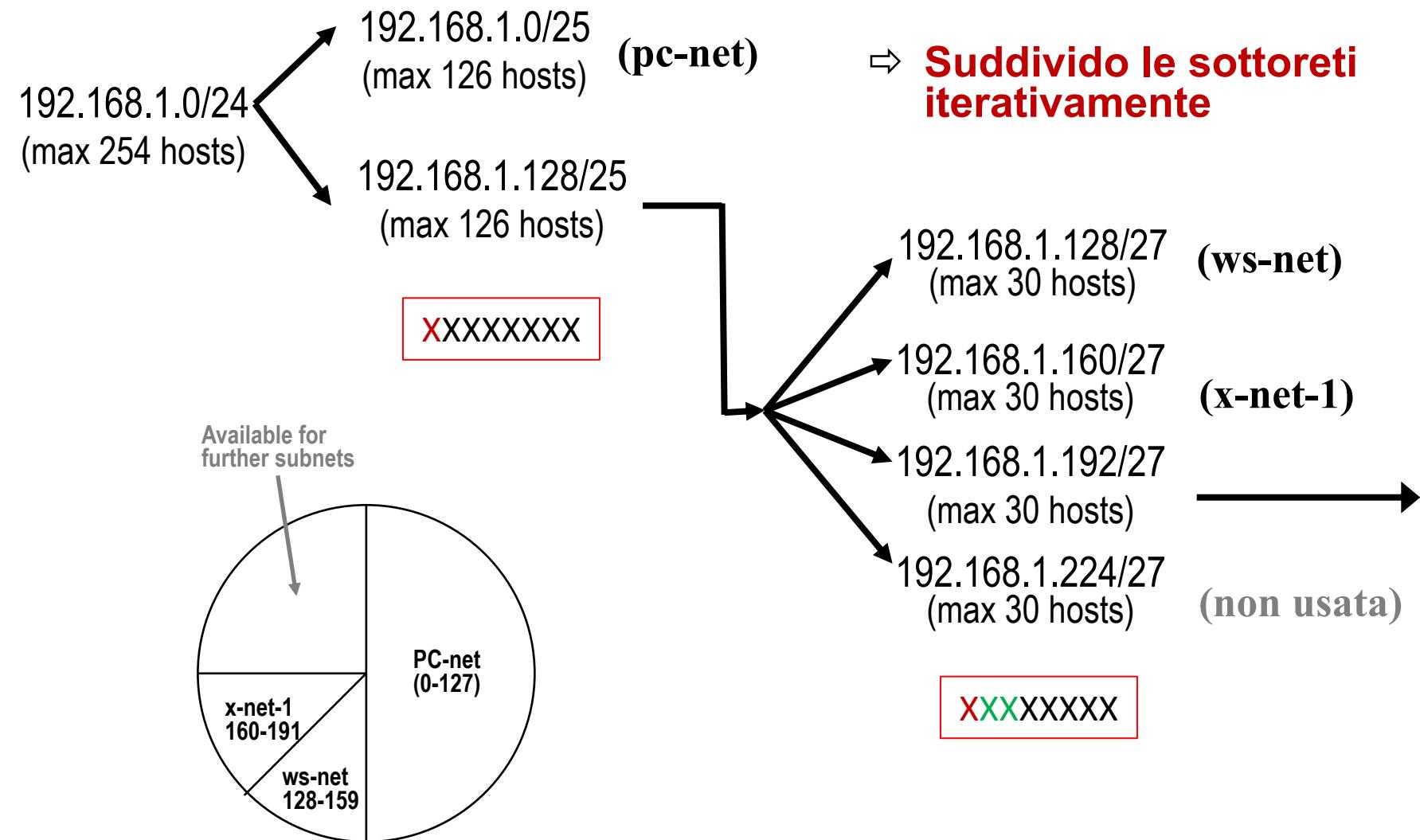
# Esempio VLSM



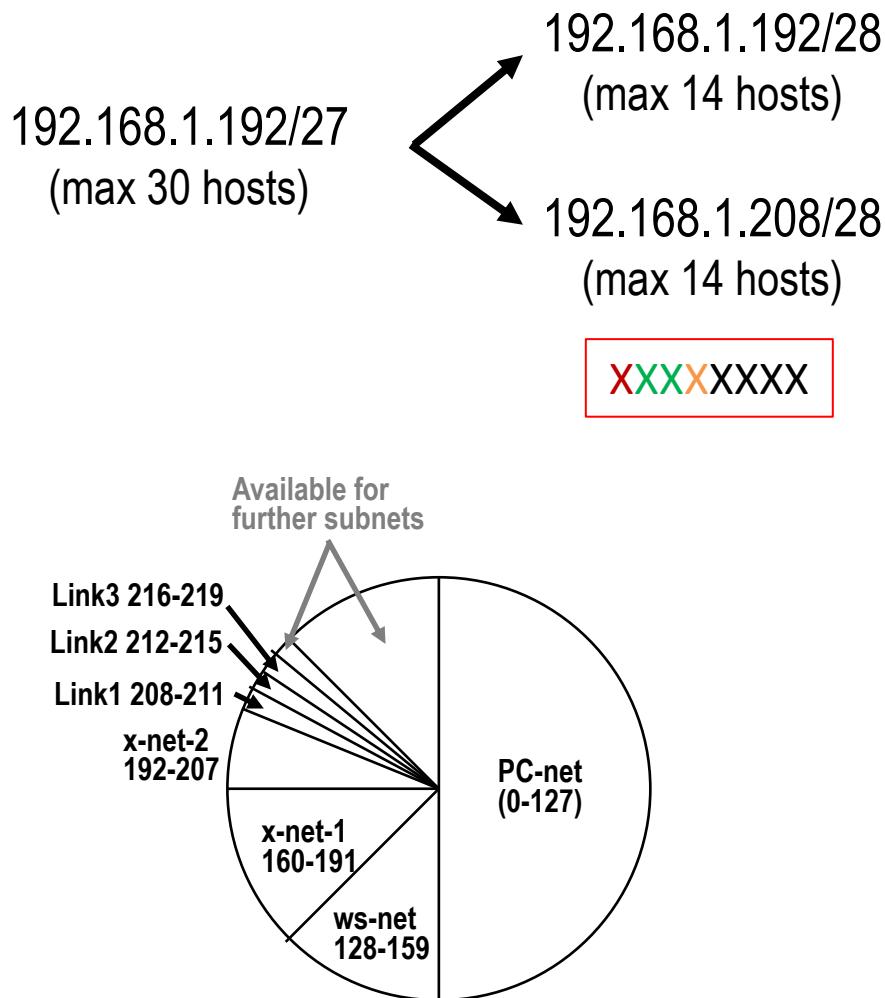
- I link punto-punto (Link-1, Link-2, Link-3) sono reti particolari
  - Hanno bisogno di due indirizzi IP, uno per interfaccia
- Necessitano comunque di un *NetID*



# Esempio VLSM

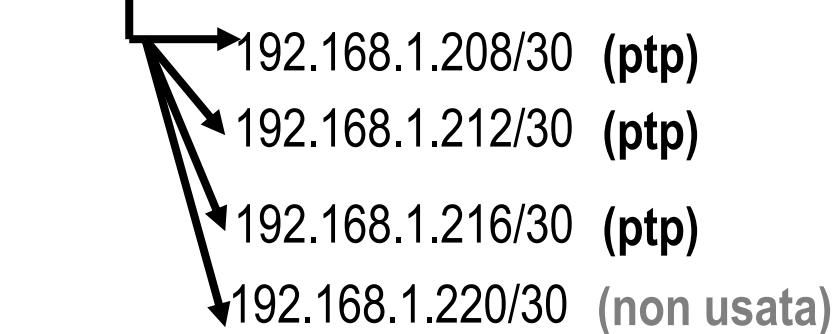


# Esempio VLSM

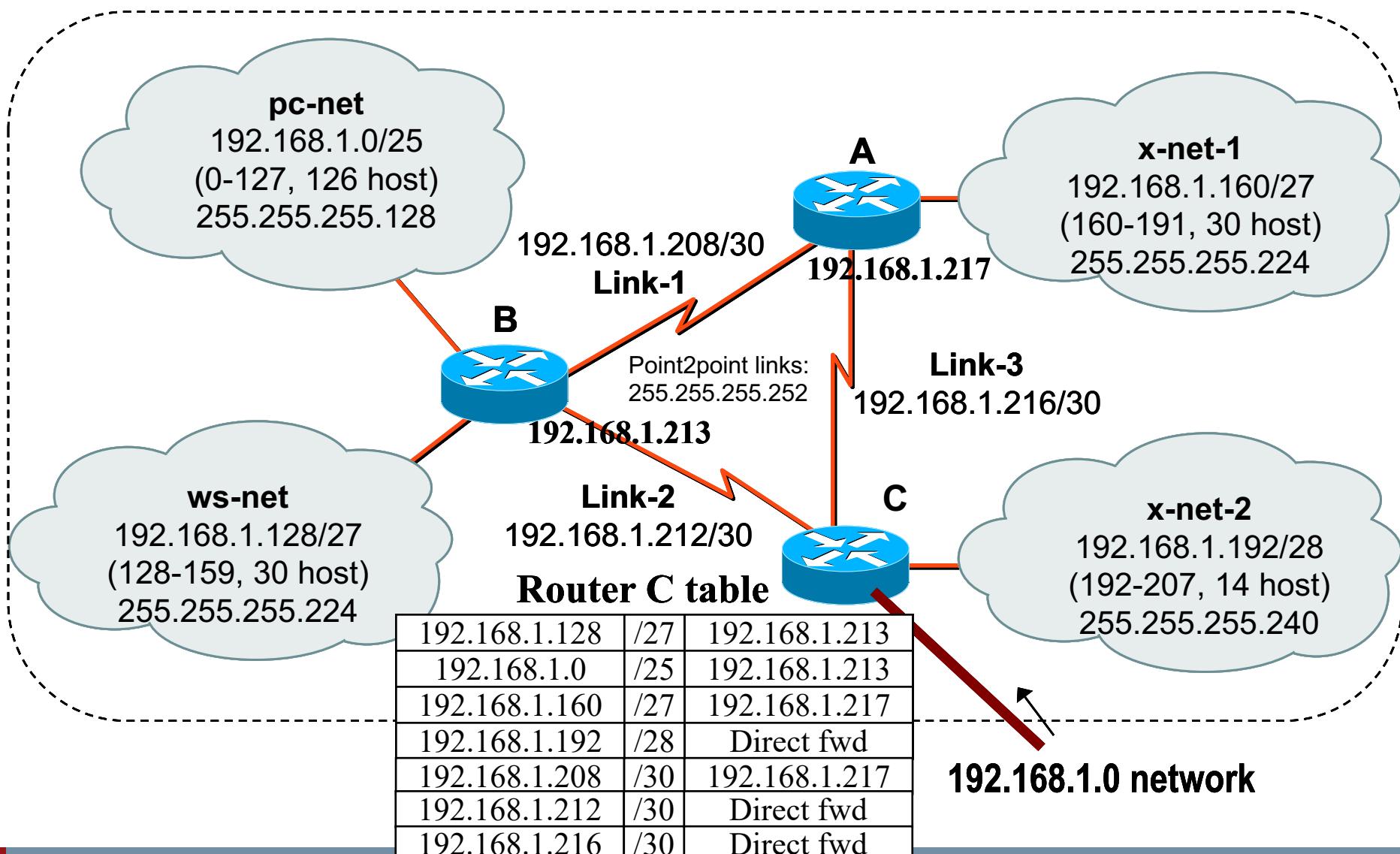


⇒ **Suddivido le sottoreti iterativamente**

(x-net-2)



# Esempio VLSM



# Agenda

- Introduzione agli indirizzi IPv4
- Indirizzamento IP
  - Indirizzamento CIDR (*Classless Inter Domain Routing*)
  - *Subnetting*
- **Inoltro dei pacchetti IP**
  - Aggregazione di indirizzi (*supernetting*)



# Inoltro dei pacchetti

- IP è una tecnica di internetworking
  - Nel trasferimento di pacchetti tra due *host* si serve della capacità di **inoltro delle reti (locali) attraversate**
- **Inoltro diretto:**
  - quando la destinazione è nella stessa rete IP (e rete locale)
- **Inoltro indiretto:**
  - quando la destinazione non è nella stessa rete IP (e rete locale)
- Trasmissione di pacchetti nelle reti locali attraversate
  - Incapsulati in trame di livello 2
  - Si basa sugli indirizzi di livello 2 (indirizzi MAC) dei dispositivi



# Inoltro diretto negli host

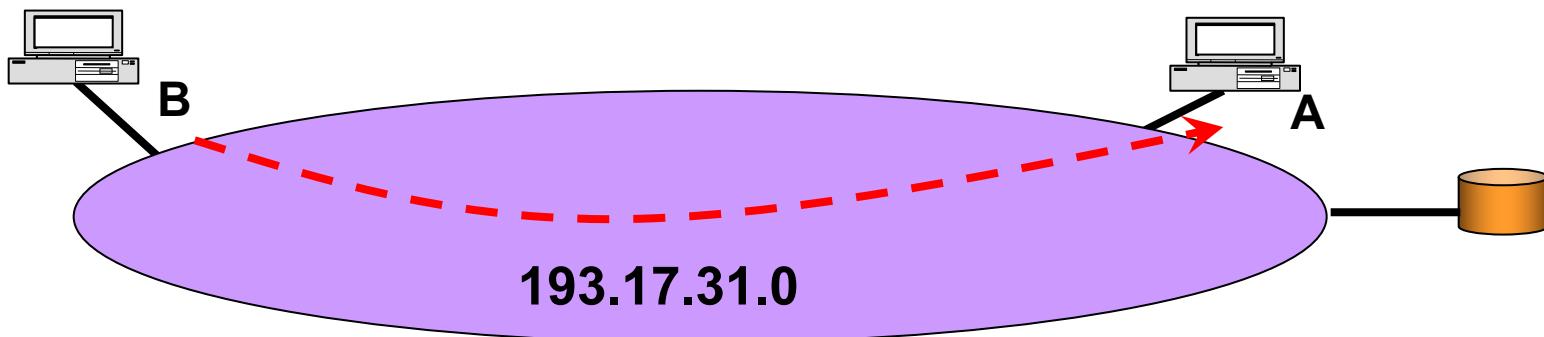
- Rete locale coincidente con rete IP

IP-B: 193.17.31.55

MAC-B: 05:98:76:6c:4a:7b

IP-A:193.17.31.45

MAC-A: 00:9f:7a:89:90:7a

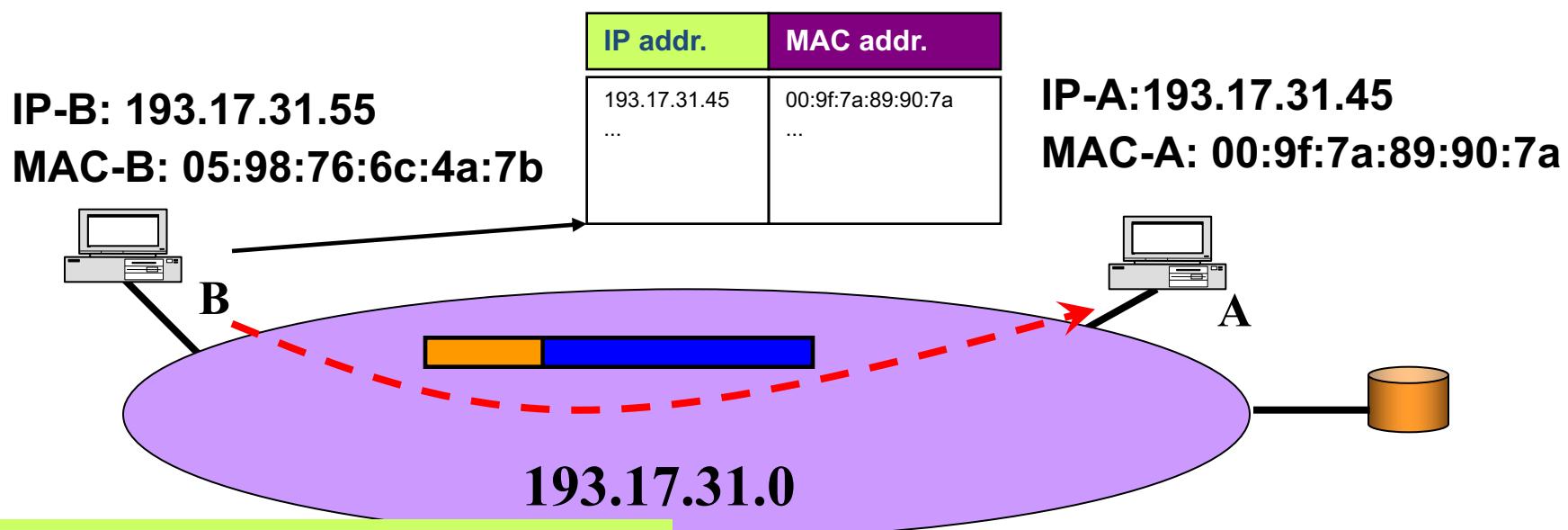


**1. L'entità IP di B deve spedire un pacchetto all'indirizzo IP-A**

**2. B conosce l'indirizzo IP-B della propria interfaccia e dal confronto con IP-A capisce che A si trova nella stessa rete**



# Inoltro diretto negli host



**3. B consulta una tabella di corrispondenza tra indirizzi IP e indirizzi della rete (indirizzi MAC nel caso di rete locale) per reperire l'indirizzo MAC-A**

**4. L'entità IP di B passa il pacchetto al livello inferiore che crea un pacchetto con destinazione MAC-A**



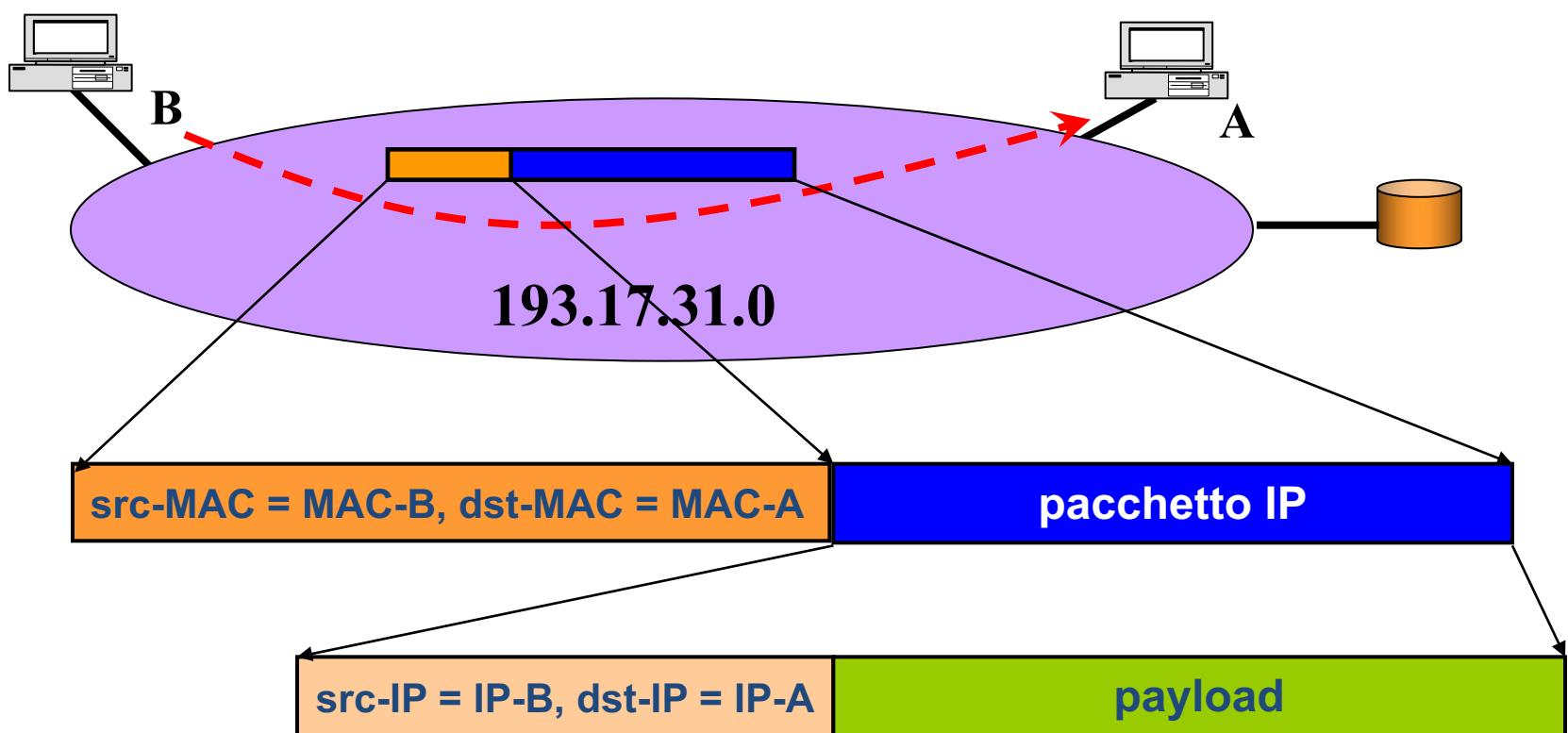
# Inoltro diretto negli host

IP-B: 193.17.31.55

MAC-B: 05:98:76:6c:4a:7b

IP-A: 193.17.31.45

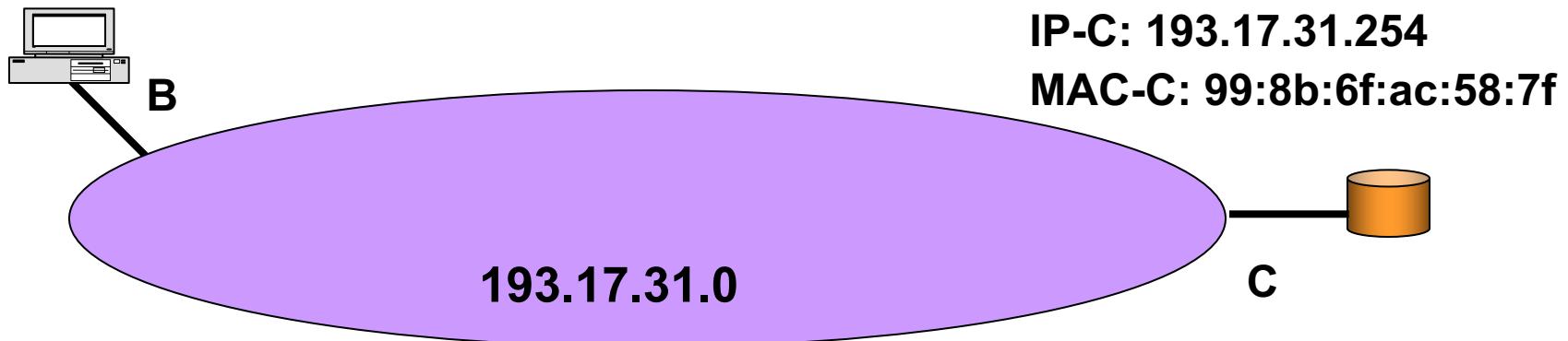
MAC-A: 00:9f:7a:89:90:7a



# Inoltro indiretto negli host

IP-B: 193.17.31.55

MAC-B: 05:98:76:6c:4a:7b

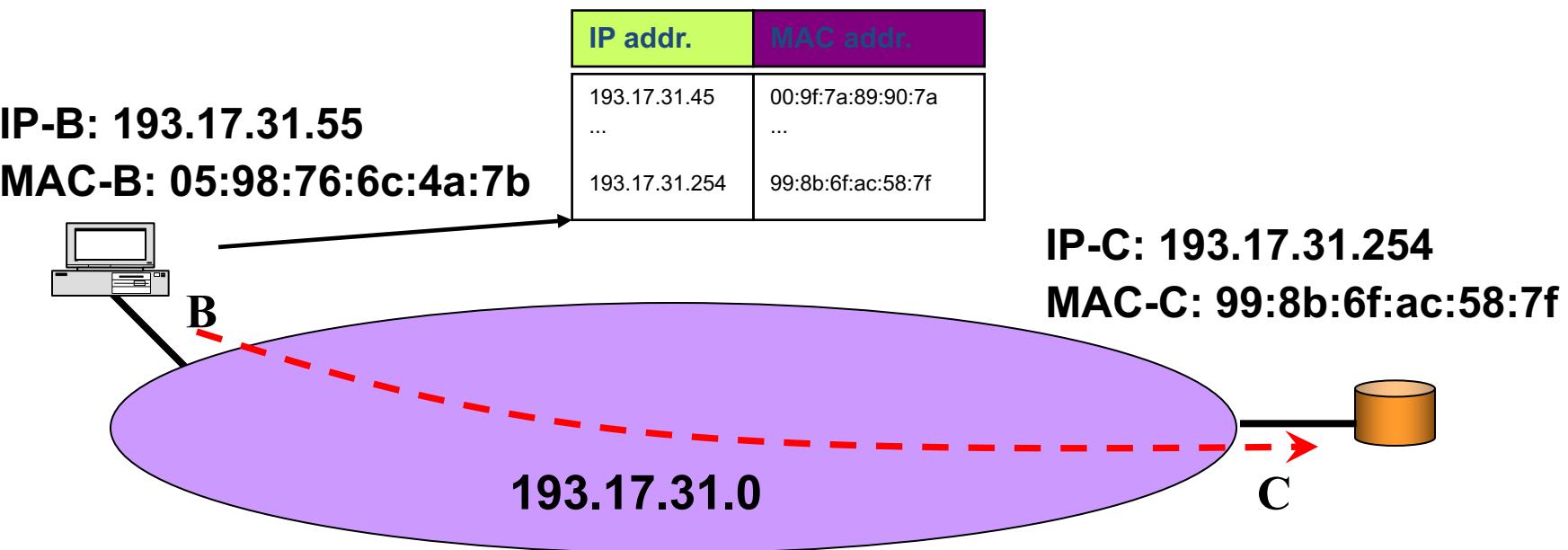


**1.** L'entità IP di B deve spedire un pacchetto all'indirizzo *IP-D=131.17.23.4*

**2.** B conosce l'indirizzo IP-B della propria interfaccia e dal confronto con IP-D capisce che D NON si trova nella stessa rete



# Inoltro indiretto negli host



**3. B deve dunque inoltrare il pacchetto ad un router (di solito è configurato un solo default router)**

**4. B recupera l'indirizzo MAC del router nella tabella di corrispondenza e passa il pacchetto al livello inferiore**



# Inoltro indiretto negli host

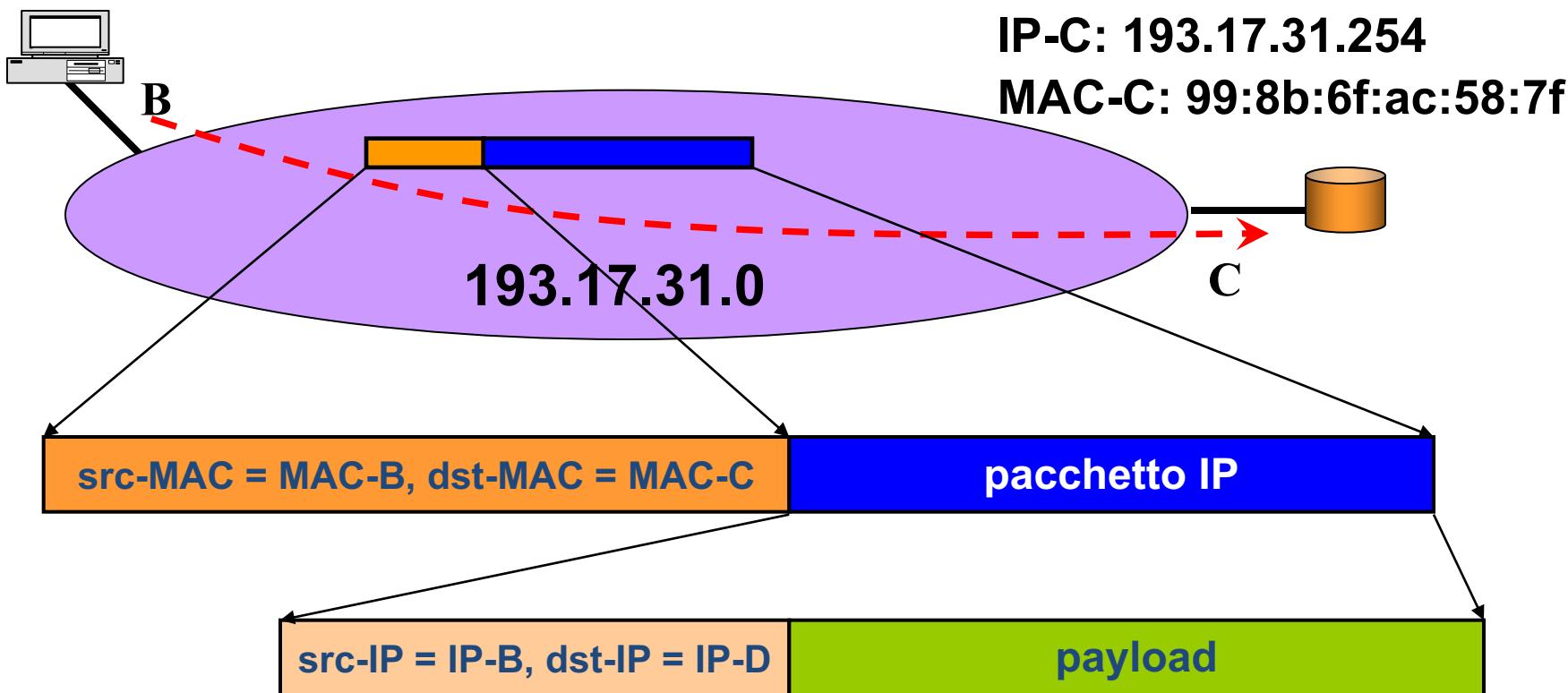
5. il pacchetto viene costruito e spedito sull'interfaccia

IP-B: 193.17.31.55

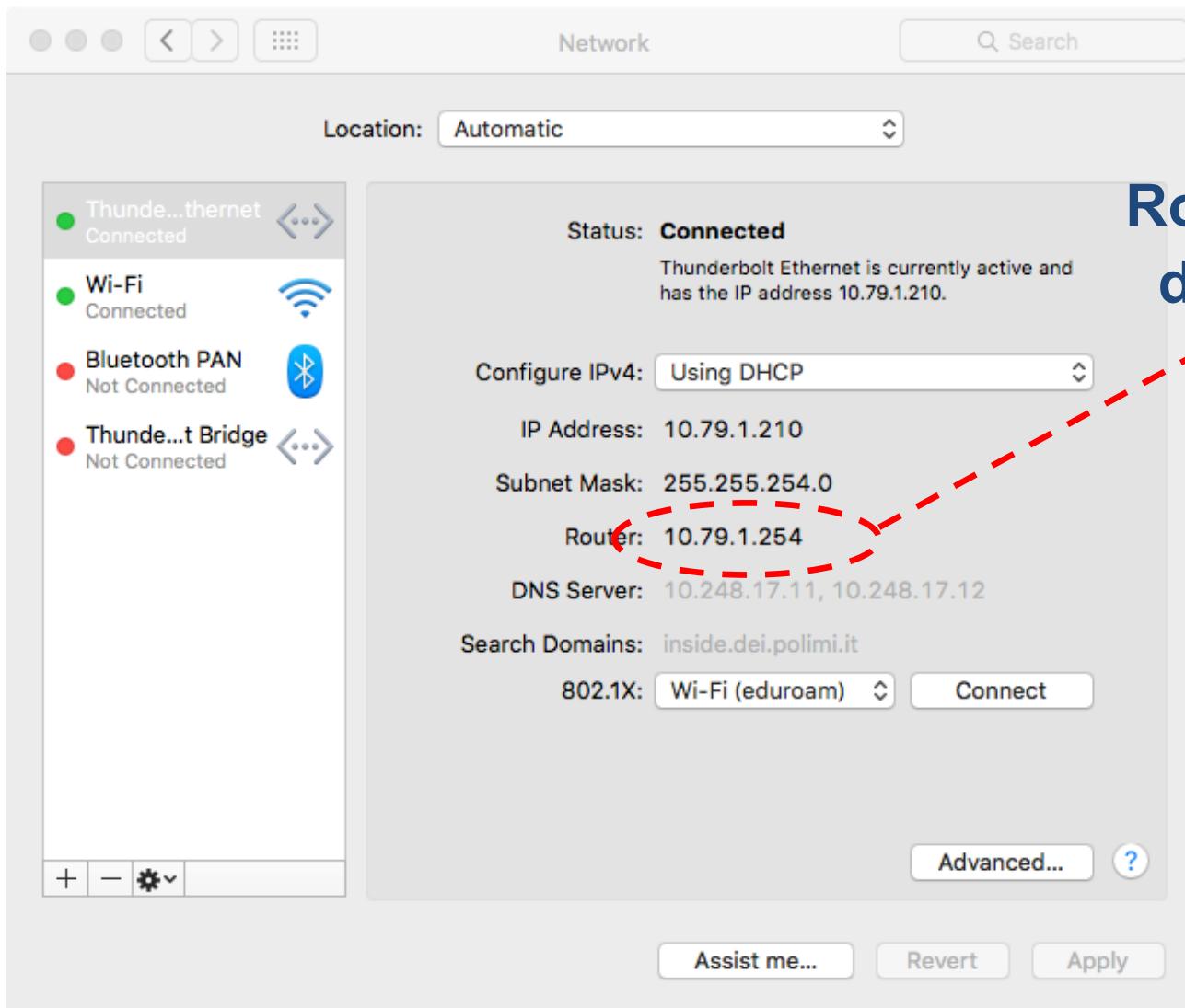
MAC-B: 05:98:76:6c:4a:7b

IP-C: 193.17.31.254

MAC-C: 99:8b:6f:ac:58:7f



# Configurazione degli host



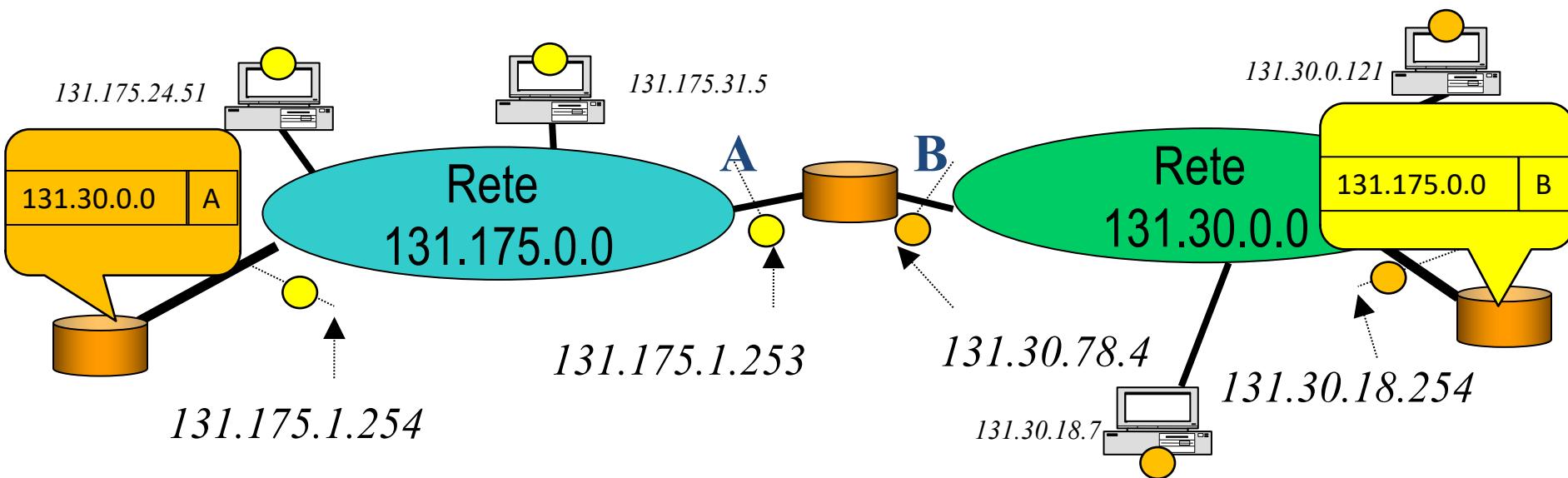
# Inoltro nei router

- Router: dispositivi di *internetworking* con interfacce di uscita multiple.
- Anche i router seguono le tecniche di inoltro diretto ed indiretto ma:
  - Inoltro diretto: hanno di solito più di una interfaccia dove poter effettuare l'inoltro diretto
  - Inoltro indiretto: si basa su *tabelle di routing* dove è definita la “rotta” di instradamento
- Caratteristiche dell'inoltro:
  - **DESTINATION BASED**: L'inoltro IP è basato sul solo indirizzo di destinazione
  - **NEXT HOP ROUTING**: Nelle tabelle di routing per ogni rete di destinazione è indicato solo il prossimo router (next-hop) nel percorso verso la destinazione



# Inoltro nei router

- L'inoltro avviene da router a router attraverso le reti IP
- I router inoltrano i pacchetti basandosi esclusivamente sulla parte di *NetID* dell'indirizzo di destinazione
- Tutti gli host che appartengono alla rete di destinazione sono identificati nelle tabelle di *routing* da una singola *entry* (il prefisso di rete) → **Address Aggregation**



# Requisiti di CIDR

- Nelle tabelle di *routing* deve sempre essere indicata per ogni *entry* in modo esplicito la lunghezza del NetID
  - La divisione tra NetID e HostID non potrebbe essere determinata altrimenti
- I *protocolli di routing* devono supportare l'invio dell'informazione di *Netmask* insieme agli indirizzi di rete in tutti i messaggi di annuncio di rotte (*route advertisement*)
  - In passato non tutti i protocolli di *routing* lo consentivano (es. RIP v1)

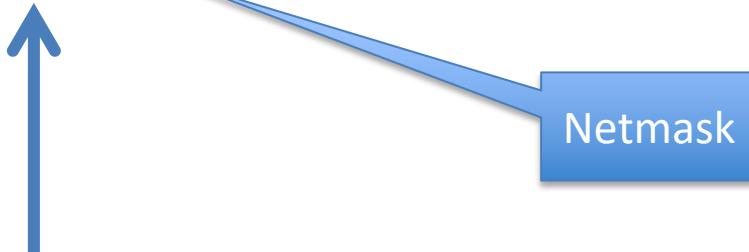
|     |      |       |
|-----|------|-------|
| ... | ...  | ...   |
| net | mask | route |
| ... | ...  | ...   |



# Tabella di routing (routing table)



| Destination prefix | Next-hop       | Output interface |
|--------------------|----------------|------------------|
| 24.40.32/20        | 192.41.177.148 | 2                |
| 130.86/16          | 191.41.177.181 | 6                |
| 208.12.16/20       | 192.41.177.241 | 4                |
| 208.12.21/24       | 192.41.177.196 | 1                |
| 167.24.103/24      | 192.41.177.148 | 2                |



Tutti gli indirizzi nell'intervallo: 167.24.103.0 –  
167.24.103.255



# Inoltro diretto e indiretto con le netmask

- Per inoltrare un pacchetto occorre capire se appartiene alla sottorete di una delle interfacce
- Per effettuare la verifica si fa un AND bit a bit tra indirizzo dell'interfaccia e *netmask* e tra indirizzo di destinazione e *netmask*
- Se i due risultati coincidono allora la sottorete è la stessa e si procede all'**inoltro diretto**

**destinazione: (131.175.21.77) AND (255.255.255.0) = 131.175.21.0**

Nota: gli indirizzi a bordo dei PACCHETTI non hanno MAI NETMASK.

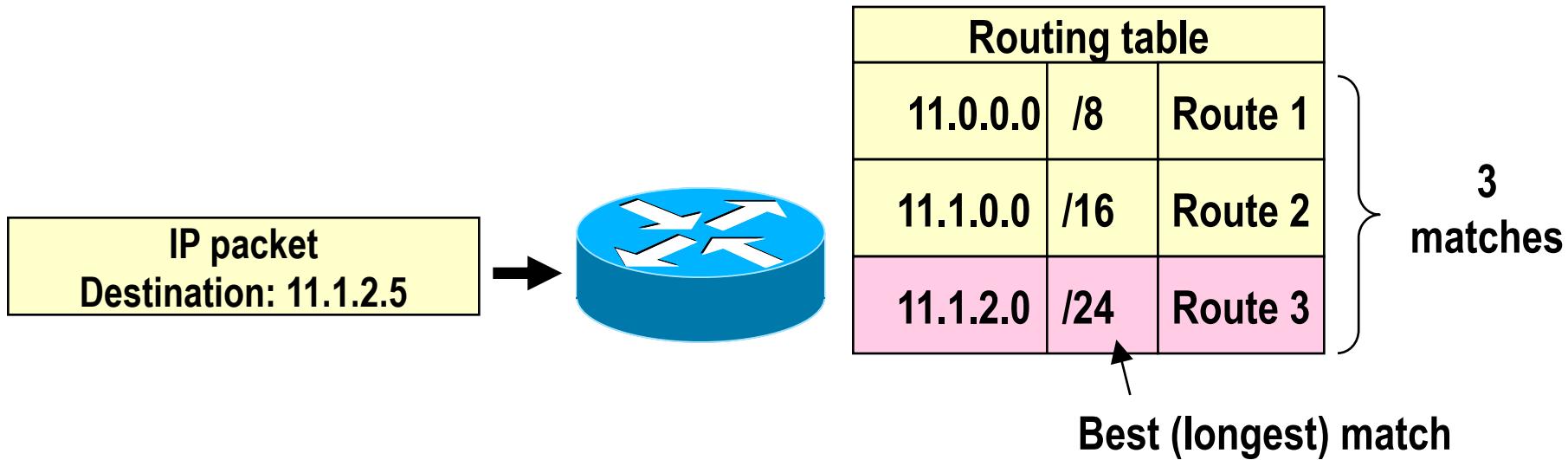
**confronto  
positivo**

**interfaccia: (131.175.21.96) AND (255.255.255.0) = 131.175.21.0**



# Tabelle di routing con le netmask

- Se i confronti con le interfacce sono negativi occorre procedere ad un inoltro indiretto
- Occorre analizzare la tabella di *routing* del router
- Il confronto riga per riga si effettua allo stesso modo (AND bit a bit) usando la *netmask* relativa a ciascuna riga
- Se il confronto dà esito positivo per più righe della tabella viene selezionata la tabella con la *netmask* che ha il maggior numero di 1 (si dice comunemente che vale il principio del prefisso più lungo o *longest prefix matching*).
  - Prefisso più lungo equivale a rotta più “specificata”



Longest match = smallest network, informazioni di routing più precise e dettagliate



# Tabelle di routing con le netmask

| network      | netmask       | first hop      |
|--------------|---------------|----------------|
| 131.175.21.0 | 255.255.255.0 | 131.17.123.254 |
| 131.175.16.0 | 255.255.255.0 | 131.17.78.254  |
| 131.56.0.0   | 255.255.0.0   | 131.17.15.254  |
| 131.155.0.0  | 255.255.0.0   | 131.17.15.254  |
| 0.0.0.0      | 0.0.0.0       | 131.17.123.254 |

interface eth0

|            |               |
|------------|---------------|
| IP address | 131.17.123.1  |
| netmask    | 255.255.255.0 |

interface eth1

|            |               |
|------------|---------------|
| IP address | 131.17.78.1   |
| netmask    | 255.255.255.0 |

interface eth2

|            |               |
|------------|---------------|
| IP address | 131.17.15.12  |
| netmask    | 255.255.255.0 |

default router:

il confronto dà sempre esito positivo ma la netmask è lunga 0 bit



# Tabelle di routing: esempio (1)

| network      | netmask       | first hop      |
|--------------|---------------|----------------|
| 131.175.15.0 | 255.255.255.0 | 131.175.21.1   |
| 131.175.16.0 | 255.255.255.0 | 131.175.21.2   |
| 131.175.17.0 | 255.255.255.0 | 131.175.21.3   |
| 131.180.23.0 | 255.255.255.0 | 131.175.21.4   |
| 131.180.18.0 | 255.255.255.0 | 131.175.21.4   |
| 131.180.21.0 | 255.255.255.0 | 131.175.21.4   |
| 131.180.0.0  | 255.255.0.0   | 131.175.21.5   |
| 0.0.0.0      | 0.0.0.0       | 131.175.12.254 |

131.175.21.86

interfaccia 1: 131.175.21.254, 255.255.255.0  
interfaccia 2: 131.175.12.253, 255.255.255.0



# Tabelle di routing: esempio (2)

| network      | netmask       | first hop      |
|--------------|---------------|----------------|
| 131.175.15.0 | 255.255.255.0 | 131.175.21.1   |
| 131.175.16.0 | 255.255.255.0 | 131.175.21.2   |
| 131.175.17.0 | 255.255.255.0 | 131.175.21.3   |
| 131.180.23.0 | 255.255.255.0 | 131.175.21.4   |
| 131.180.18.0 | 255.255.255.0 | 131.175.21.4   |
| 131.180.21.0 | 255.255.255.0 | 131.175.21.4   |
| 131.180.0.0  | 255.255.0.0   | 131.175.21.5   |
| 0.0.0.0      | 0.0.0.0       | 131.175.12.254 |

131.175.16.65

X

OK



X

X

X

X

X

OK

interfaccia 1: 131.175.21.254, 255.255.255.0 X  
interfaccia 2: 131.175.12.253, 255.255.255.0 X



# Tabelle di routing: esempio (3)

| network      | netmask       | first hop      |
|--------------|---------------|----------------|
| 131.175.15.0 | 255.255.255.0 | 131.175.21.1   |
| 131.175.16.0 | 255.255.255.0 | 131.175.21.2   |
| 131.175.17.0 | 255.255.255.0 | 131.175.21.3   |
| 131.180.23.0 | 255.255.255.0 | 131.175.21.4   |
| 131.180.18.0 | 255.255.255.0 | 131.175.21.4   |
| 131.180.21.0 | 255.255.255.0 | 131.175.21.4   |
| 131.180.0.0  | 255.255.0.0   | 131.175.21.5   |
| 0.0.0.0      | 0.0.0.0       | 131.175.12.254 |

131.180.21.78

X

X

X

X

X

OK

OK

OK

interfaccia 1: 131.175.21.254, 255.255.255.0 X  
interfaccia 2: 131.175.12.253, 255.255.255.0 X



# Tabelle di routing: esempio (4)

| network      | netmask       | first hop      |    |
|--------------|---------------|----------------|----|
| 131.175.15.0 | 255.255.255.0 | 131.175.21.1   | X  |
| 131.175.16.0 | 255.255.255.0 | 131.175.21.2   | X  |
| 131.175.17.0 | 255.255.255.0 | 131.175.21.3   | X  |
| 131.180.23.0 | 255.255.255.0 | 131.175.21.4   | X  |
| 131.180.18.0 | 255.255.255.0 | 131.175.21.4   | X  |
| 131.180.21.0 | 255.255.255.0 | 131.175.21.4   | X  |
| 131.180.0.0  | 255.255.0.0   | 131.175.21.5   | X  |
| 0.0.0.0      | 0.0.0.0       | 131.175.12.254 | OK |

**interfaccia 1: 131.175.21.254, 255.255.255.0 X**  
**interfaccia 2: 131.175.12.253, 255.255.255.0 X**

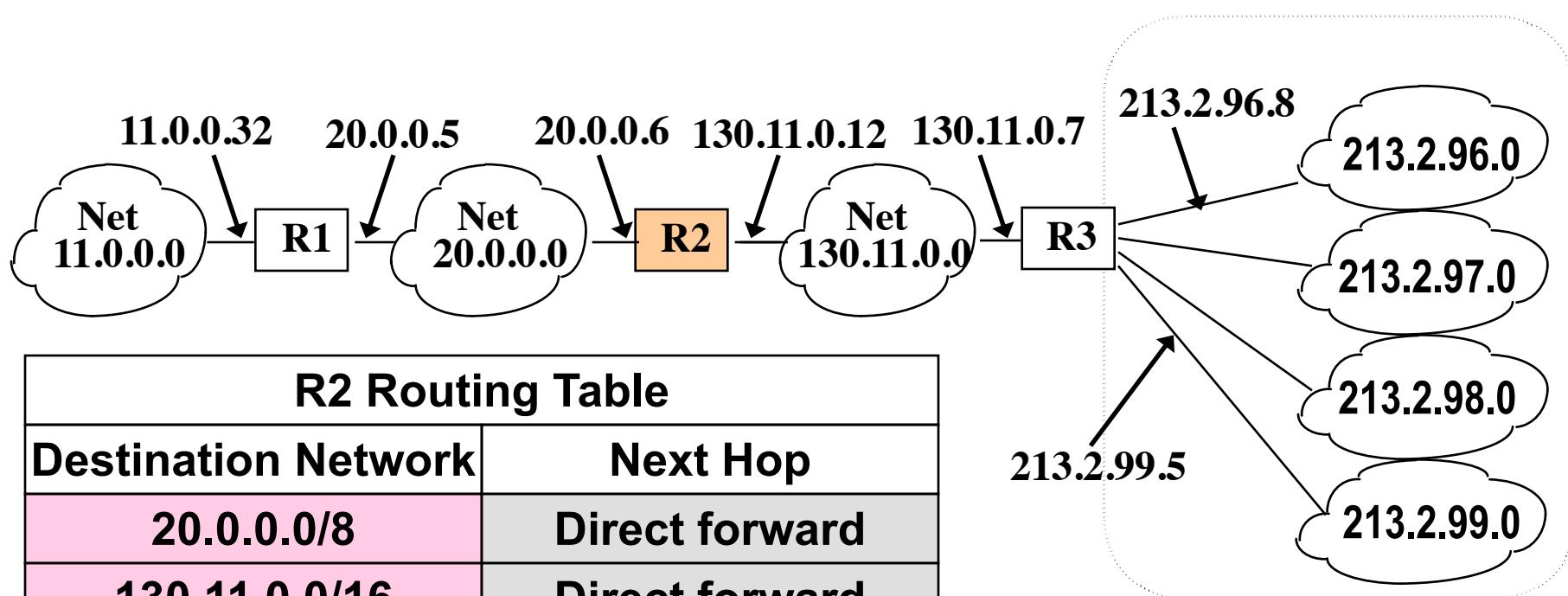


# Agenda

- Introduzione agli indirizzi IPv4
- Indirizzamento IP
  - Indirizzamento CIDR (*Classless Inter Domain Routing*)
  - *Subnetting*
- Inoltro dei pacchetti IP
  - Aggregazione di indirizzi (*supernetting*)



# Crescita tabelle di routing



213.2.99.5

213.2.99.5

- Limiti HW e SW sul tempo di ricerca nelle routing table
- Gli aggiornamenti della routing table diventano critici (tante info trasferite tra i router)
- Il default routing non è una soluzione ottima



# Supernet mask

- 4 reti con indirizzi contigui
  - 213.2.96.0    11010101.00000010.01100000.00000000
  - 213.2.97.0    11010101.00000010.01100001.00000000
  - 213.2.98.0    11010101.00000010.01100010.00000000
  - 213.2.99.0    11010101.00000010.01100011.00000000
- Definisco una Supernet con “2 bit in meno”
  - Da 255.255.255.0 a 255.255.252.0

11111111.11111111.11111111.00000000



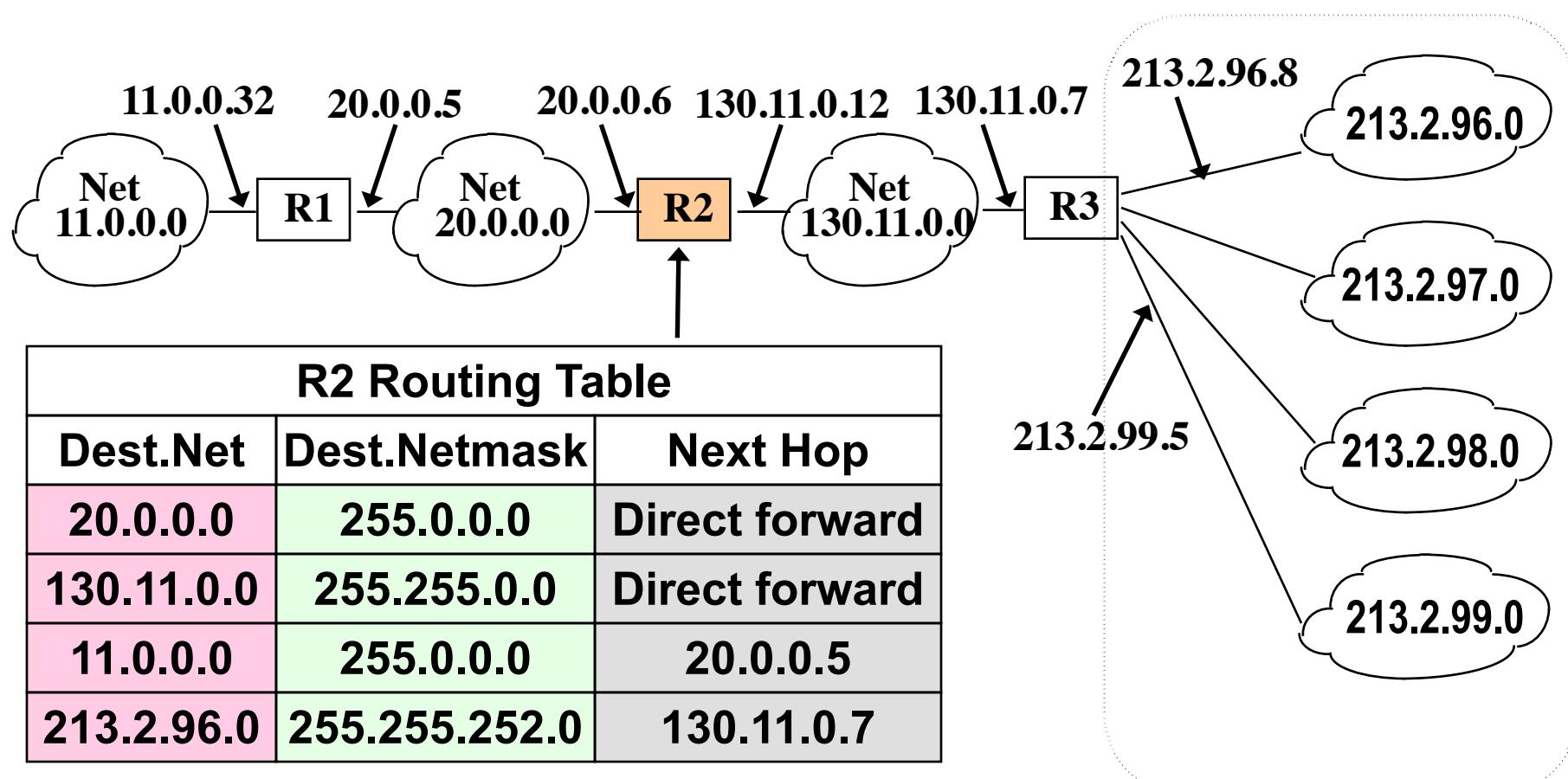
11111111.11111111.11111100.00000000



- Supernet address: 213.2.96.0/22
  - 11010101 . 00000010 . 011000 00 . 00000000



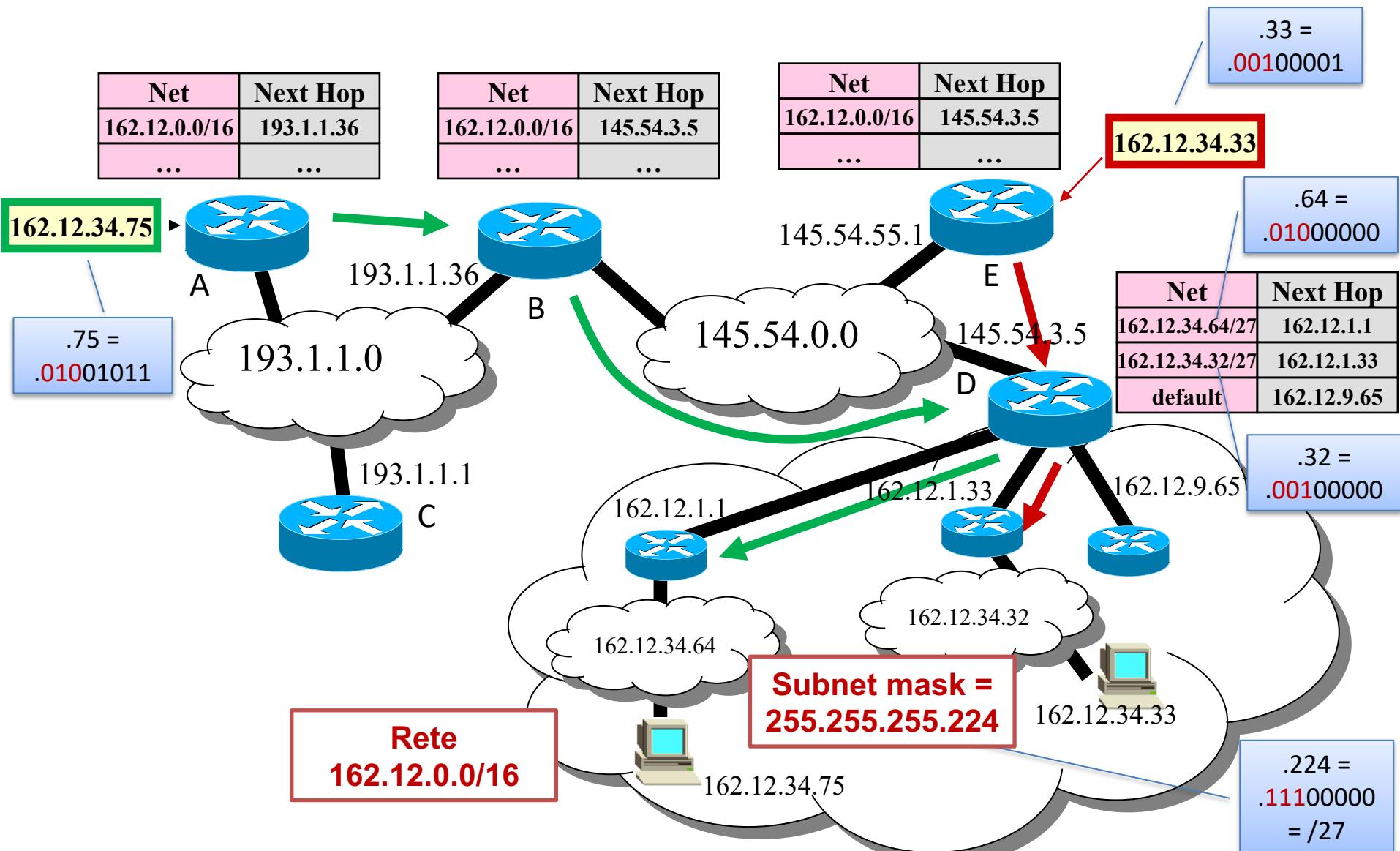
# Supernetting (aggregazione di rotte)



- Per consentire l'aggregazione delle rotte in modo significativo è opportuno che i blocchi di indirizzi vengano assegnati su base geografica (blocchi adiacenti nella stessa area)

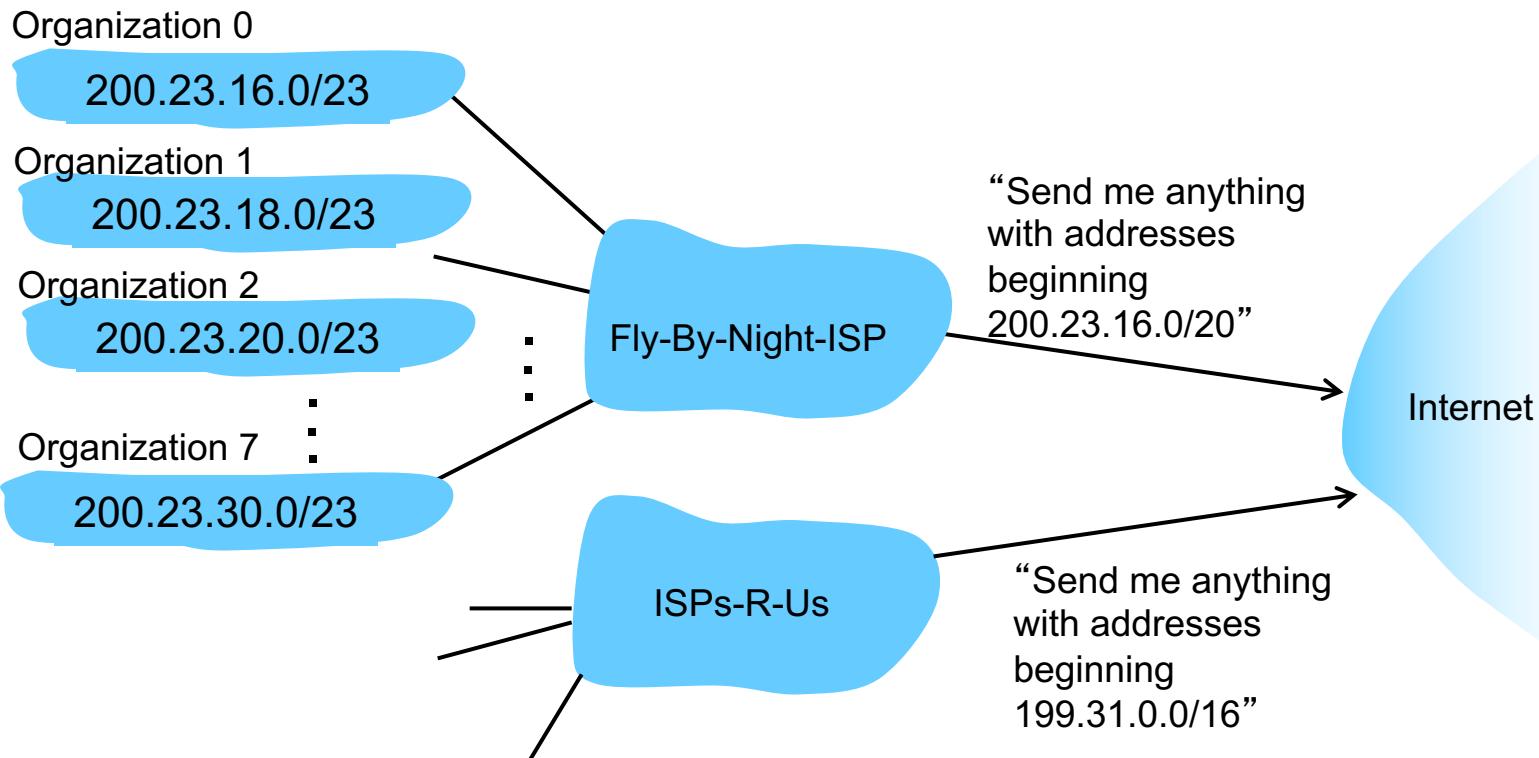


# I core router ignorano il subnetting



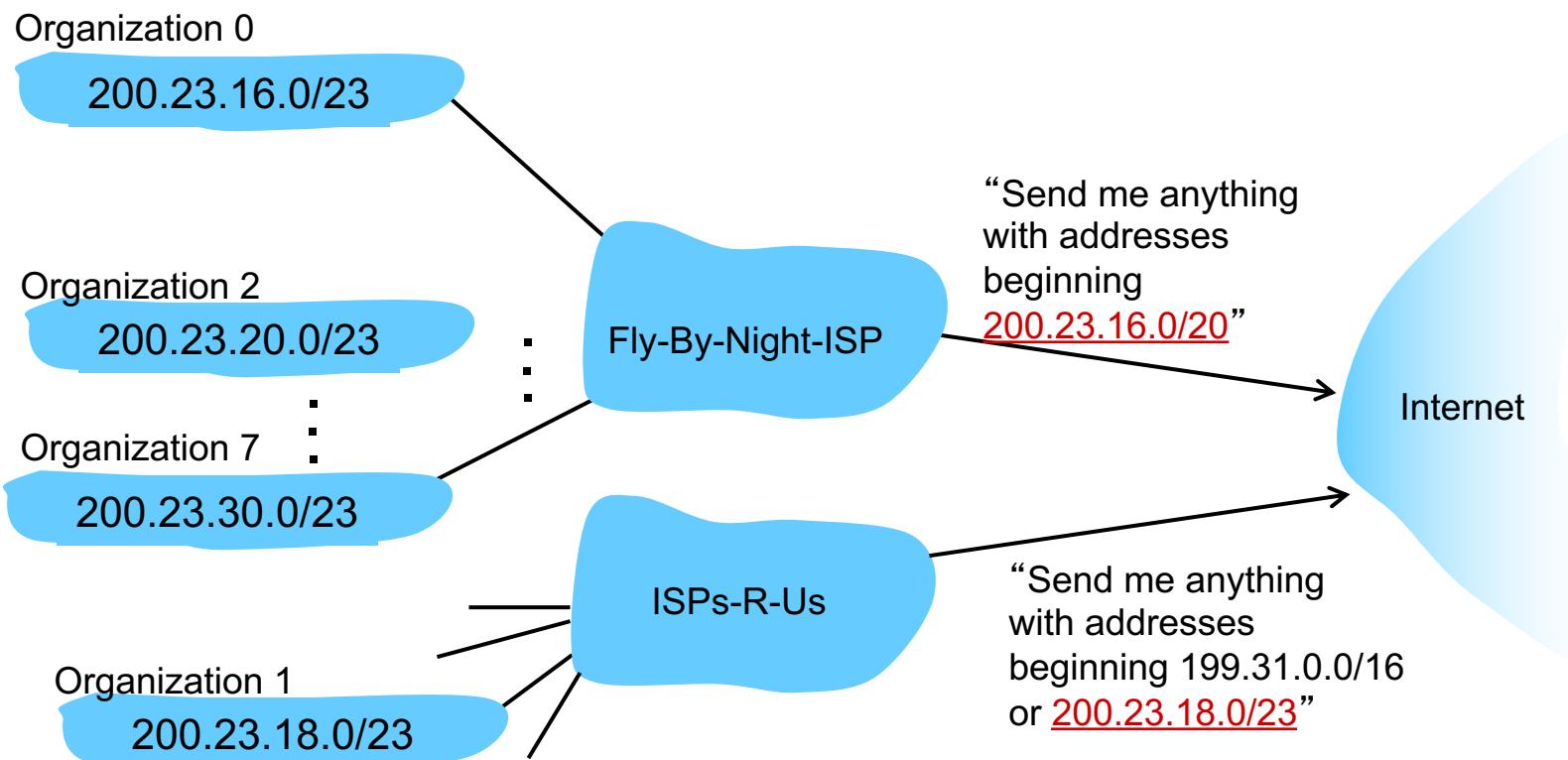
# Indirizzamento gerarchico: aggregazione di rotte

- L'aggregazione degli indirizzi consente un'efficiente propagazione delle rotte



# Indirizzamento gerarchico: exception route

- ISPs-R-Us ha una rotta più specifica (eccezione) verso l'Organizzazione 1
- La rete funziona ugualmente grazie al longest prefix matching



# Formato della tabella di routing

| Netmask   | Destination | Next Hop (Gateway) | Flag  | Metric | Use   | Interface |
|-----------|-------------|--------------------|-------|--------|-------|-----------|
| 255.0.0.0 | 124.0.0.0   | 145.6.7.23         | UG    | 4      | 20    | Eth1      |
| .....     | .....       | .....              | ..... | .....  | ..... | .....     |
| .....     | .....       | .....              | ..... | .....  | ..... | .....     |
| .....     | .....       | .....              | ....  | .....  | ...   | .....     |

- Flag:
  - U: route attiva
  - G: destinatario fuori dalla sottorete (default gateway)
  - H: destinatario è un host specifico (non una rete)
  - D: rota dinamica creata da protocollo di routing o reindirizzamento ICMP
  - M: rota modificata da protocollo di routing o reindirizzamento ICMP
- Reference Count: numero connessioni attive per quella rota
- Use: numero pacchetti verso il destinatario (numero di lookup)
- Interface: nome interfaccia di uscita

