Computer Networks

Internet Protocol

Luigi Vetrano

Corso Reti A.A. 2013-2014

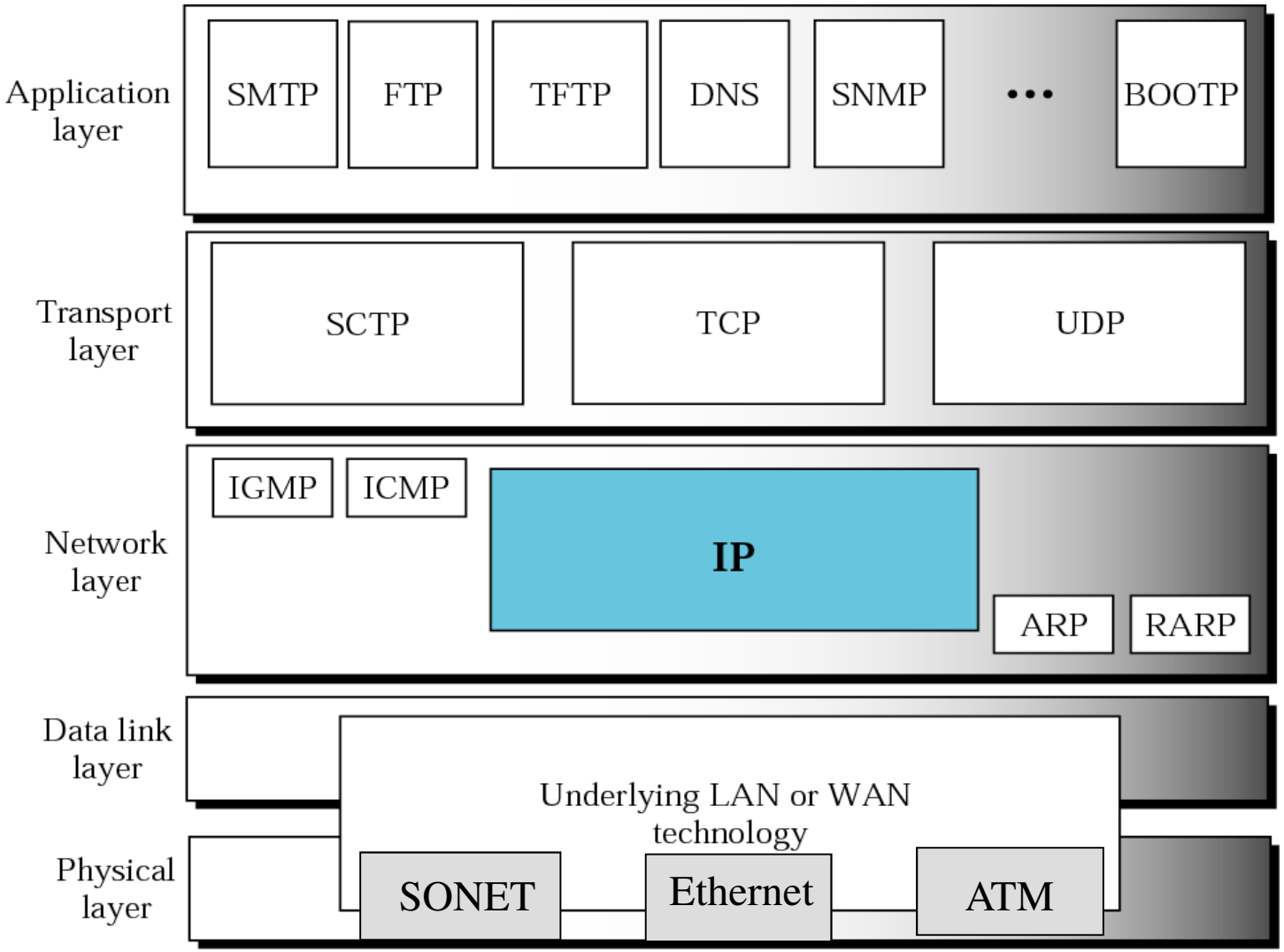
TechnoLabs

Livello di rete

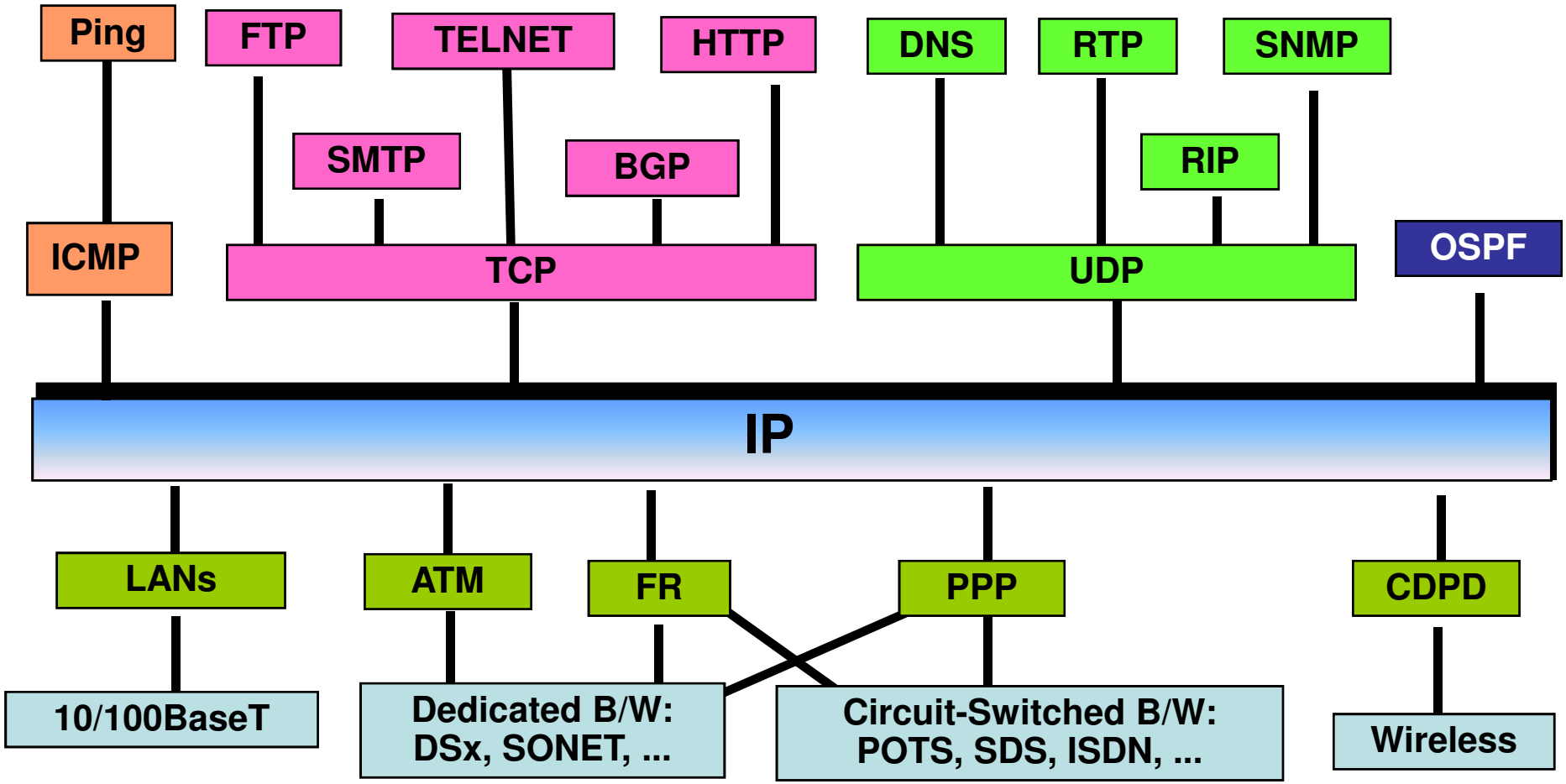
Obiettivi:

- **Capire i principi che stanno dietro i servizi del livello di rete:**
  - Indirizzamento
  - Instradamento (scelta del percorso)
  - Scalabilità
  - Funzioni di un router
  - Implementazione in Internet

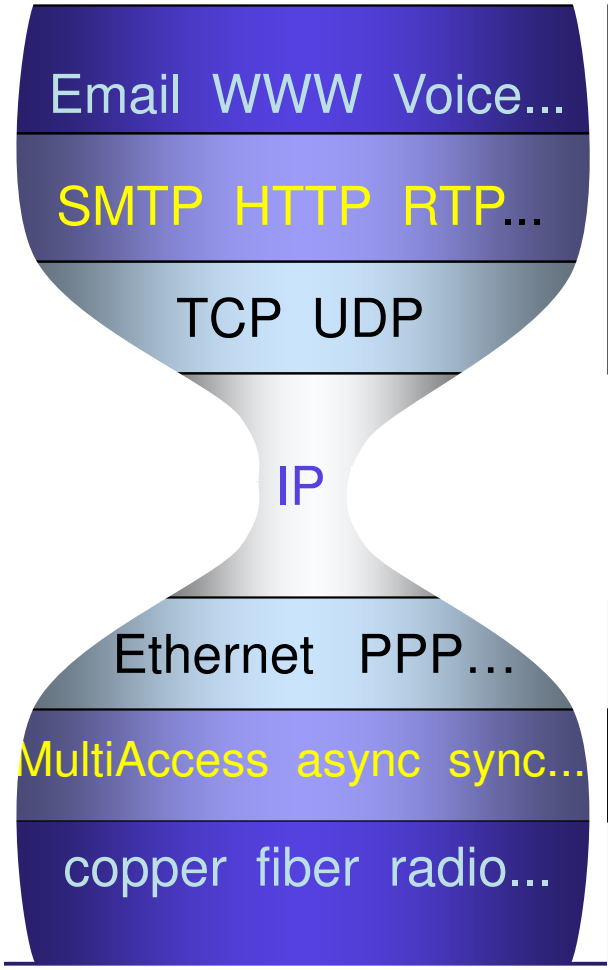
Protocollo IP



Internet Protocol Architecture

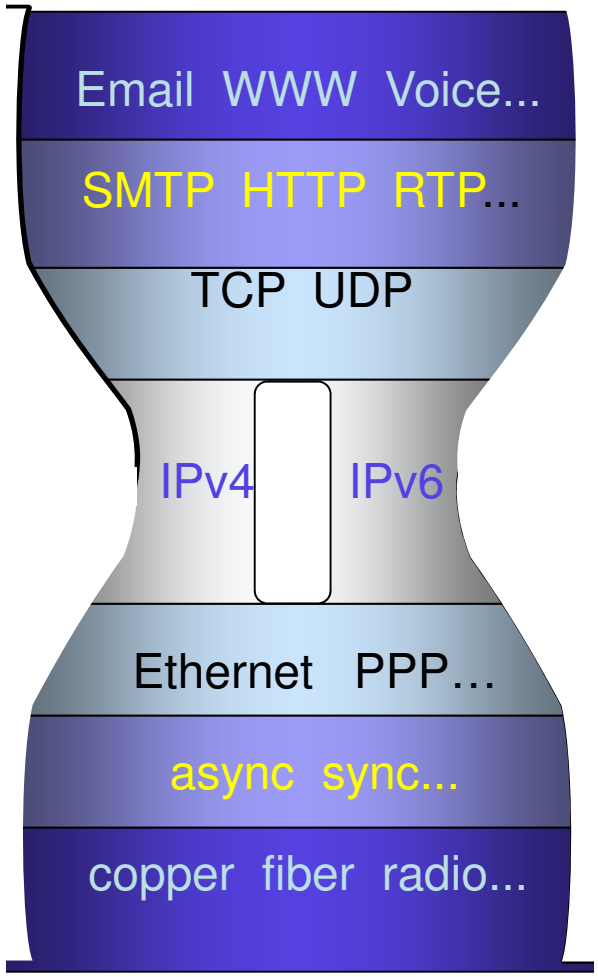


The Hourglass IP Model



- User Application
- End-to-end Application Protocol
- Transport Protocol
- Internet Layer
- Media Access Protocol
- Media Format
- Physical System

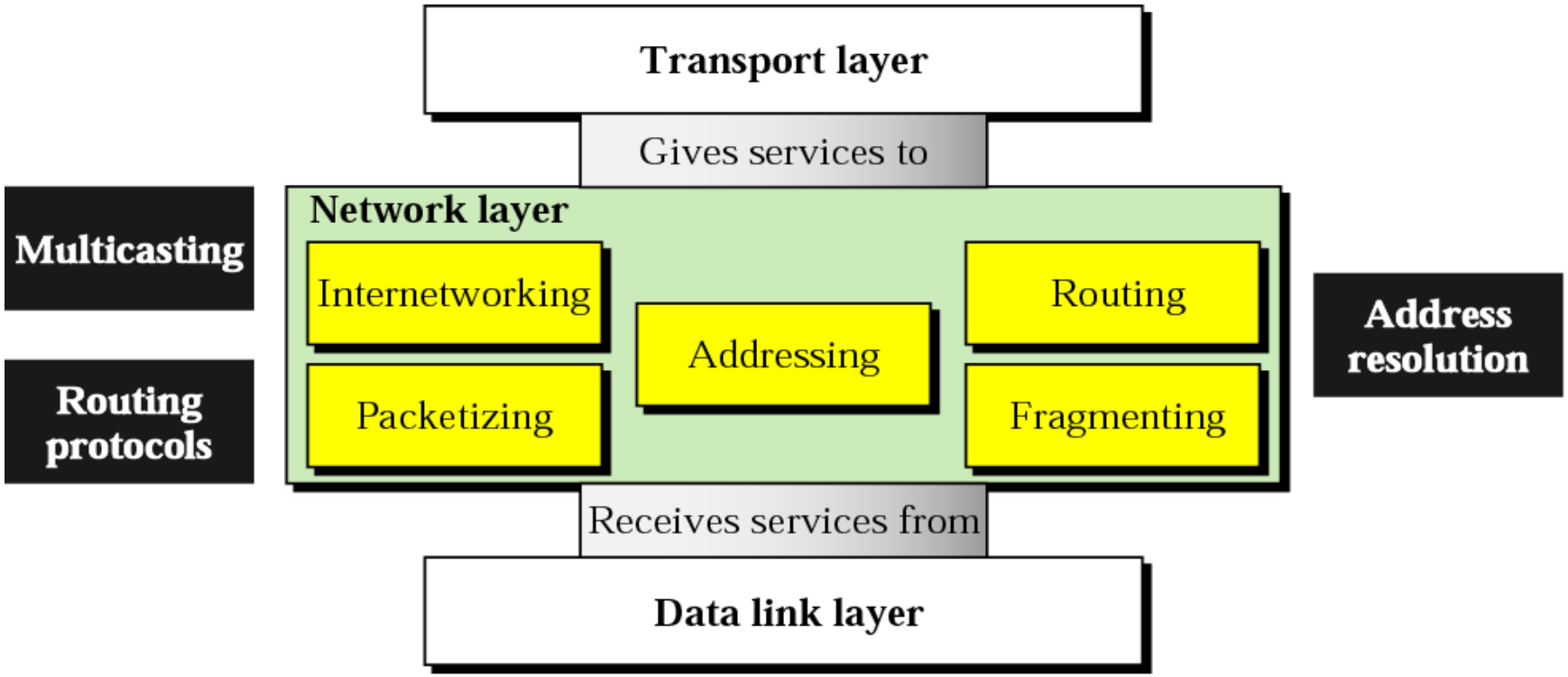
Ora abbiamo una crisi di identità (Mid-Life) !



L' introduzione di una transizione in IP : da V4 a V6

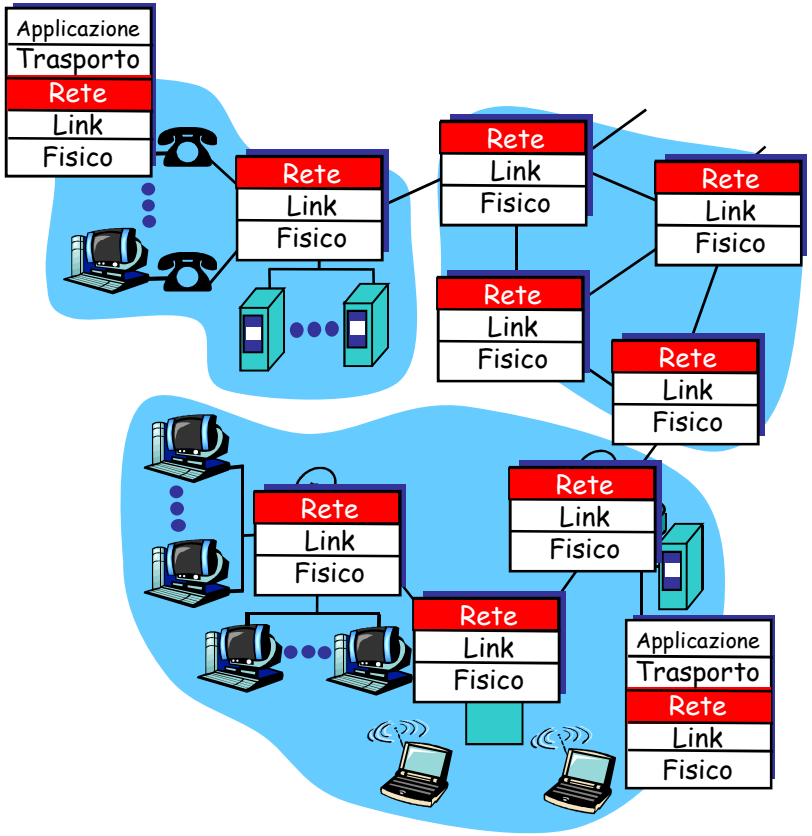
- **Raddoppia il numero delle interfacce di servizio**
- **Richiede cambiamenti sopra e sotto il livello IP**
- **Crea problemi di interoperabilità**

Network Layer



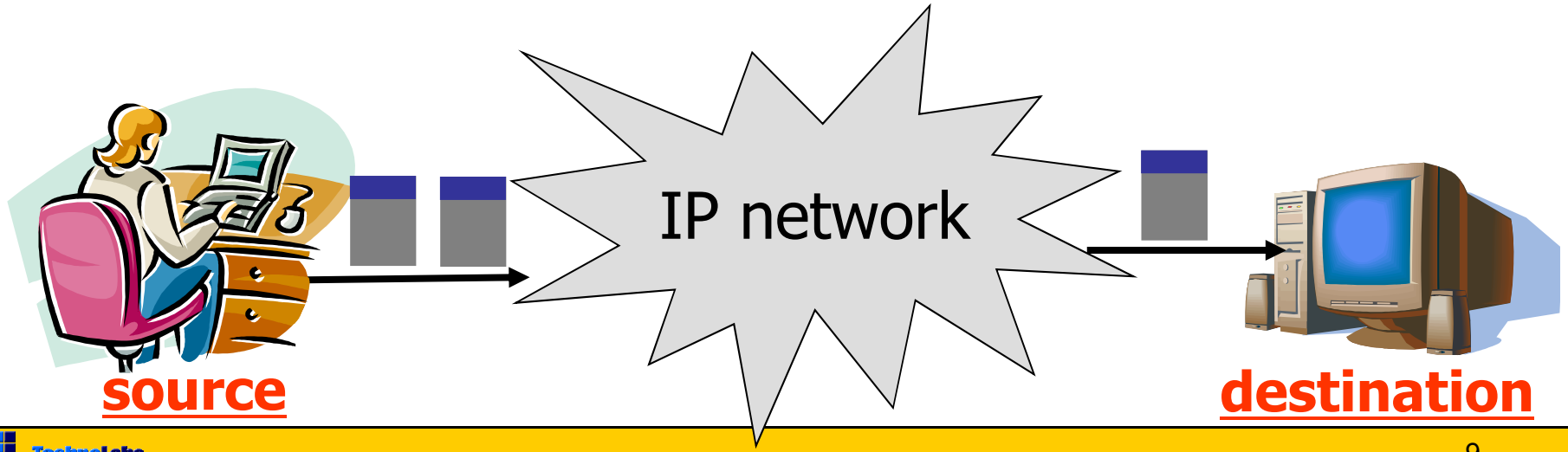
Livello di rete

- **Il livello di rete prende i segmenti dal livello di trasporto nell'host mittente**
- **Sul lato mittente, incapsula i segmenti in datagrammi**
- **Sul lato destinatario, consegna i segmenti al livello di trasporto**
- **Protocolli del livello di rete sono presenti in *ogni* host e router**
- **Il router esamina i campi intestazione in tutti i datagrammi IP che lo attraversano**



Servizio IP : Best-Effort Packet Delivery

- **Packet switching**
  - Divide i messaggi in una sequenza di packets
  - Header con source e destination address
- **Best-effort delivery**
  - I pacchetti possono perdersi
  - I pacchetti possono arrivare con errori
  - I pacchetti possono non essere consegnati nell'ordine originale



Modello di Servizio: perchè i pacchetti ?

- **Il traffico dati è 'bursty'**
  - Logging remoto
  - Messaggi e-mail
- **Non voglio sprecare banda**
  - Non c'è traffico durante i periodi di inattività
- **Meglio permettere il multiplexing**
  - Differenti trasferimenti condividono lo stesso link
- **I pacchetti possono essere consegnati da chiunque**
  - RFC 1149: IP Datagrams over Avian Carriers (aka piccioni)
- **… ma, il packet switching può essere inefficiente**
  - Consumo di bit extra in ogni pacchetto (header)

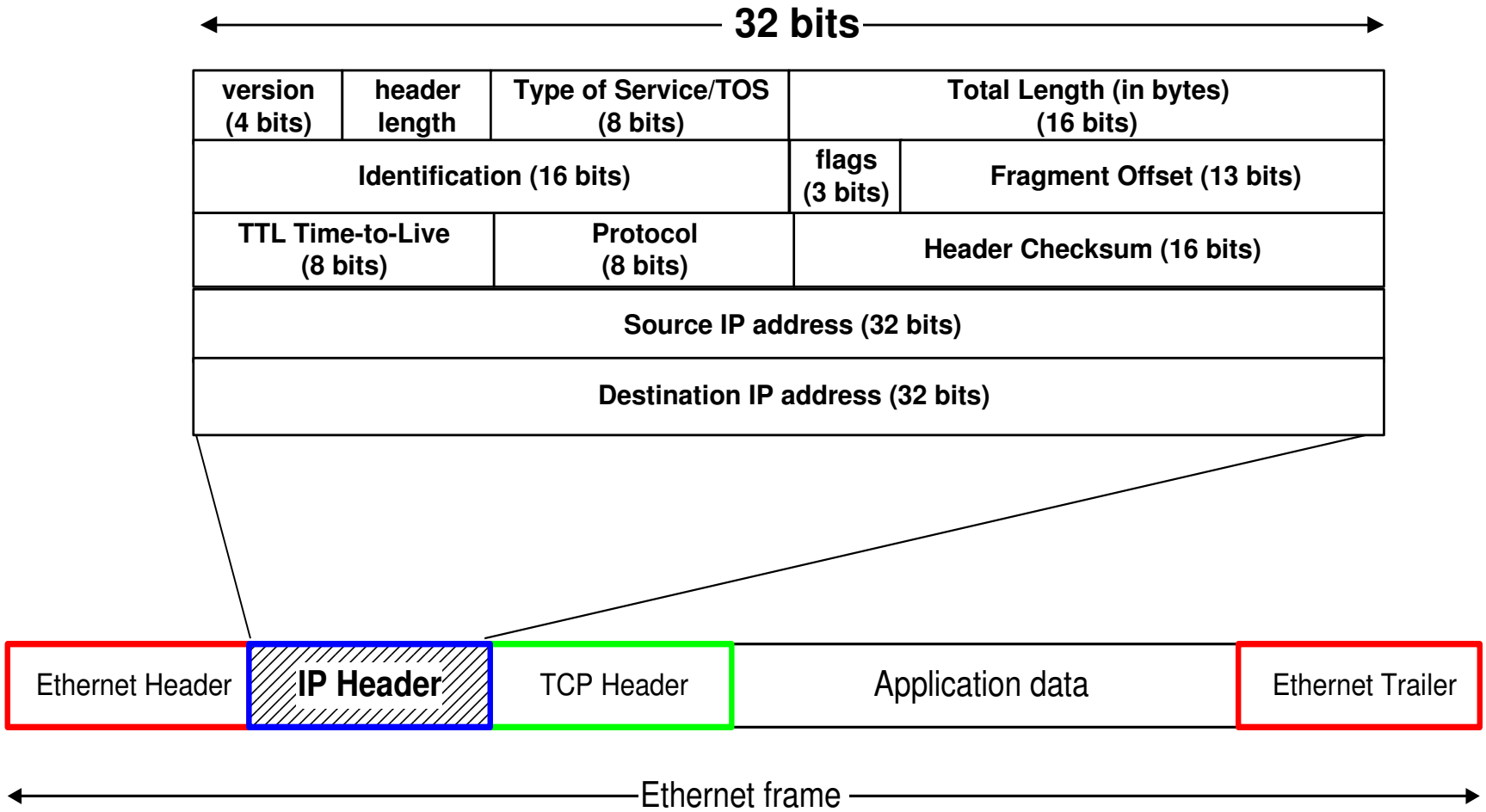
IP Service Model: Perchè Best-Effort?

- **IP vuol dire non dover mai dire “mi spiace” …**
  - Non necessita di riservare banda e memoria
  - Non necessita di fare error detection e/o correction
  - Non ha bisogno di ricordarsi che i pacchetti sono tra loro legati
- **Più facile sopravvivere a guasti**
  - Si tollerano perdite temporanee durante la fase di failover
- **… ma, le applicazioni vogliono efficienza, un trasferimento accurato, dati in ordine, in tempi ragionevoli**

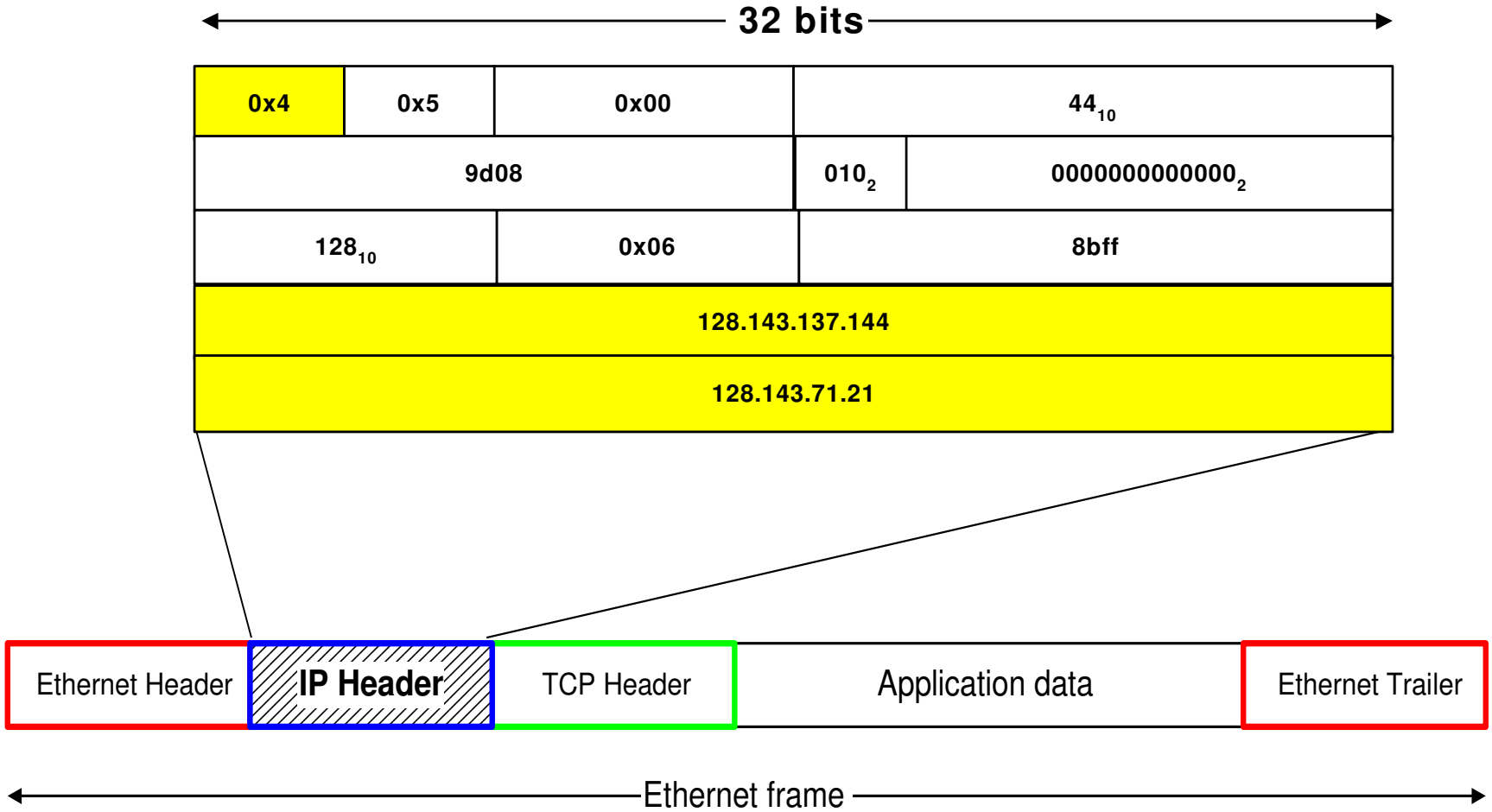
IP Service: Best-Effort è sufficiente ?

- **Assenza di error detection o correction**
  - Protocolli di più alto livello possono fare l'error checking
- **Pacchetti successivi possono fare percorsi diversi**
  - Non è un problema basta che i pacchetti arrivino a destinazione
- **I pacchetti possono essere consegnati non in ordine**
  - Il Receiver può ordinare i pacchetti (se necessario)
- **I pacchetti possono andare perduti o subire ritardi arbitrari**
  - Il Sender può rispedire i pacchetti (se desiderato)
- **Non c'è un controllo di congestione (escluso il "drop")**
  - Il Sender può rallentare a seguito di perdita o di ritardo

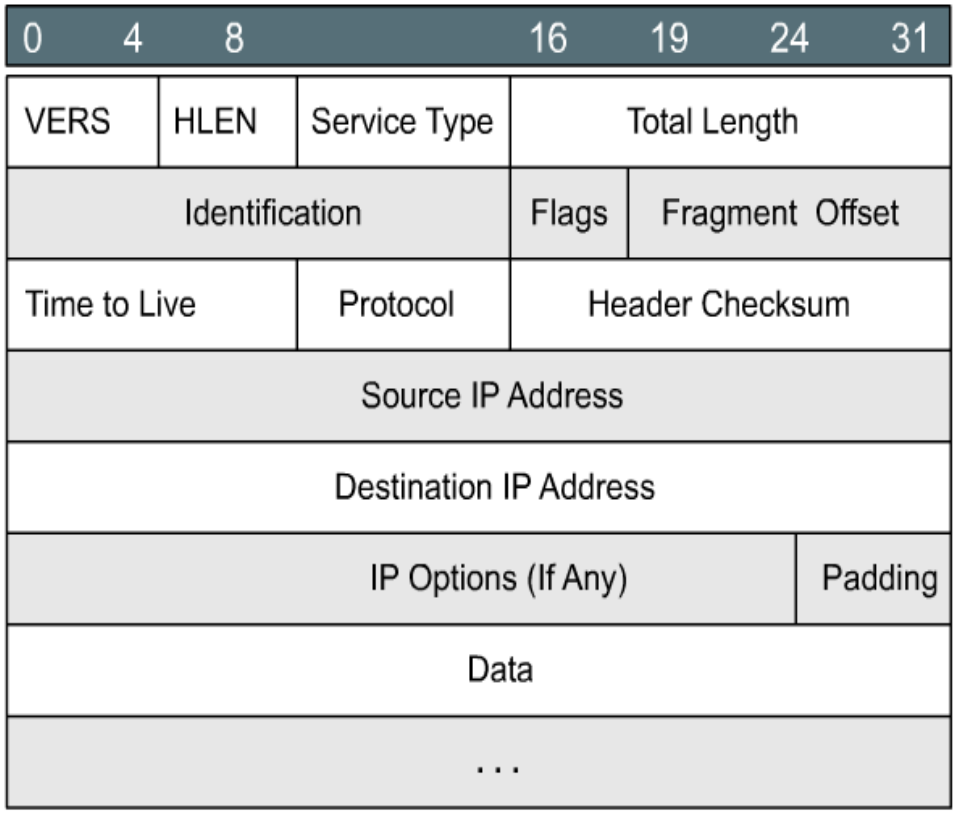
Pacchetto IP in una trama Ethernet



Indirizzi IPv4

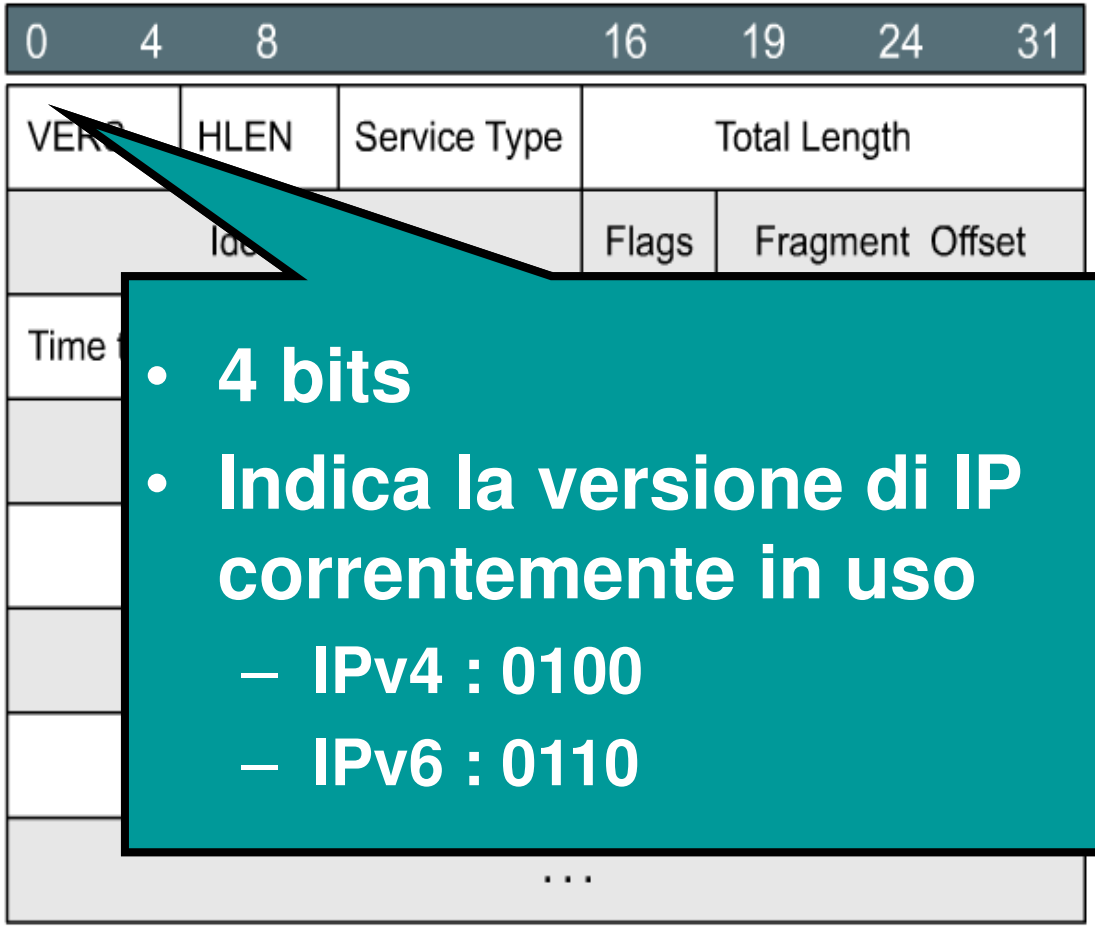


Internet Protocol (IPv4)



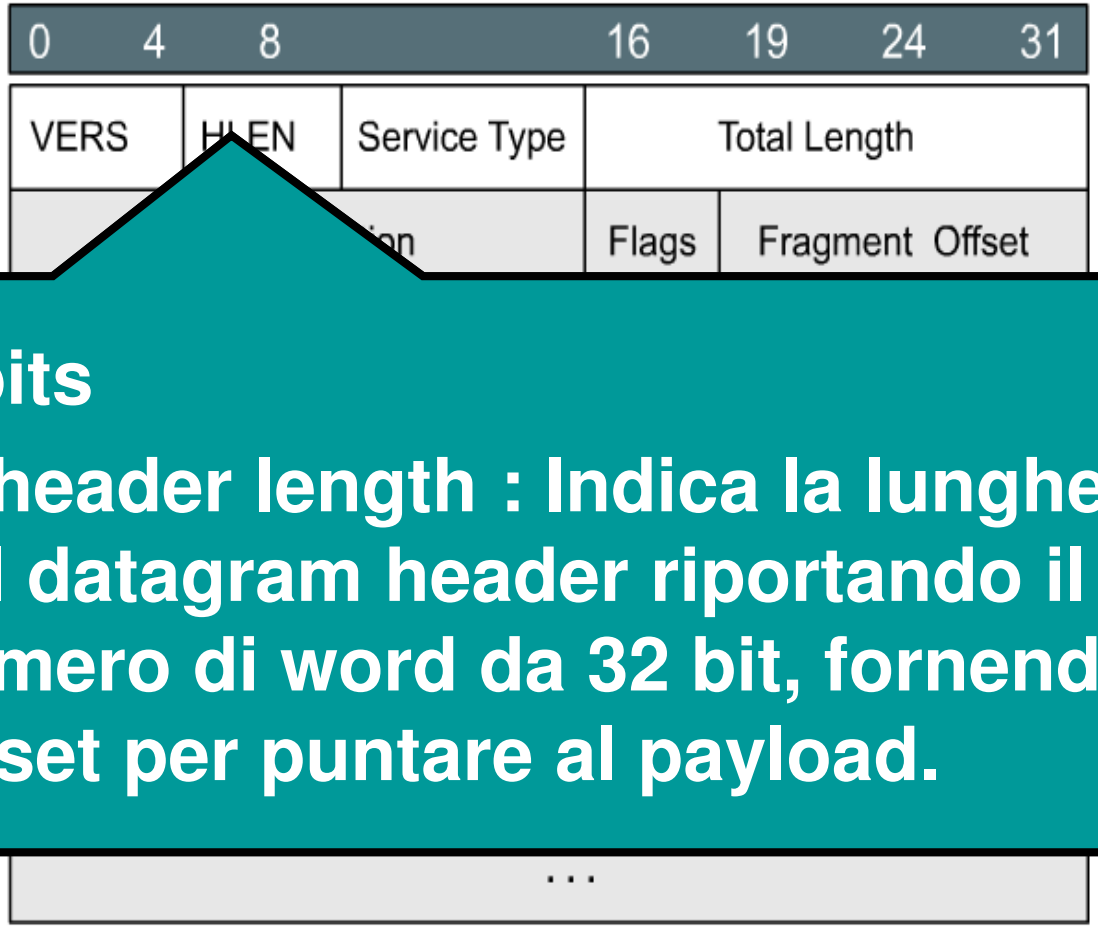
- **IP datagram**
  - Contiene le informazioni necessarie ai routers per trasferire dati tra due o più subnet

IP header format: Version



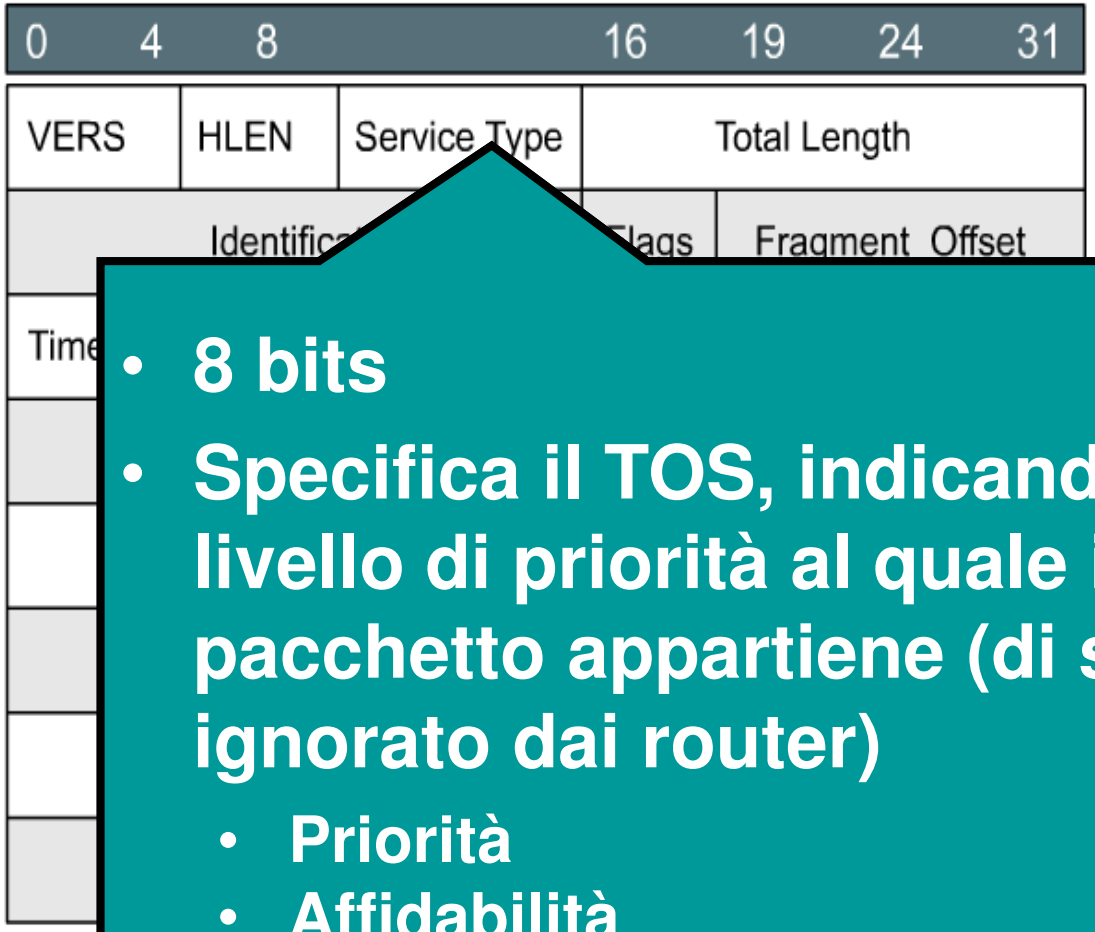
- **4 bits**
- **Indica la versione di IP correntemente in uso**
  - **IPv4 : 0100**
  - **IPv6 : 0110**

IP header format: Header length



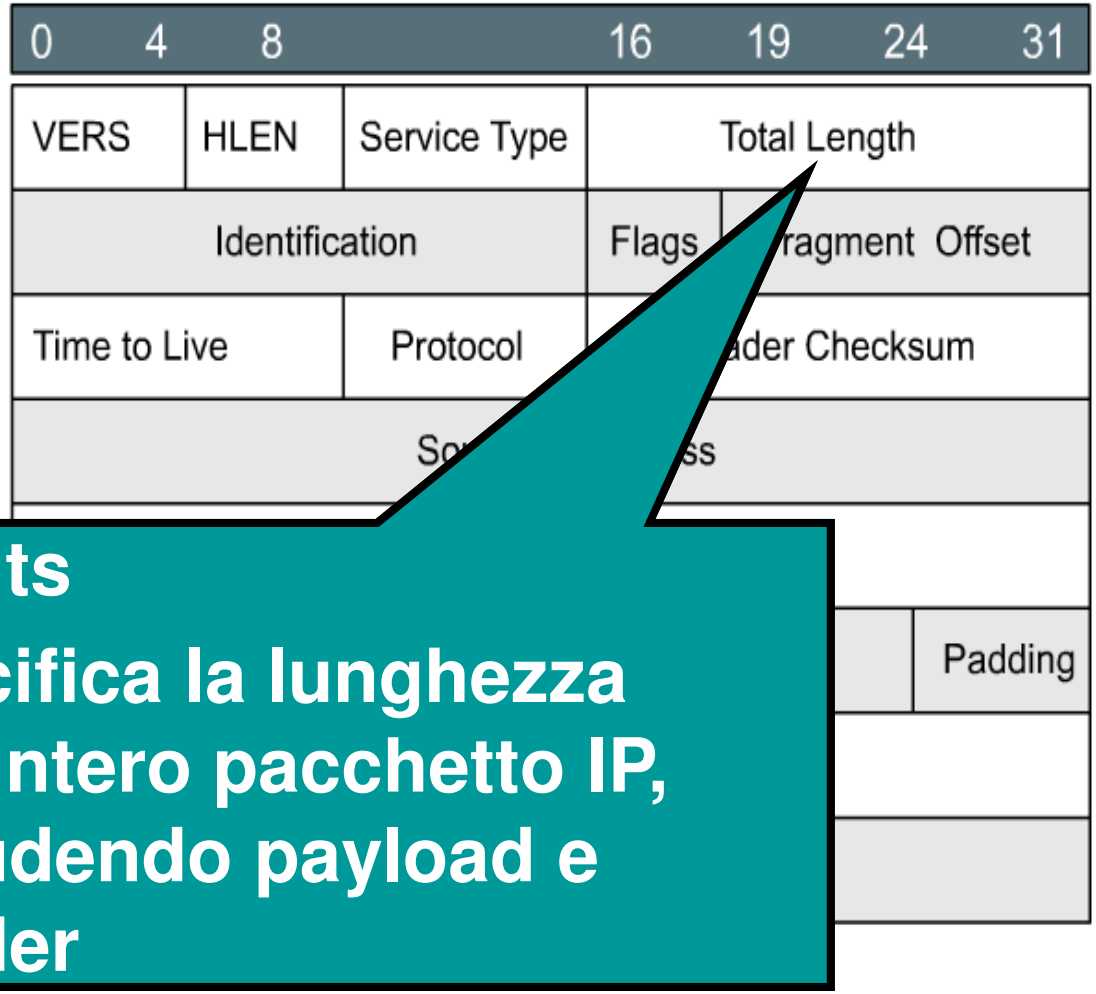
- **4 bits**
- **IP header length : Indica la lunghezza del datagram header riportando il numero di word da 32 bit, fornendo un offset per puntare al payload.**

IP header format: Service type



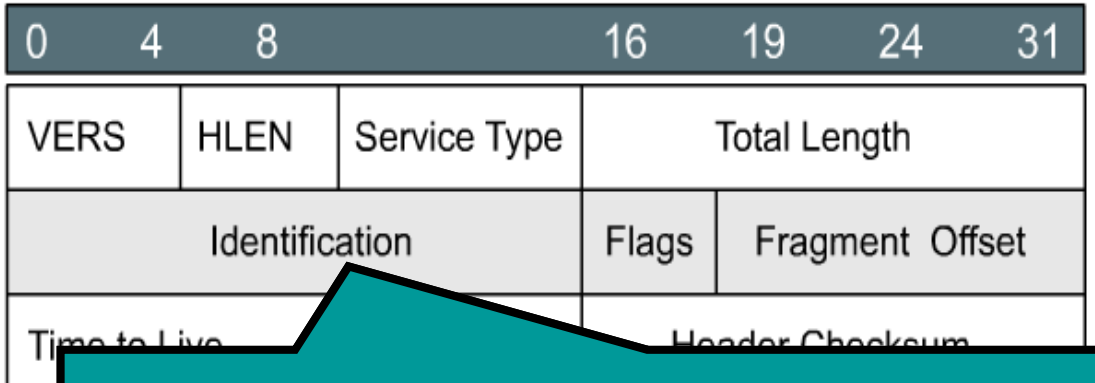
- 8 bits
- Specifica il TOS, indicando il livello di priorità al quale il pacchetto appartiene (di solito è ignorato dai router)
  - Priorità
  - Affidabilità
  - Velocità

IP header format: Total length



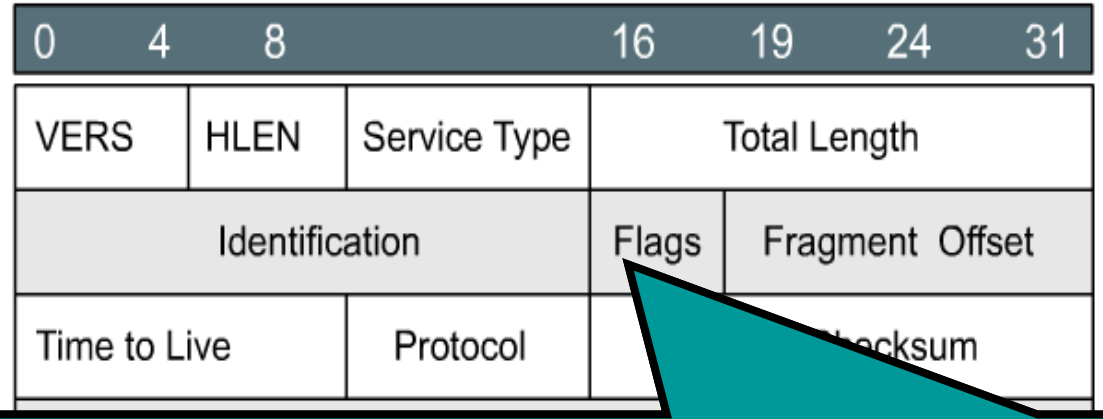
- **16 bits**
- **Specifica la lunghezza dell'intero pacchetto IP, includendo payload e header**

IP header format: Identification



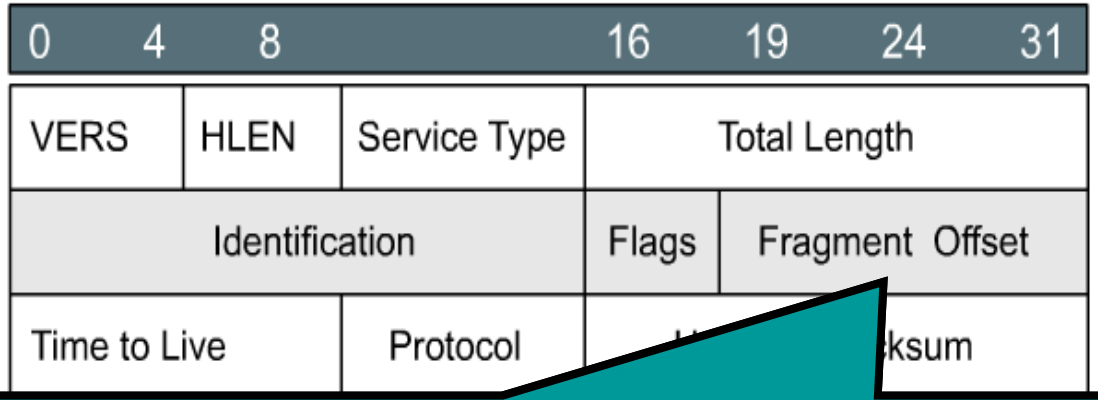
- **16 bits**
- **Contiene un intero progressivo che identifica il datagramma**
- **Assegnato alla sorgente per agevolare la ricostruzione dei frammenti di datagramma**

IP header format: Flags



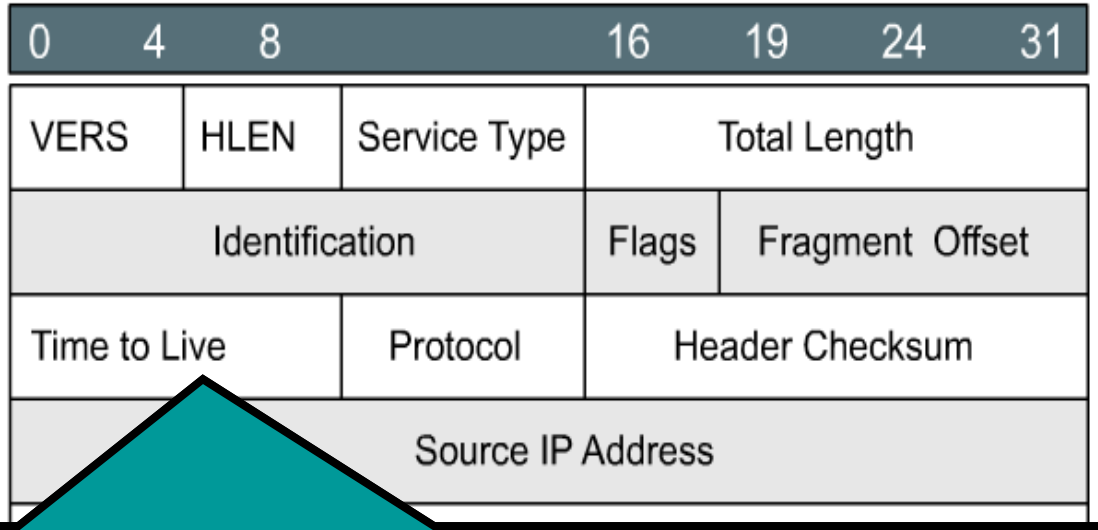
- **3 bits di cui il primo non in uso**
- **Il secondo bit indica se il pacchetto può essere frammentato (DF)**
- **L'ultimo indica se il datagramma corrente è l'ultimo della serie (MF)**

IP header format: Fragment offset



- **13 bits**
- **Indica il punto di congiunzione dove i datagrammi devono essere ricollegati (espresso in unità di 8 bytes)**
- **Il primo frammento ha offset ZERO!**

IP header format: Time to Live



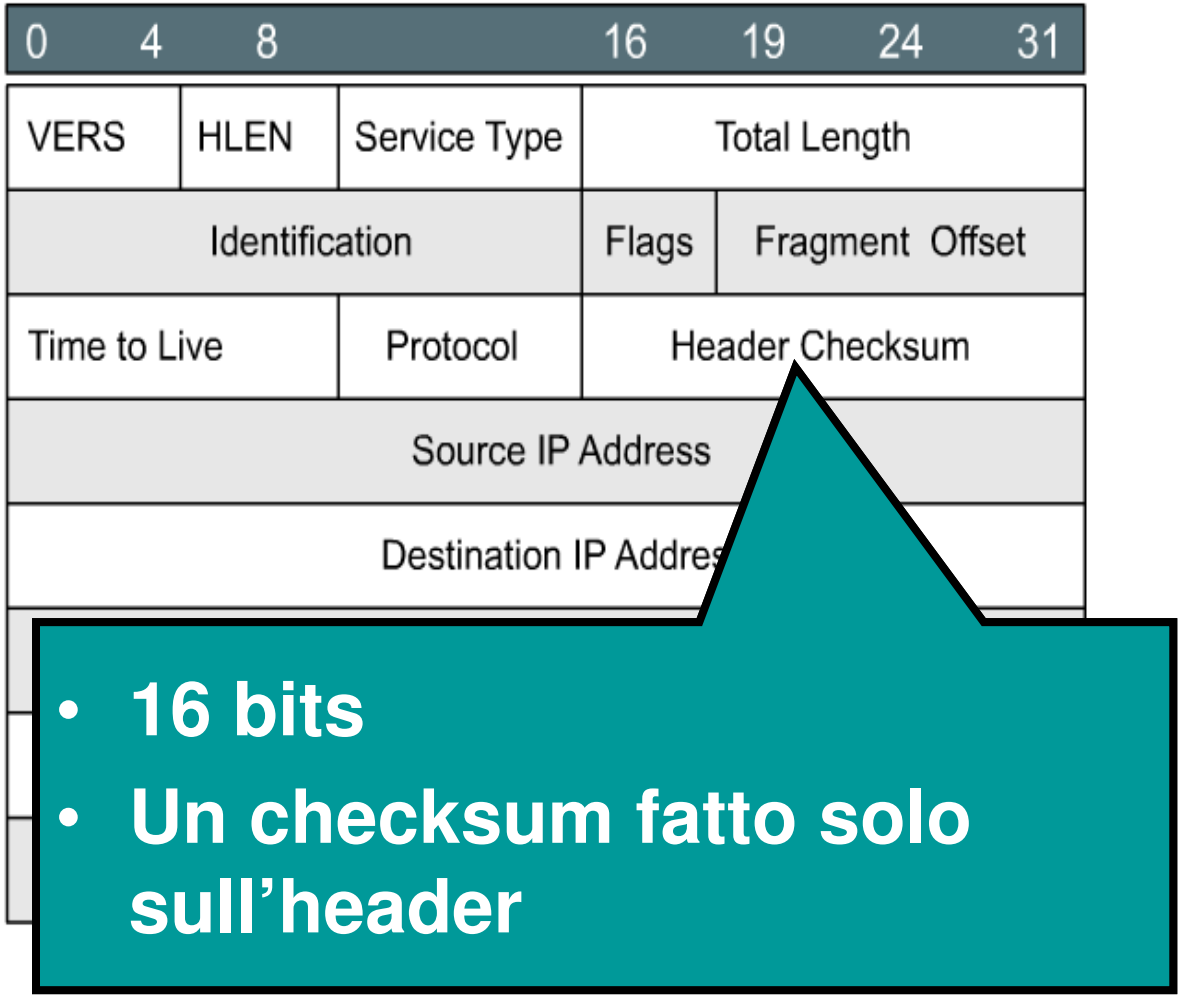
- **8 bits**
- **Time-to-Live contiene un counter che viene decrementato fino ad arrivare a zero, garantendo che un pacchetto non cicli a vita in una rete**

IP header format: Protocol

0	4	8	16	19	24	31
VERS	HLEN	Service Type	Total Length			
Identification			Flags	Fragment Offset		
Time to Live		Protocol	Header Checksum			

- **8 bits**
- **Indica a quale substrato superiore il payload verrà passato dopo l'aver processato ed epurato il datagram dalle informazioni necessarie a IP**
  - **06 : TCP**
  - **17 : UDP**

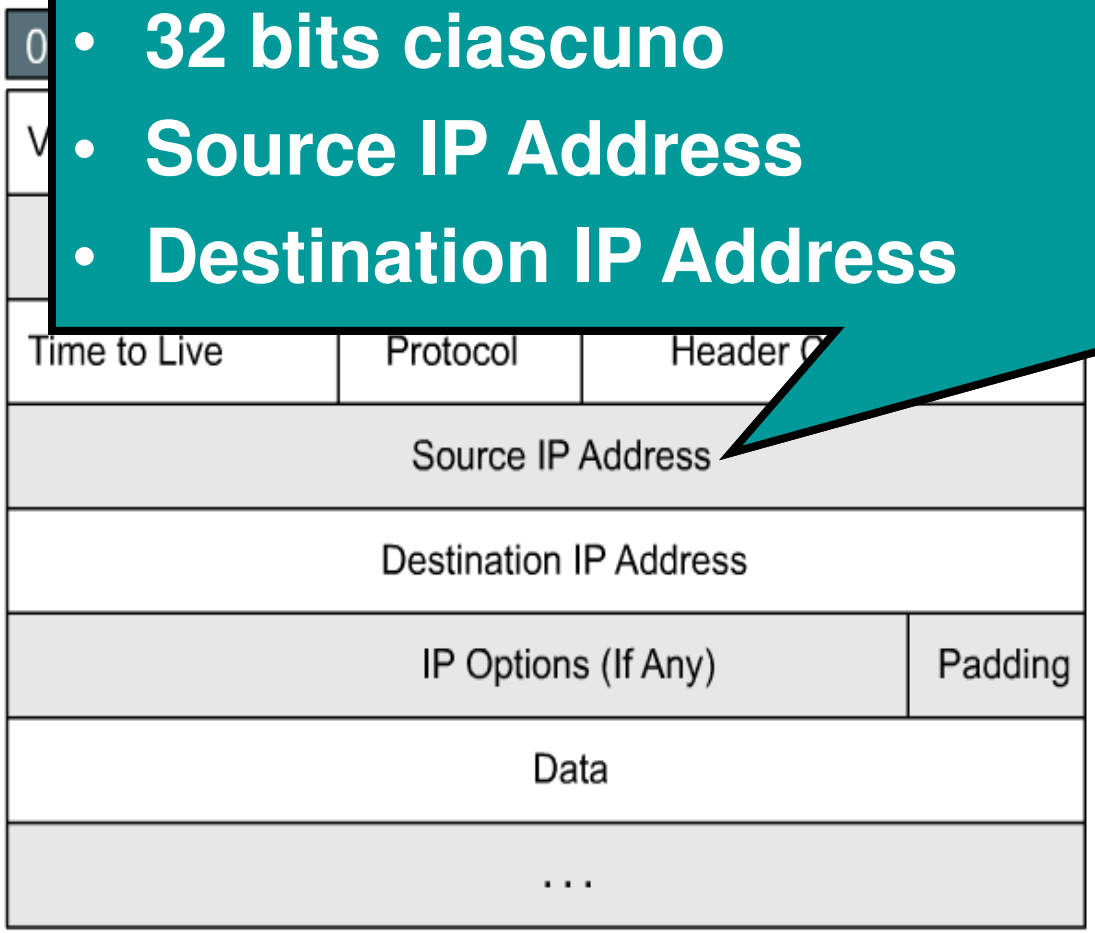
IP header format: Header checksum



- **16 bits**
- **Un checksum fatto solo sull'header**

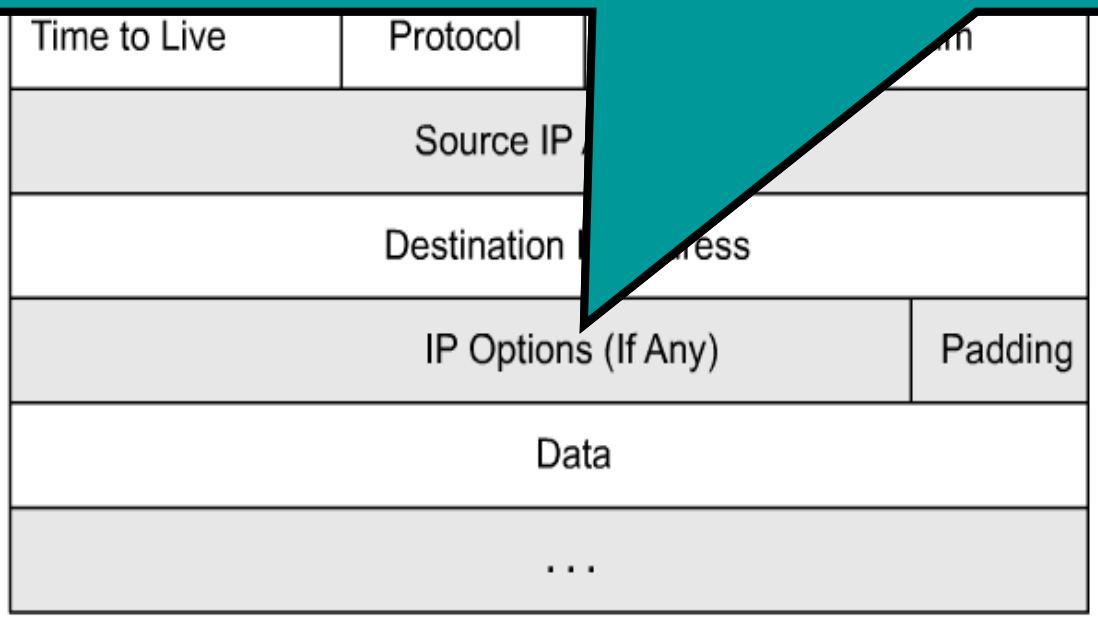
IP header format: Addresses

- 32 bits ciascuno
- Source IP Address
- Destination IP Address



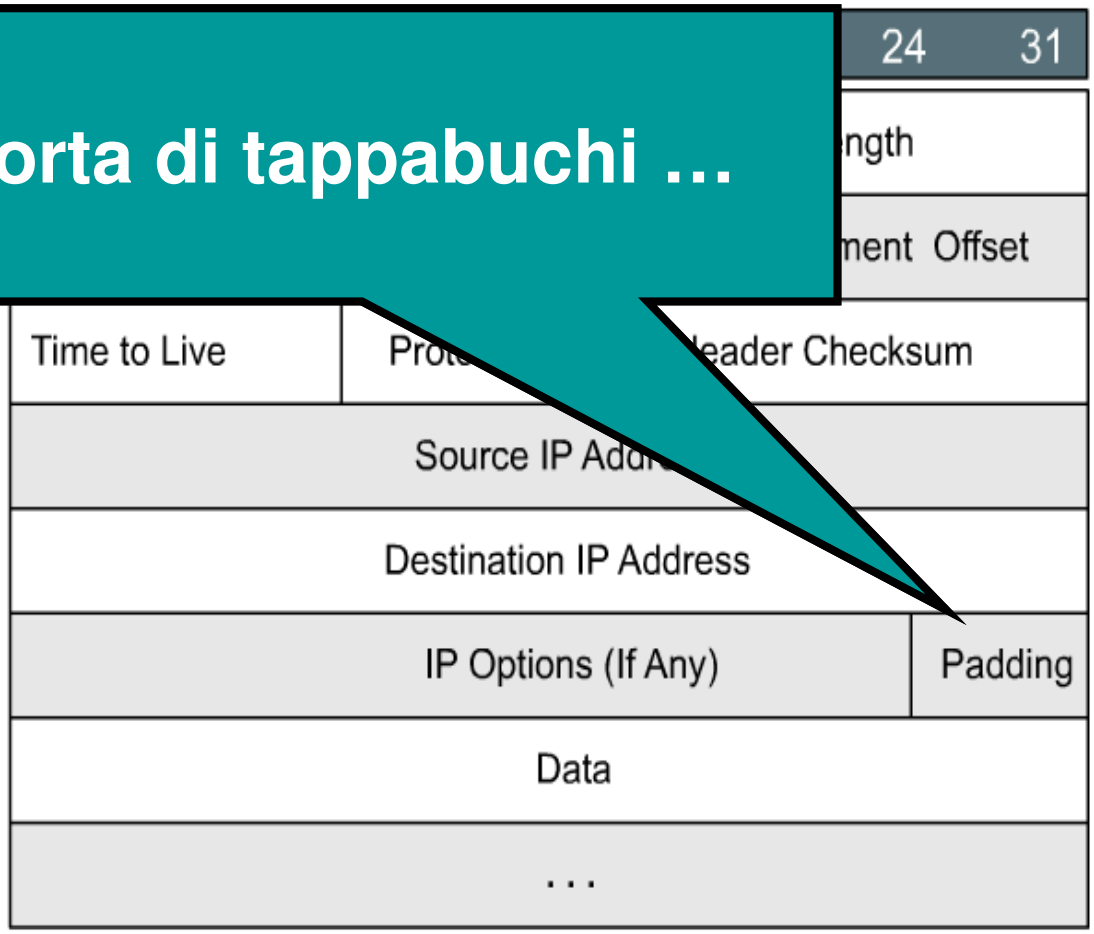
IP header format: Options

- Di lunghezza variabile
- Permette a IP di supportare molte opzioni tra cui security, routing, error report, molte altre …



IP header format: Padding

- **Una sorta di tappabuchi …**



IP Header: Version, Length, ToS

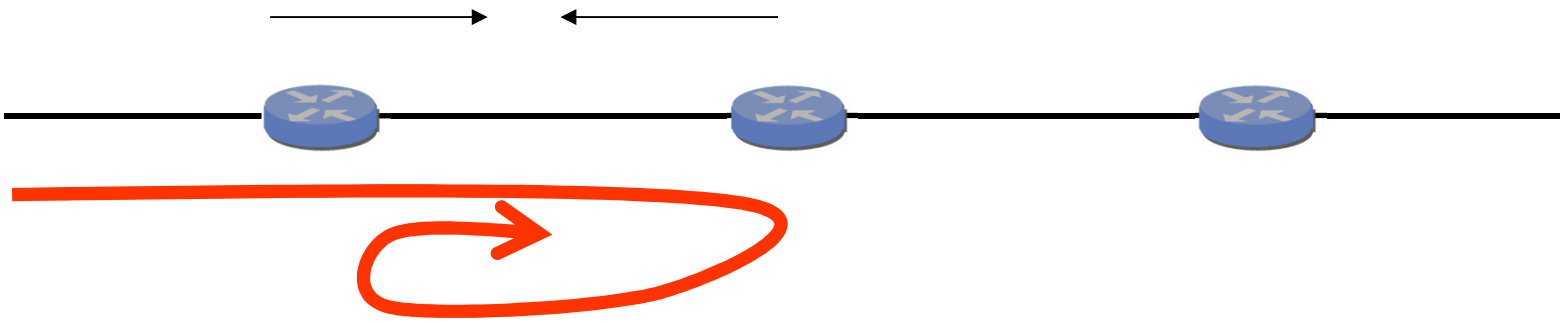
- **Version number (4 bits)**
  - Indica la versione di IP protocol
  - Necessario per sapere quali campi aspettarsi
  - Tipicamente “4” (per IPv4), e talvolta “6” (per IPv6)
- **Header length (4 bits)**
  - Numero di words da 32-bit nell’ header
  - Tipicamente “5” (per un 20-byte IPv4 header)
  - Può essere maggiore sse sono usate “IP options”
- **Type-of-Service (8 bits)**
  - Differenziare i packets in funzione dei servizi
  - E.g., low delay per audio, high bandwidth per FTP

IP Header: Length, Fragments, TTL

- **Total length (16 bits)**
  - Numero di bytes nel packet
  - Il valore Max è 63,535 bytes (2^{16} -1)
  - … sebbene il livello Datalink impone limiti più stringenti
- **Fragmentation information (16+3+13 bits)**
  - Packet identifier + flags + fragment offset
  - Supporta la divisione di un IP packet in fragments
  - Fragment offset indica l'offset (misurato in blocchi di 8 byte) di un particolare frammento relativamente all'inizio del datagramma IP originale: il primo frammento ha offset 0.
- **Time-To-Live (8 bits)**
  - Usato per identificare i packets persi in forwarding loops
  - … e alla fine scartarli dalla rete

IP Header: Time-to-Live (TTL)

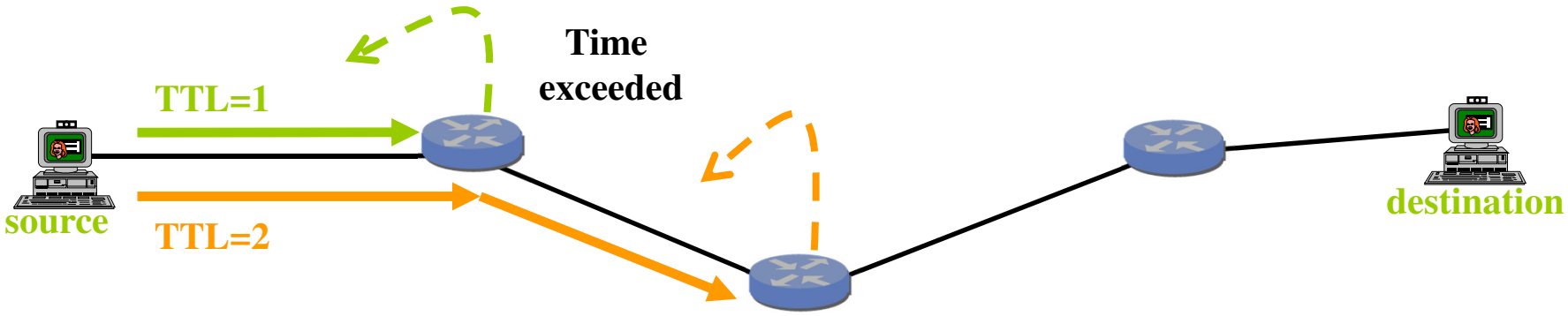
- **Potenziale problema**
  - Un forwarding loop può causare packets che girano per sempre
  - Confusione se il packet arriva con ritardo eccessivo



- **Campo Time-to-live nel packet header**
  - TTL è decrementato da ogni router sul cammino da Sorgente a Destinazione
  - Il Packet è scartato quando TTL diventa 0…
  - ...e un messaggio (ICMP) “time exceeded” è spedito alla sorgente

IP Header: Uso di TTL in Traceroute

- **Time-To-Live in IP**
  - Source spedisce un packet con TTL = *n*
  - Ogni router sul path decrementa TTL
  - "Time exceeded" viene spedito quando TTL = *0* e il packet viene scartato
- **Traceroute sfrutta le vicissitudini di TTL**



Spedisce packets con TTL=1, 2, … e memorizza l'IP del router che ha generato il messaggio "time exceeded" e il tempo intercorso

Esempio di Traceroute: da TL a Google

Nessuna risposta dal router

```
C:\> tracert www.google.it

Tracing route to www.l.google.com [209.85.129.104]
over a maximum of 30 hops:

 1    1 ms   <1 ms   <1 ms  141.29.169.17
 2    1 ms    1 ms   <1 ms  host19-155-static.82-213-b.business.telecomitalia.it [213.82.155.19]
 3  306 ms  290 ms  296 ms  host181-25-static.38-88-b.business.telecomitalia.it [88.38.25.181]
 4  310 ms  270 ms  228 ms  r-rm214-vl19.opb.interbusiness.it [80.21.7.24]
 5    *       *       *     Request timed out.
 6    *       *       *     Request timed out.
 7  315 ms  280 ms  276 ms  85.36.9.57
 8  303 ms  289 ms  272 ms  mil52-ibs-resid-2-it.mil.seabone.net [195.22.196.145]
 9  265 ms  347 ms  313 ms  72.14.196.141
10  276 ms  318 ms  291 ms  209.85.251.108
11  334 ms  332 ms  308 ms  209.85.251.113
12  292 ms  289 ms  308 ms  72.14.232.165
13  245 ms  244 ms  293 ms  72.14.233.210
14  272 ms  245 ms  291 ms  fk-in-f104.google.com [209.85.129.104]

Trace complete.
```

IP Header Fields: Transport Protocol

- **Protocollo (8 bits)**
  - Identifica il protocollo trasportato
    - E.g., “6” per Transmission Control Protocol (TCP)
    - E.g., “17” per User Datagram Protocol (UDP)
  - Importante per il demultiplexing lato ricevente
    - Indica che tipo di header aspettarsi

IP Header: Header Checksum

- **Checksum (16 bits)**
  - Somma tutte le 16-bit words nell' header IP (solo l'Header non i dati)
  - Se un bit dell' header è alterato in transito
  - … la checksum sarà diversa da quella in ricezione
  - L' host ricevente scarta i packets corrotti
    - L'host trasmittente dovrà ritrasmettere il packet, se richiesto

```
   134
 + 212
 -----
 = 346
```

```
   134
 + 216
 -----
 = 350
```

Mismatch!

IP Header: To and From Addresses

- **Due indirizzi IP**
  - Source IP address (32 bits)
  - Destination IP address (32 bits)
- **Destination address**
  - Identificatore unico per l'host ricevente
  - Permette ad ogni nodo di prendere le decisioni di instradamento
- **Source address**
  - Identificatore unico per l'host trasmittente
  - Il ricevente può decidere se accettare il packet
  - Abilita il ricevente a rispondere al trasmittente

Source Address: Cosa succede se il Source mente ?

- **Source address dovrebbe essere quello del sending host**
  - Ma, chi controlla ?
  - Voi potete generare packets con qualunque source
- **Perchè qualcuno dovrebbe fare ciò ?**
  - Lancio di un attacco denial-of-service
    - Spedire un eccessivo numero di packets alla destinazione
    - … sovraccaricare il nodo, o i links che portano al nodo
  - Evitare di essere intercettati (by “spoofing”)
    - Infatti, la vittima vi identificherebbe dal source address
    - ..ecco perchè mettere il source address di qualcun altro nei packets
  - Oppure, scatenare un attacco contro lo ‘spoofed host’
    - Lo spoofed host è erroneamente accusato
    - Lo spoofed host riceve il traffico di ritorno dal receiver

Maximum Transmission Unit

- **Un datagramma IP può arrivare a 65535 bytes, ma il data link layer impone un limite che è molto inferiore**
  - Per esempio:
    - Una frame Ethernet ha un payload massimo di 1500 bytes
- **Il limite imposto dal data link layer è detto** maximum transmission unit (MTU)
- **MTU per vari data link layers:**

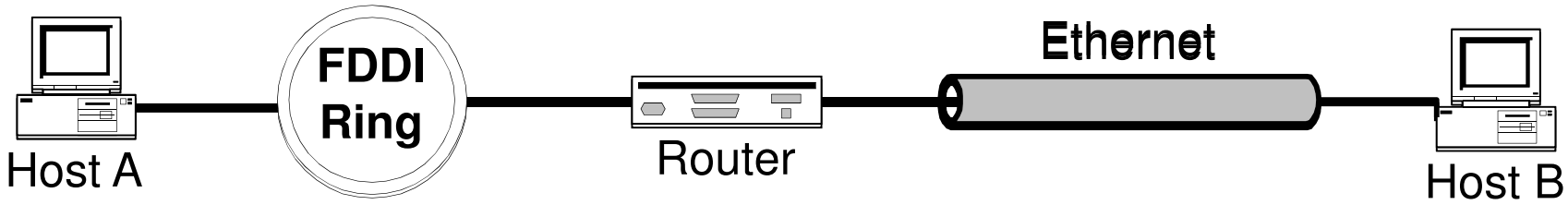
Ethernet:	1500	FDDI:	4352
802.3:	1492	ATM AAL5:	9180
802.5:	4464	PPP:	296

- **Cosa succede se un datagramma IP eccede la MTU ?**
  - Il datagramma IP è frammentato in unità più piccole.
- **Cosa se il cammino include reti con differenti MTU ?**

Dimensioni tipiche della MTU

Network	MTU (bytes)
IBM token ring 16Mbps	17914
4 Mbps Token Ring (IEEE 802.5)	4464
FDDI	4352
Ethernet	1500
IEEE 802.2 802.3	1492
X.25	576
point to point (PPP, SLIP)	May be set to 296 for interactive use

Frammentazione IP

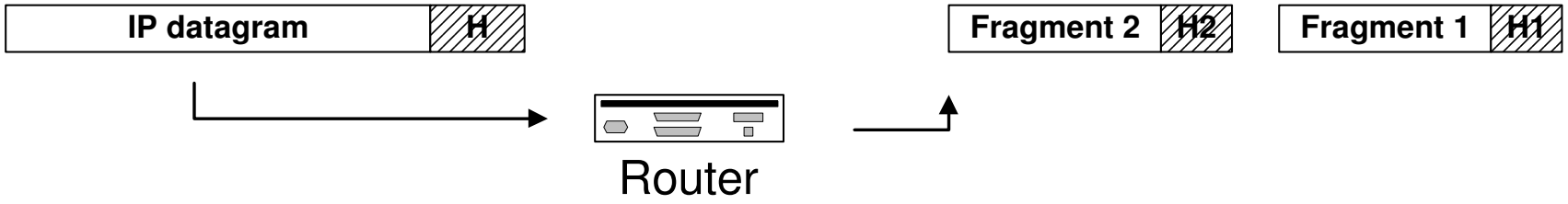


MTUs: FDDI: 4352 Ethernet: 1500

- **Frammentazione**:
  Il router IP spezza il datagramma in tanti chunks auto-consistenti (frammenti)
  I frammenti sono riassemblati dal receiver

Dove avviene la Frammentazione ?

- La Frammentazione viene fatta o dal sender o dai routers intermedi
- Lo stesso datagramma può essere frammentato più volte.
- Riassemblaggio dei datagrammi originali solo al destination host !!



Campi coinvolti nella Frammentazione

- **I seguenti campi nell'header IP sono coinvolti:**

<table>
<tr><td>version</td><td>header length</td><td>DS</td><td>ECN</td><td colspan="4">total length (in bytes)</td></tr>
<tr><td colspan="4">Identification</td><td>0</td><td>DF</td><td>MF</td><td>Fragment offset</td></tr>
<tr><td colspan="2">time-to-live (TTL)</td><td colspan="2">protocol</td><td colspan="4">header checksum</td></tr>
</table>

Identification — Quando un datagramma è frammentato, l'identification è lo stesso per tutti i frammenti

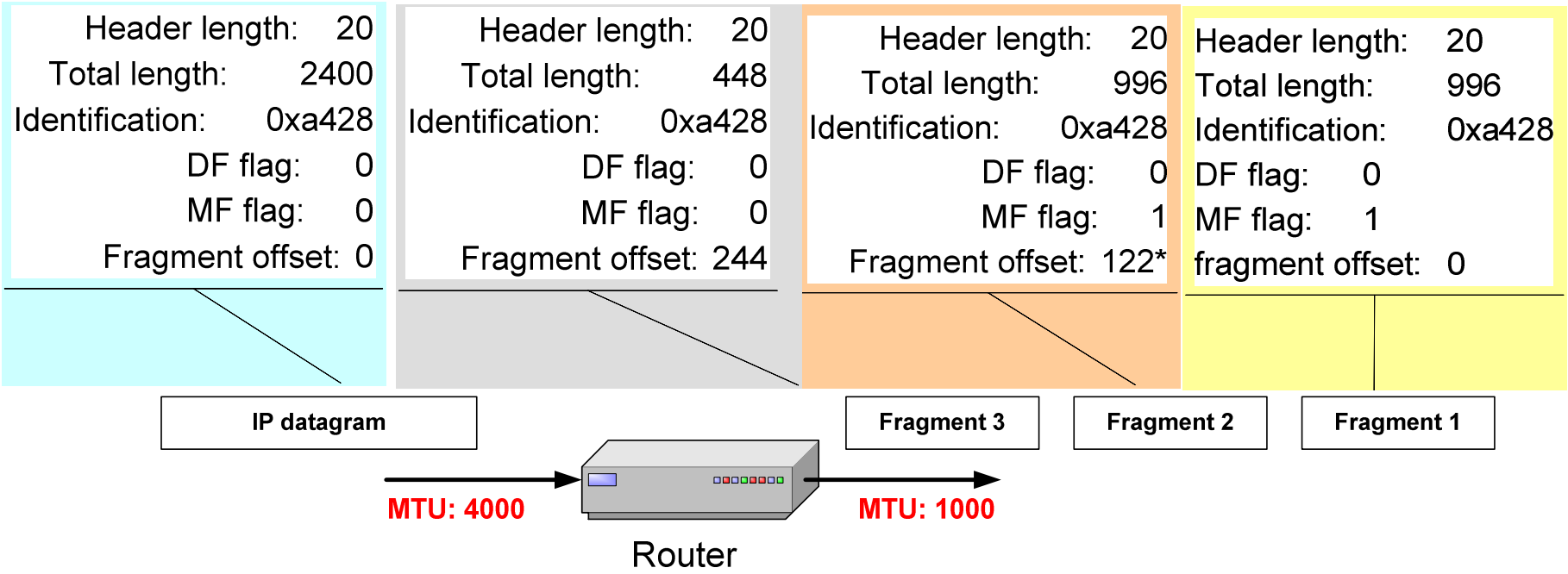
Flags

DF bit set: Datagramma non può essere frammentato e deve essere scartato se MTU è piccola

MF bit set: Datagramma parte di un frammento e altri frammenti sono in arrivo

Esempio di Frammentazione

Un datagramma da 2400 bytes deve essere frammentato perchè la MTU ha un limite di 1000 bytes



*** 122 * 8 = 976 + 20 = 996 bytes** **976 bytes di payload + 20 di header**

Fragment Offset

U	M	D	Fragment Offset												
X	X	X	0	0	0	0	0	0	1	1	1	1	0	1	0
0	0	0	0	0	0	1	1	1	1	0	1	0	0	0	0

976 >> 3 = 122

IP Addressing e Forwarding

Luigi Vetrano

TechnoLabs

Obiettivi

- **IP addresses**
  - Notazione Dotted-quad → AAA.BBB.CCC.DDD
  - Prefissi IP per l'aggregazione
- **Allocazione degli Indirizzi**
  - Classful addresses
  - Classless InterDomain Routing (CIDR)
- **Packet forwarding**
  - Tabelle di Forwarding
  - Longest-prefix match forwarding

- **Da dove derivano le forwarding tables ?**

IP Address (IPv4)

- Un numero unico da 32-bit
- Identifica una interfaccia (host/router, …)
- Rappresentato in notazione dotted-quad (decimale puntata)

12	34	158	5
00001100	00100010	10011110	00000101

IP address (IPV4)

- 32 bits – un “long binary integer”, per esempio:

  11000000101010001111111101100100

- Interpretato come quattro bytes, 8-bit numbers, ognuno dei quali lo chiamiamo “ottetto”
- Si riporta l’ IP address con il carattere “.” tra gli ottetti

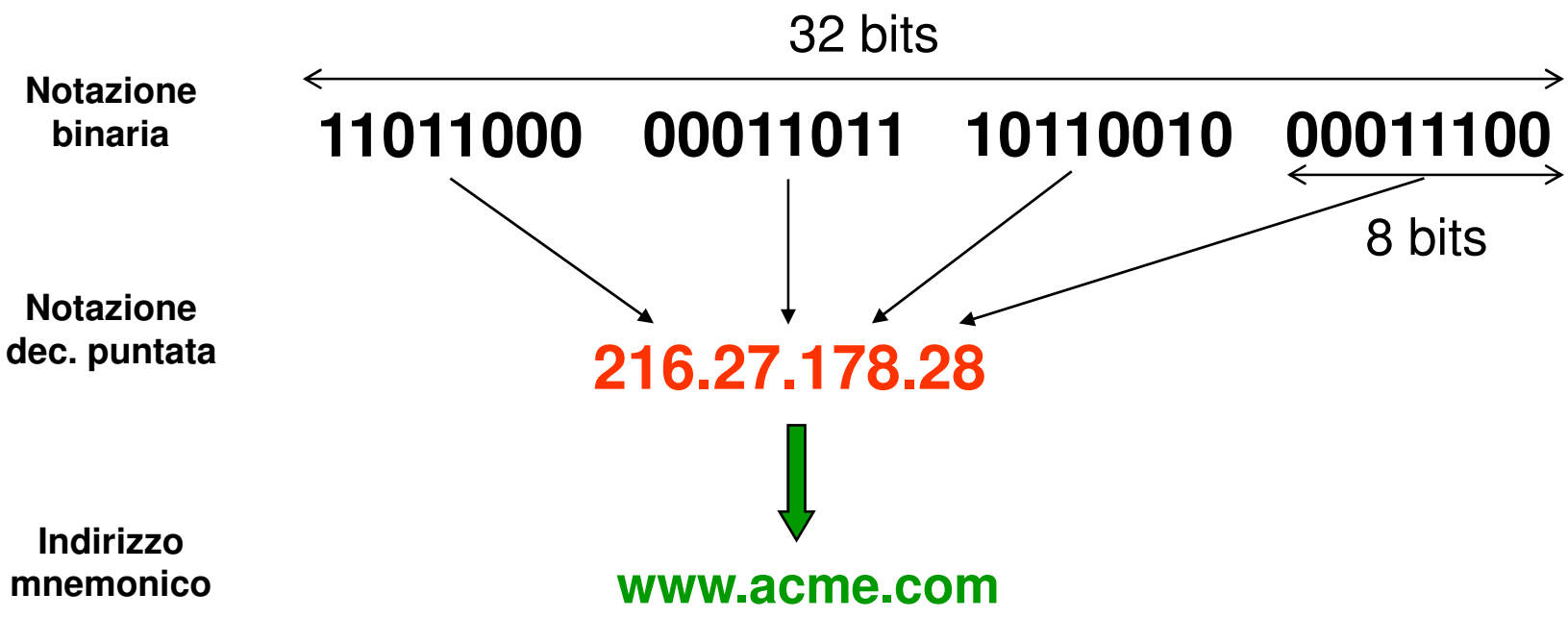
  11000000 . 10101000 . 11111111 . 01100100

- Solitamente ogni ottetto è espresso in decimale, e ogni valore decimale è separato da un punto (dot)
  - Notazione “Decimale Puntata” o “Dotted decimal”
- Finalmente, l’ IP di sopra in notazione dotted decimal:

  192.168.255.100

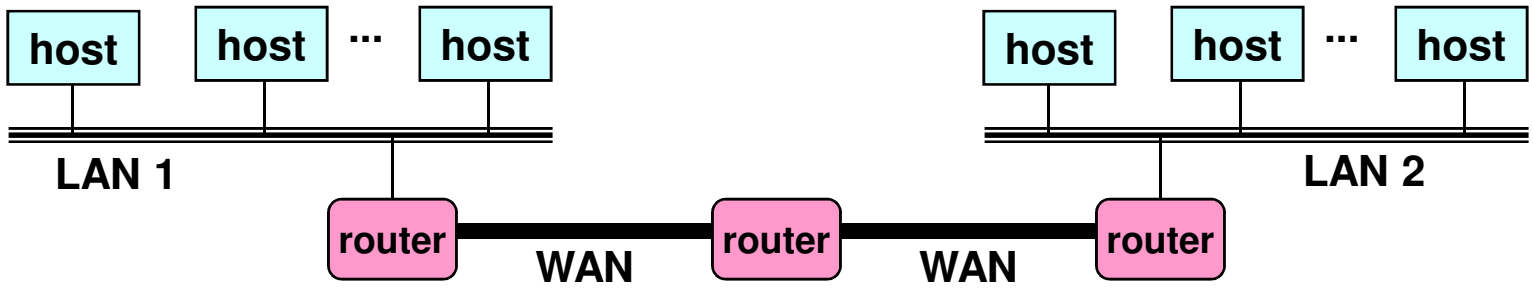
Schema di indirizzamento

- **Un indirizzo IPv4 è espresso in stringhe di 32 bit ...**
  … che possono essere espresse in notazione decimale puntata (dotted)
- **Ad un indirizzo IP può essere associato un nome (DNS)**



Raggruppare Hosts

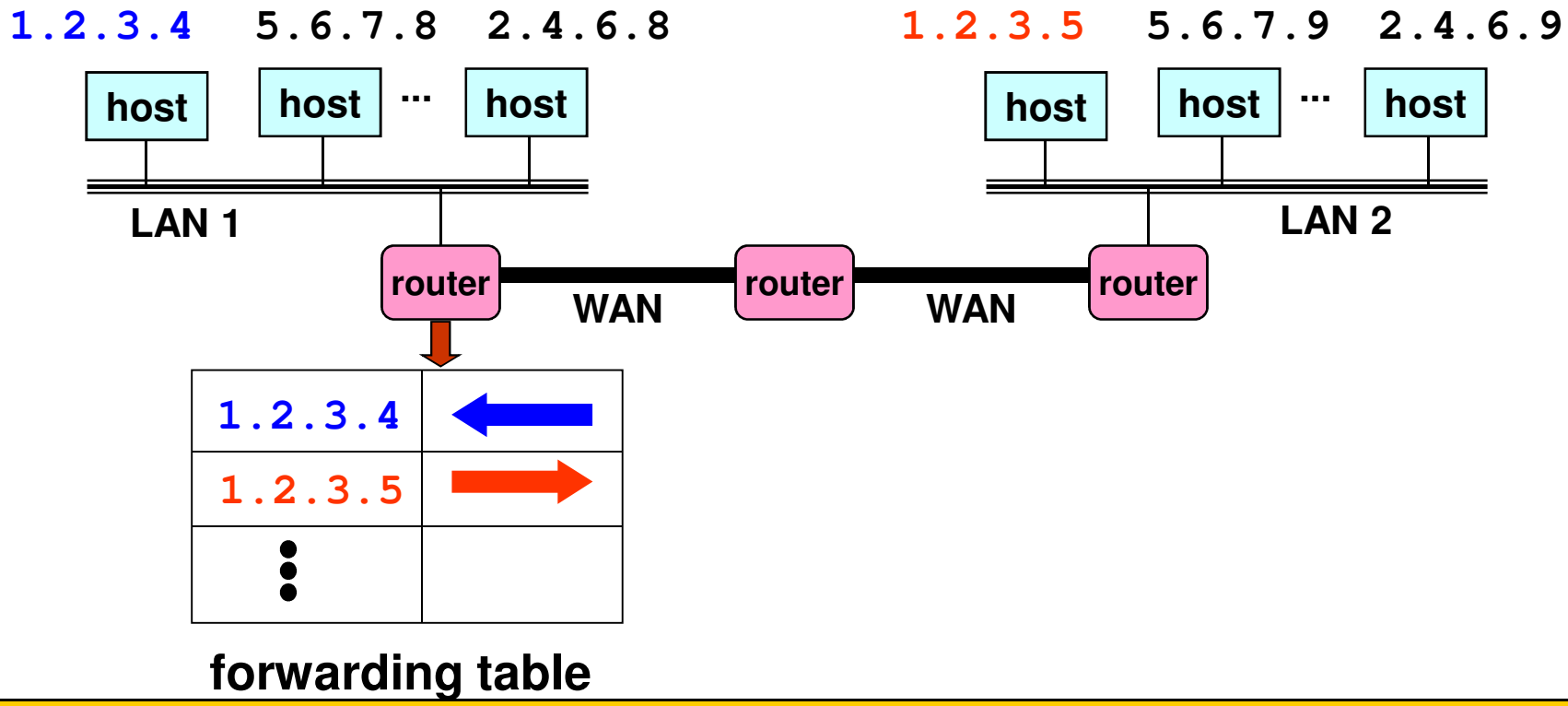
- **Internet è una "inter-network"**
  - Usata per connettere *networks*, non *hosts*
  - Necessita di un meccanismo per indirizzare una network (i.e., gruppo di hosts)



LAN = Local Area Network
WAN = Wide Area Network

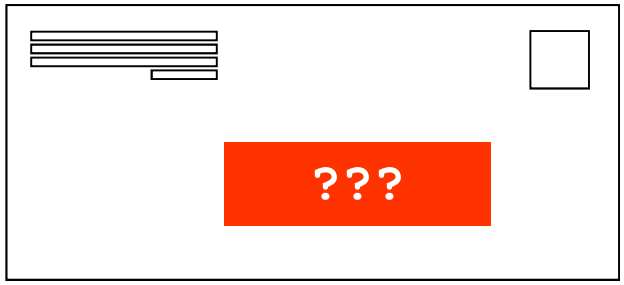
Sfida della Scalabilità

- **Supponiamo di assegnare agli hosts indirizzi arbitrari**
  - Allora ogni router necessita di memorizzare tante informazioni
  - …per conoscere come indirizzare i packets verso l'host



Indirizzamento gerarchico in snail Mail

- **Indirizzamento in snail mail**
  - CAP: 80033
  - Via: G. Mazzini
  - Numero Civico: 35
  - Interno: 306
  - Nome del destinatario: Mario Bianchi

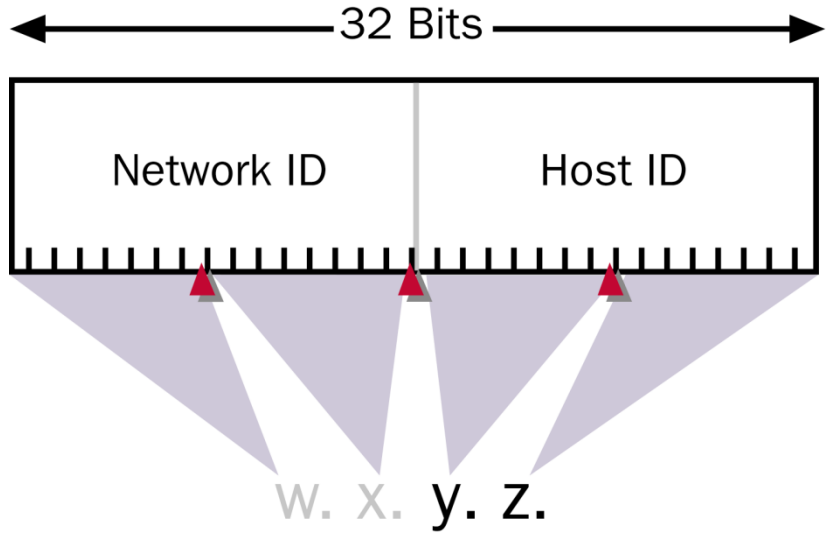


- **Forwarding**
  - Consegna della lettera all'ufficio postale (CAP)
  - Consegna della lettera al postino che copre la via
  - Lettera nella mailbox del civico 35
  - Portiere dà la lettera alla persona appropriata (interno 306)

Pensiamo ad IP come indirizzo composto

Una parte dei 32 bits rappresenta un ID di RETE

La parte rimanente la usiamo per rappresentare un HOST della RETE

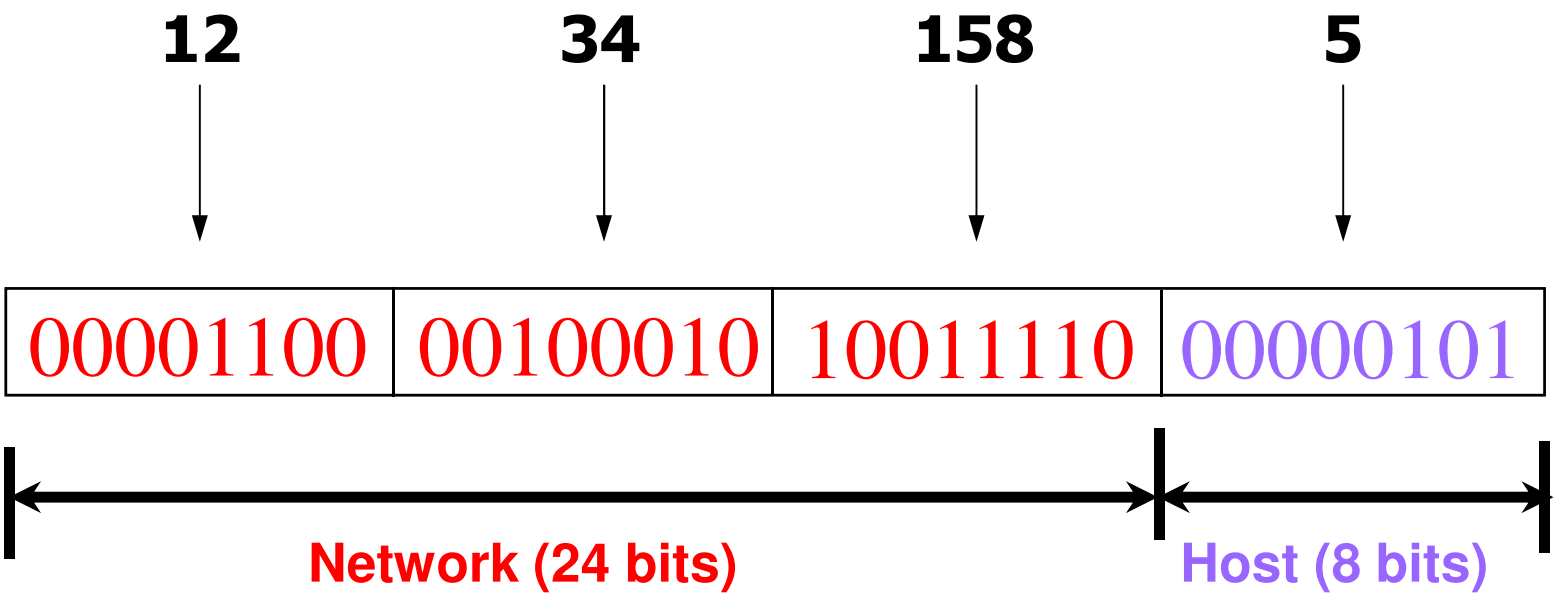


Esempio: **131.107.3.24**

Indirizzamento gerarchico: Prefissi IP

Dividere in sezione network & host (left e right)

– 12.34.158.0/24 è un prefisso a 24-bit con 2^8 indirizzi

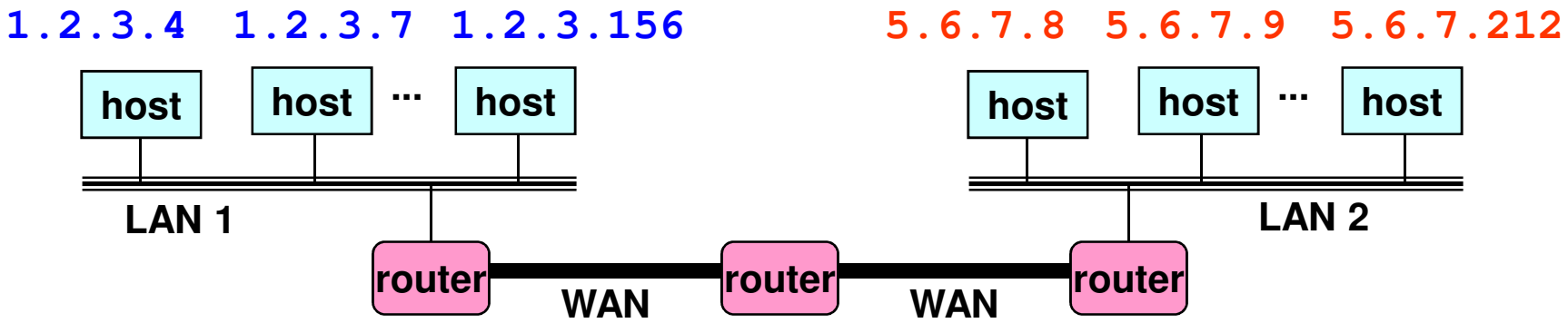


IP Address e 24-bit Subnet Mask

Address	**12**	**34**	**158**	**5**
	00001100	00100010	10011110	00000101
	11111111	11111111	11111111	00000000
Mask	**255**	**255**	**255**	**0**

La scalabilità migliora

- **Numero di hosts di una comune subnet**
  - 1.2.3.0/24 sulla LAN di sinistra
  - 5.6.7.0/24 sulla LAN di destra

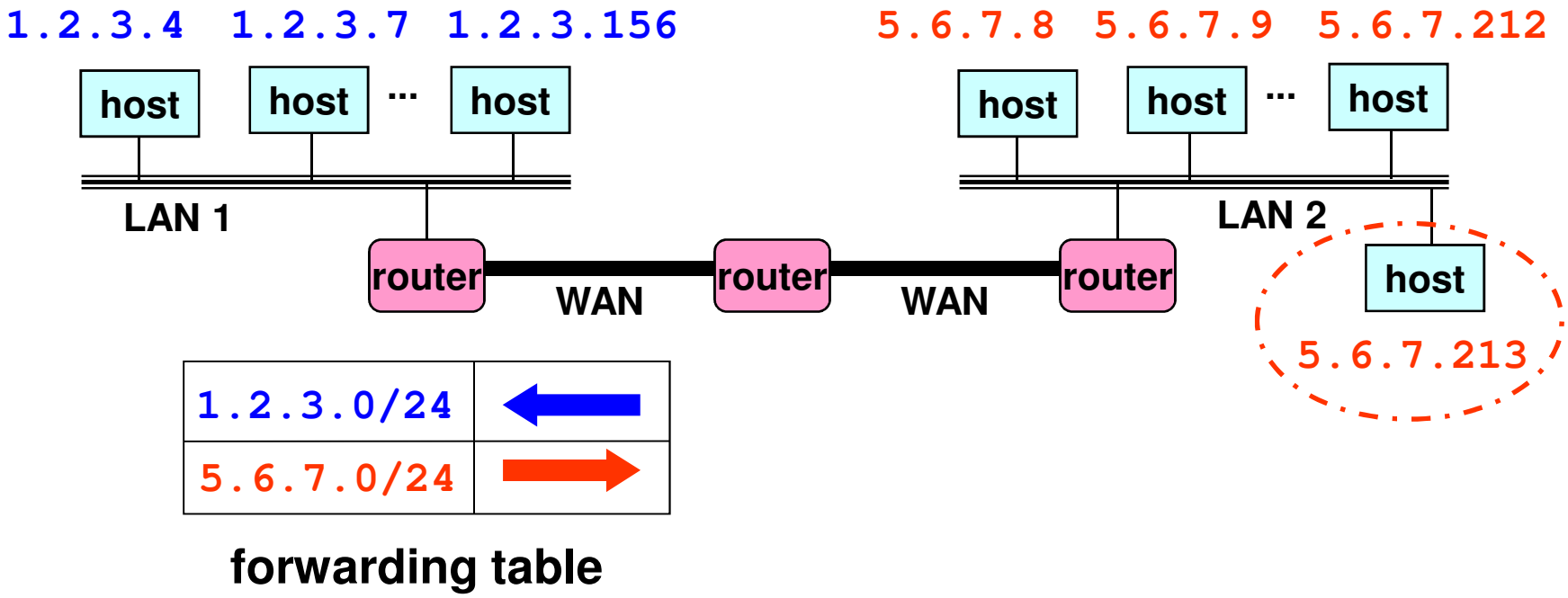


1.2.3.0/24	←
5.6.7.0/24	→

forwarding table

Facile aggiungere nuovi Hosts

- **Non c'è necessità di updatare i routers**
  - E.g., aggiungere un nuovo host 5.6.7.213 sulla destra
  - Non richiede un nuovo entry nella forwarding-table



Default Route

- **La default route in IPv4 (in CIDR) è 0.0.0.0/0, detta anche la quad-zero route.**
- **Poichè la subnet mask è /0, la default route indica ‘no network’, ed è lo "shortest" match possibile**
- **Il significato della ‘default route’ diventa:**
  - “quando non sai cos’altro fare, usa la default route”

Richiami sull'instradamento IP

- **Un host mittente determina se l'instradamento è diretto o indiretto ispezionando il net_id effettivo (tenendo conto della maschera di sottorete) dell'indirizzo IP di destinazione:**

  – **Corrispondenza** ➔ instradamento **diretto**
  L'host può inviare il pacchetto direttamente senza passare attraverso router. L'host deve "risolvere" l'indirizzo IP in un indirizzo Ethernet con l'Address Resolution Protocol (ARP).

  – **Non corrispondenza** ➔ instradamento **indiretto**
  L'host usa il router più vicino (**default gateway**).

Instradamento table-driven

- L'instradamento IP si basa su tabelle presenti su host e router.

Le tabelle d'instradamento elencano, per ciascuna sottorete nota, il relativo net_id e l'indirizzo IP del router d'inoltro.

Esse sono costituite da 4 colonne:

- Net_id della rete di destinazione, da confrontare con quella del pacchetto da inoltrare, tenendo conto della maschera.

  Attenzione! 0.0.0.0 indica il net_id di default. L' immissione di default della tabella viene usata per instradare i pacchetti il cui net_id di destinazione non compare esplicitamente in nessuna riga della tabella stessa.
- Maschera di sottorete associata al net_id.
- Indirizzo IP del gateway d'inoltro, a cui mandare il pacchetto se il suo net_id di destinazione coincide con la coppia net_id/maschera.
- Interfaccia di livello 2 usata per l'inoltro (es. eth=ethernet).

Instradamento table driven

- **Le tabelle di instradamento sono presenti anche sugli host.**
- **Nella tabella di un host si possono trovare tre tipi di relazioni, a volte dette rotte:**
  - rotte dirette;
  - rotte indirette, verso reti raggiungibili tramite uno o più router;
  - una rotta di default, che contiene l'indirizzo del router da usare per raggiungere tutti gli altri host.
- **Esempio di visualizzazione delle rotte su un host:**

```
[root@pluto root]# netstat -rn          (oppure route -n)
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
130.22.37.0     0.0.0.0         255.255.255.0   U        40    0        0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U        40    0        0 lo
0.0.0.0         130.22.37.1     0.0.0.0         UG       40    0        0 eth0
```

- **U: la rotta è UP**
- **G: la rotta è verso un gateway, altrimenti è diretta**
- **H:la rotta è verso un host, altrimenti è verso una rete**

Longest Prefix Match

- **Nella routing table, la route che ha il maggior numero di bit in comune con l'indirizzo di destinazione, a partire dalla sinistra, è detto best match o longest prefix match**
- **Questa route è quella prescelta dal router per instradare il pacchetto.**

IP Packet Destination	172.16.0.10	10101100.00010000.00000000.00001010
Route 1	172.16.0.0/12	10101100.00010000.00000000.00000000
Route 2	172.16.0.0/18	10101100.00010000.00000000.00000000
Route 3	172.16.0.0/26	10101100.00010000.00000000.00000000

Longest Match to IP Packet Destination

Allocazione degli Indirizzi

Luigi Vetrano

TechnoLabs

Classful Addressing

- **Agli inizi era prevista una allocazione fissa**
  - Classe A: 0*
    - Blocchi enormi /8 (e.g. MIT ha 18.0.0.0/8)
  - Classe B: 10*
    - Blocchi Larghi /16 (e.g. Princeton ha 128.112.0.0/16)
  - Classe C: 110*
    - Blocchi Piccoli /24 (e.g. AT&T Labs ha 192.20.225.0/24)
  - Classe D: 1110*
    - Multicast
  - Classe E: 11110*
    - Reserved (uso futuro)

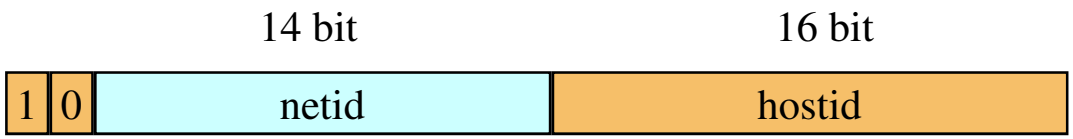
Indirizzi IP: classful addressing

- **Ogni host ha un indirizzo diviso in due parti IP_Address=Net_ID.Host_ID**
  - Host_ID identifica l'host all'interno della sottorete
  - Net_ID identifica la sotto-rete su Internet
- **Sono state inizialmente definite 5 classi di indirizzi (classful addressing):**

Classe A **(0.x.x.x → 127.x.x.x)**

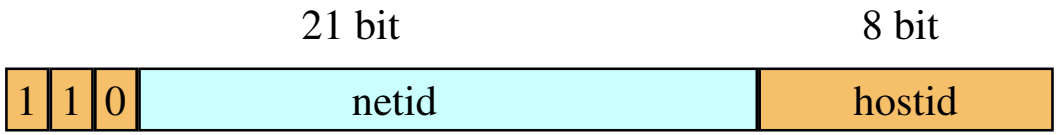
127.0.0.0 riservato

Classe B **(128.x.x.x → 191.x.x.x)**

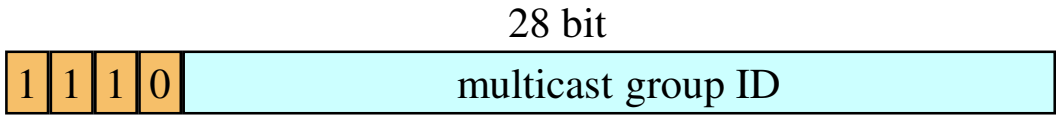


Indirizzi IP: classful addressing

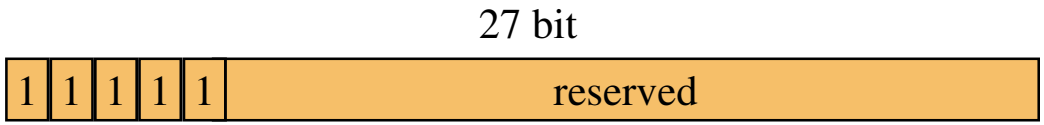
Classe C **(192.x.x.x → 223.x.x.x)**



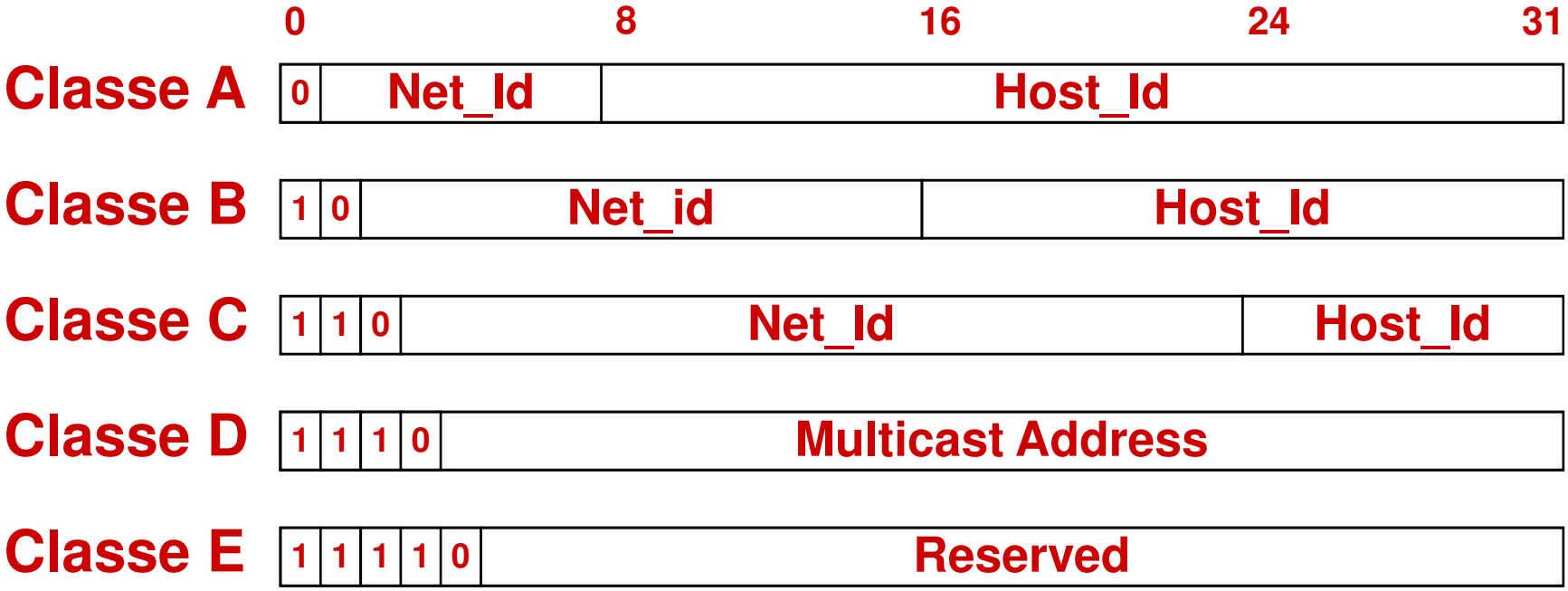
Classe D (per multicast) **(224.x.x.x → 239.x.x.x)**



Classe E (per sperimentazione) **(240.x.x.x → 255.x.x.x)**



Riepilogo delle Classi IP

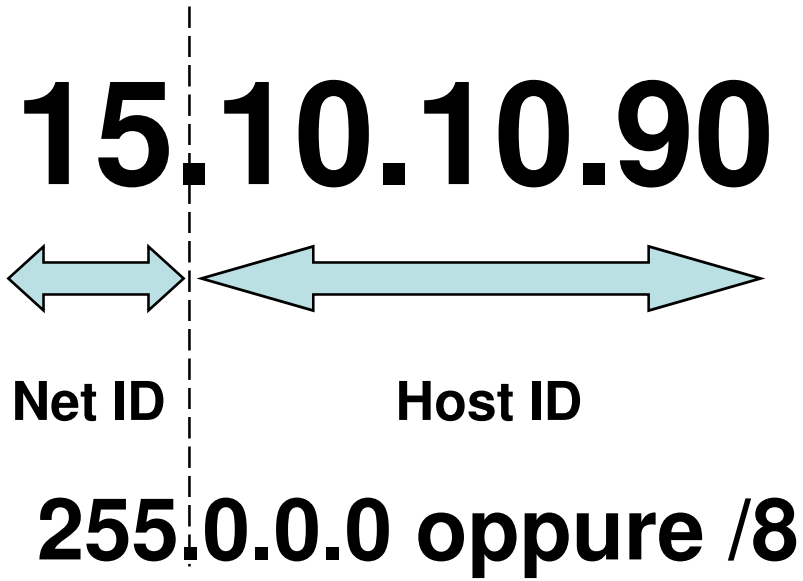


Riepilogo delle Classi IP

Classe	Leading bits	Inizio intervallo	Fine intervallo
Class A	0	0.0.0.0	127.255.255.255
Class B	10	128.0.0.0	191.255.255.255
Class C	110	192.0.0.0	223.255.255.255
Class D (multicast)	1110	224.0.0.0	239.255.255.255
Class E	1111	240.0.0.0	255.255.255.255

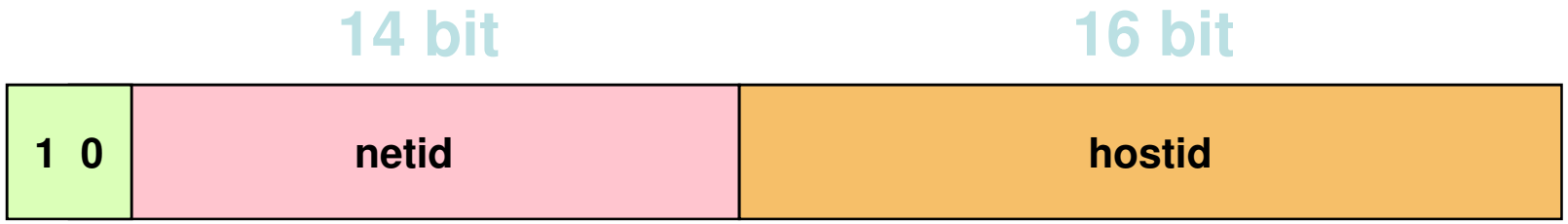
Indirizzi di classe A

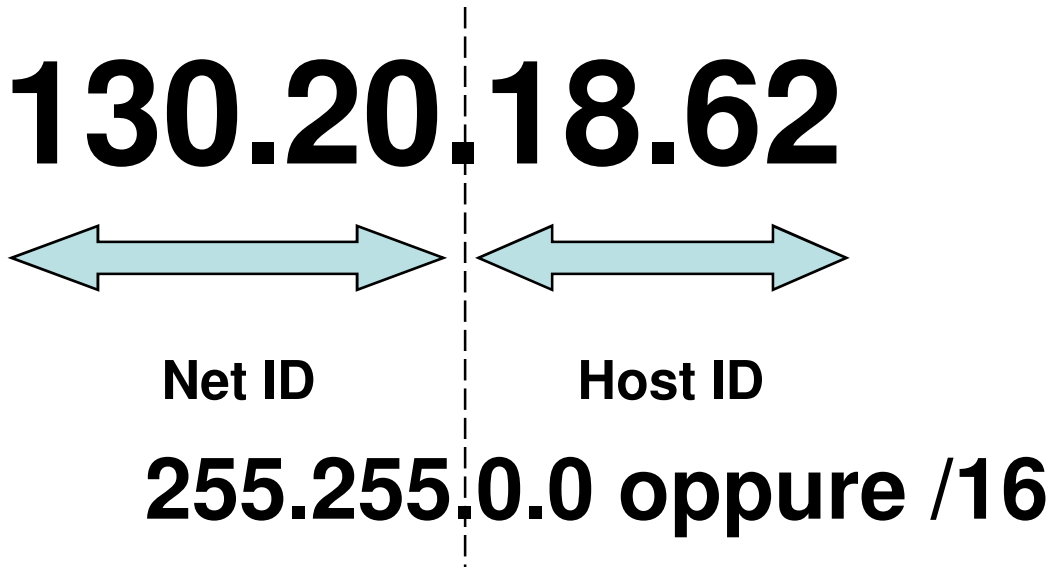
- **Esempio di indirizzo di classe A:**



Indirizzi di classe B

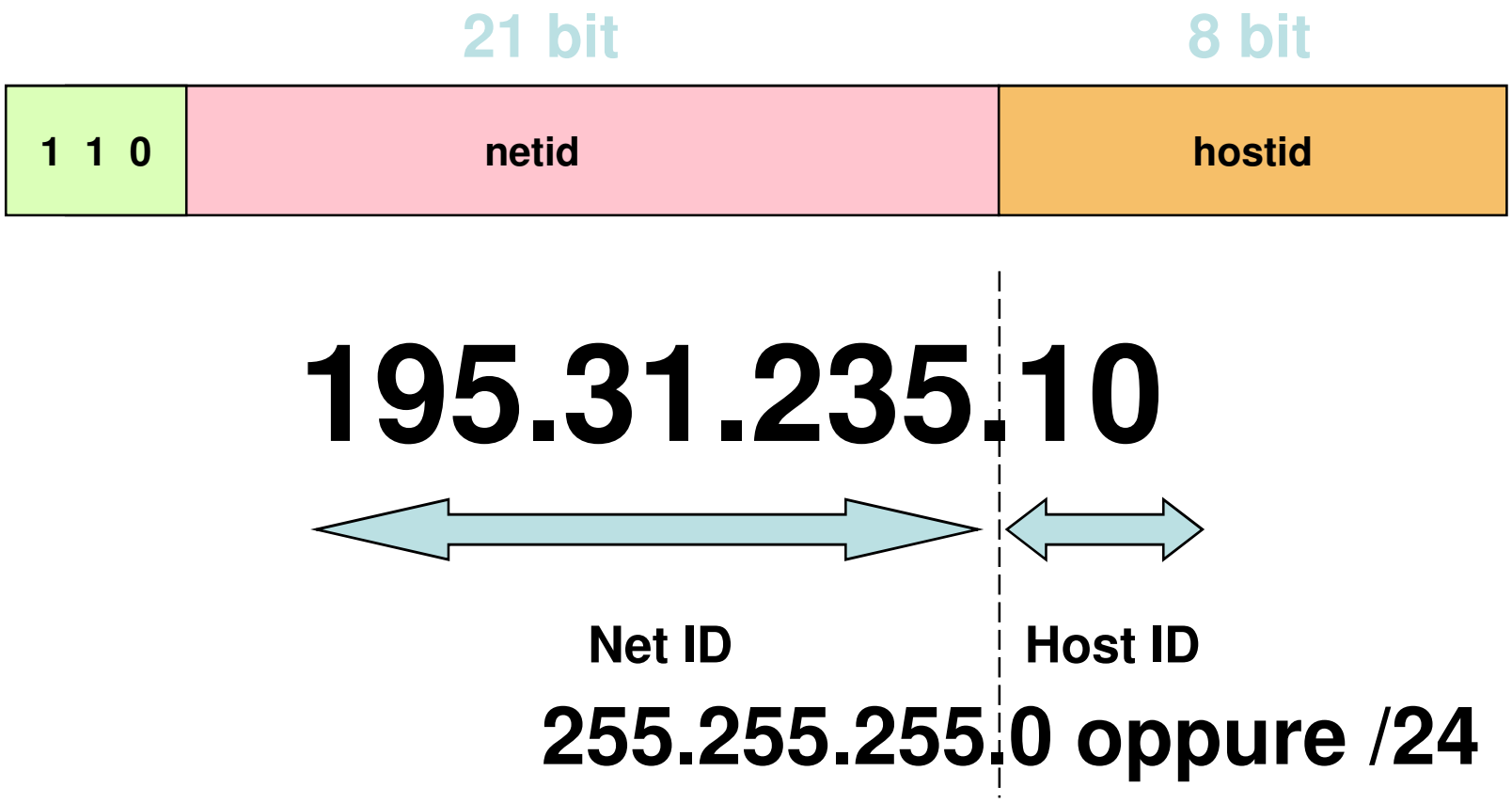
- **Esempio di indirizzo di classe B:**





Indirizzi di classe C

- **Esempio di indirizzo di classe C:**



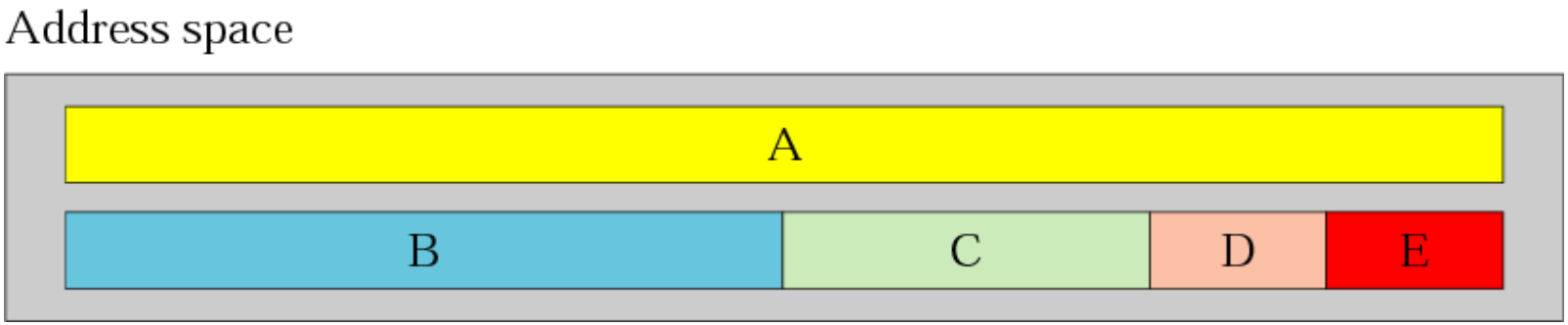
Indirizzi riservati per usi speciali (RFC 3330)

Indirizzi	CIDR	Funzione	RFC	CL	# indirizzi
0.0.0.0 - 0.255.255.255	0.0.0.0/8	Indirizzi zero	1700	A	16.777.216
10.0.0.0 - 10.255.255.255	10.0.0.0/8	IP privati	1918	A	16.777.216
127.0.0.0 - 127.255.255.255	127.0.0.0/8	Localhost Loopback	1700	A	16.777.216
169.254.0.0 - 169.254.255.255	169.254.0.0/16	Zeroconf	3330	B	65.536
172.16.0.0 - 172.31.255.255	172.16.0.0/12	IP privati	1918	B	1.048.576
192.0.2.0 - 192.0.2.255	192.0.2.0/24	Documentation and Examples	3330	C	256
192.88.99.0 - 192.88.99.255	192.88.99.0/24	IPv6 to IPv4 relay Anycast	3068	C	256
192.168.0.0 - 192.168.255.255	192.168.0.0/16	IP privati	1918	C	65.536
198.18.0.0 - 198.19.255.255	198.18.0.0/15	Network Device Benchmark	2544	C	131.072
224.0.0.0 - 239.255.255.255	224.0.0.0/4	Multicast	3171	D	268.435.456
240.0.0.0 - 255.255.255.255	240.0.0.0/4	Riservato	1700	E	268.435.456

Spazio di indirizzamento

Indirizzi di Classe A coprono la metà dell'intero spazio di indirizzamento !!

Milioni di indirizzi di classe A sono sprecati* !



*** Si tenga presente che le seguenti reti di classe A sono riservate:**

1. **0.0.0.0**
2. **10.0.0.0**
3. **127.0.0.0**

Percentuale di indirizzi per classe

Class	*Number of Addresses*	*Percentage*
A	$2^{31} = 2{,}147{,}483{,}648$	50%
B	$2^{30} = 1{,}073{,}741{,}824$	25%
C	$2^{29} = 536{,}870{,}912$	12.5%
D	$2^{28} = 268{,}435{,}456$	6.25%
E	$2^{28} = 268{,}435{,}456$	6.25%

Indirizzi Particolari

a) **127.x.y.z (tipicamente 127.0.0.1) → loopback (localhost)**

b) **Net_ID.(tutti 1 nel campo Host_ID) → broadcast sulla rete Net_ID**

c) **Net_ID.(tutti 0 nel campo Host_ID) → sottorete indicata da Net_ID**

d) **255.255.255.255 (tutti 1) → broadcast locale**

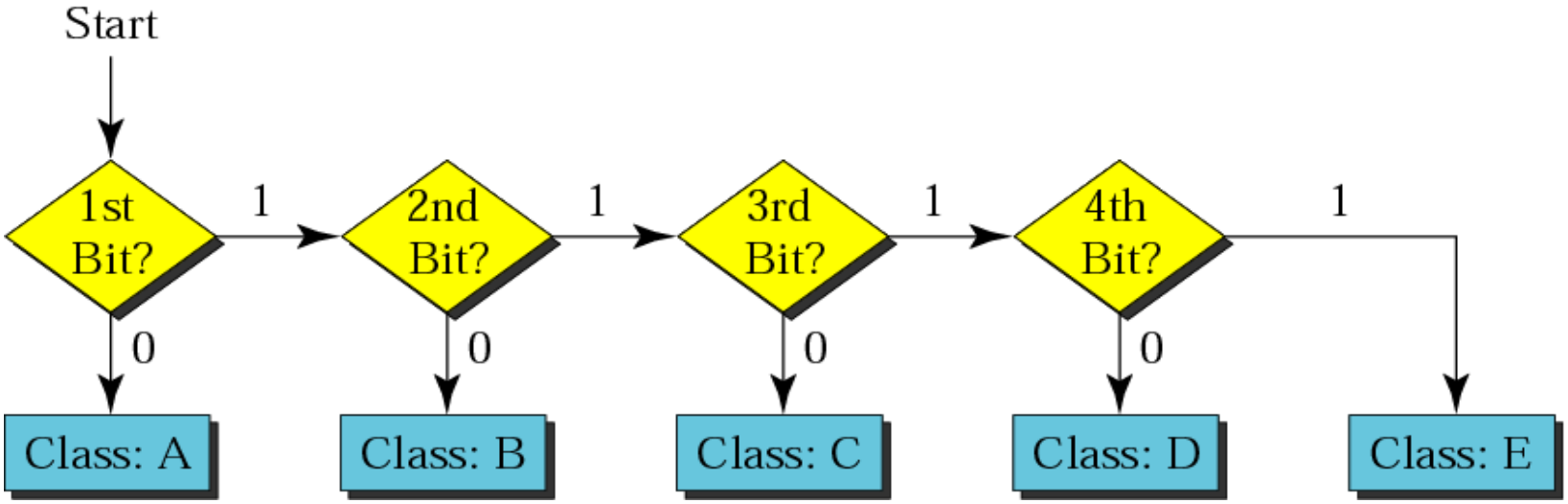
Nota: all'interno di una sottorete non è possibile assegnare a host o router gli indirizzi b) e c) → il numero di indirizzi assegnabili è pari a

$$2^{(\#\text{bit di Host_ID})} - 2$$

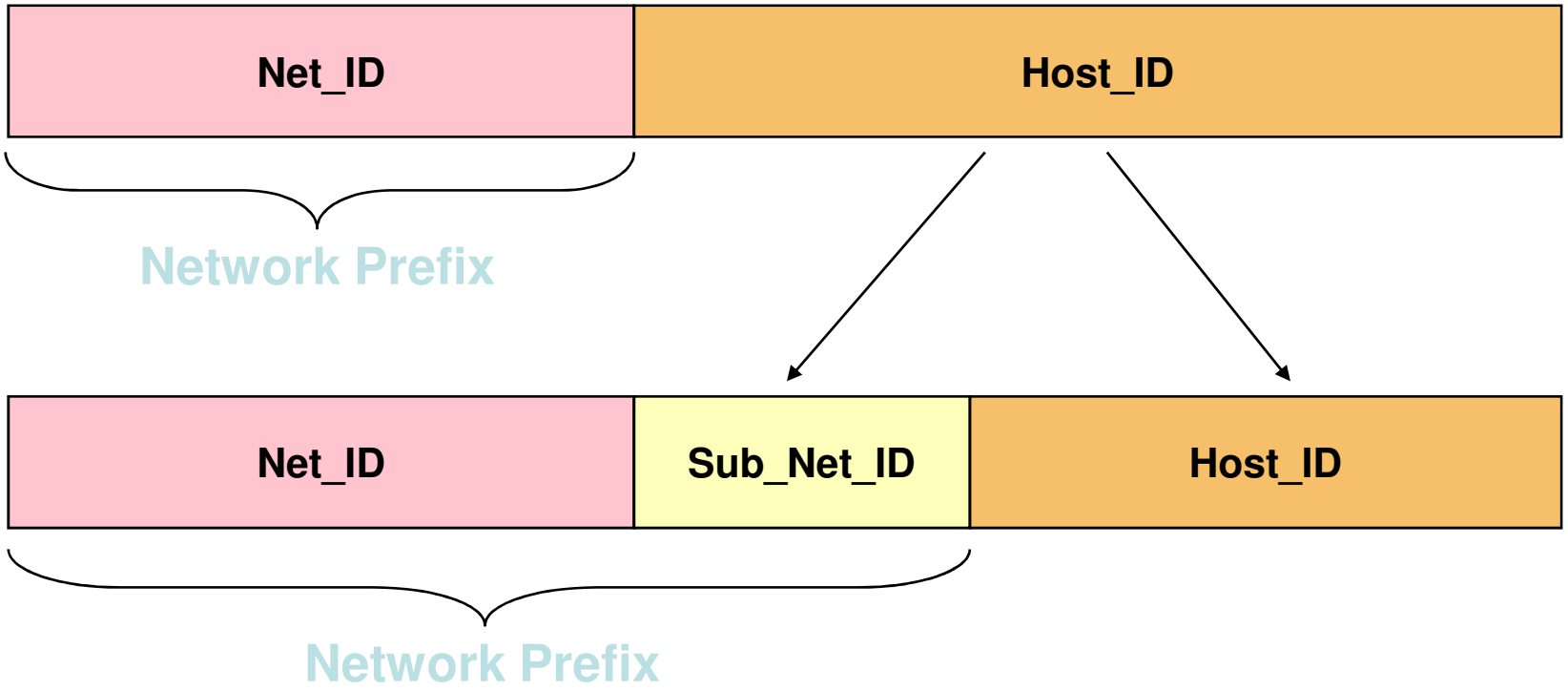
IANA-Allocated, Non-Internet Routable, IP Address Schemes

Class	Network Address Range
A	10.0.0.0 → 10.255.255.255
B	172.16.0.0 → 172.31.255.255
C	192.168.0.0 → 192.168.255.255

Trovare la classe di un indirizzo

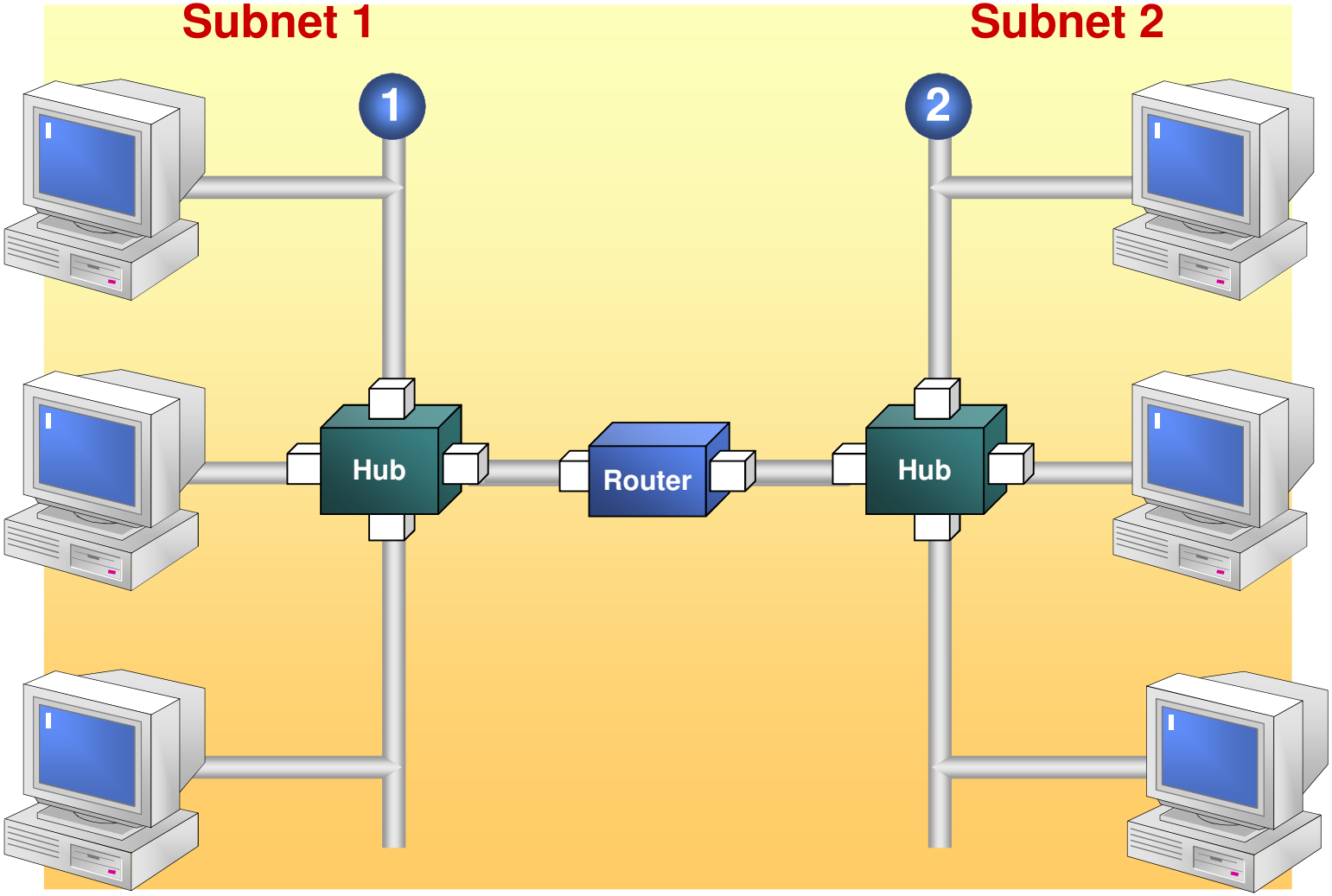


Subnetting



- **Dato un certo indirizzo di rete, la dimensione del Sub_Net_ID può essere:**
  - Fissa (subnet con ugual numero di host) → **subnetting con maschera fissa**
  - Variabile (subnet con diverso numero di host) → **subnetting con maschera variabile**

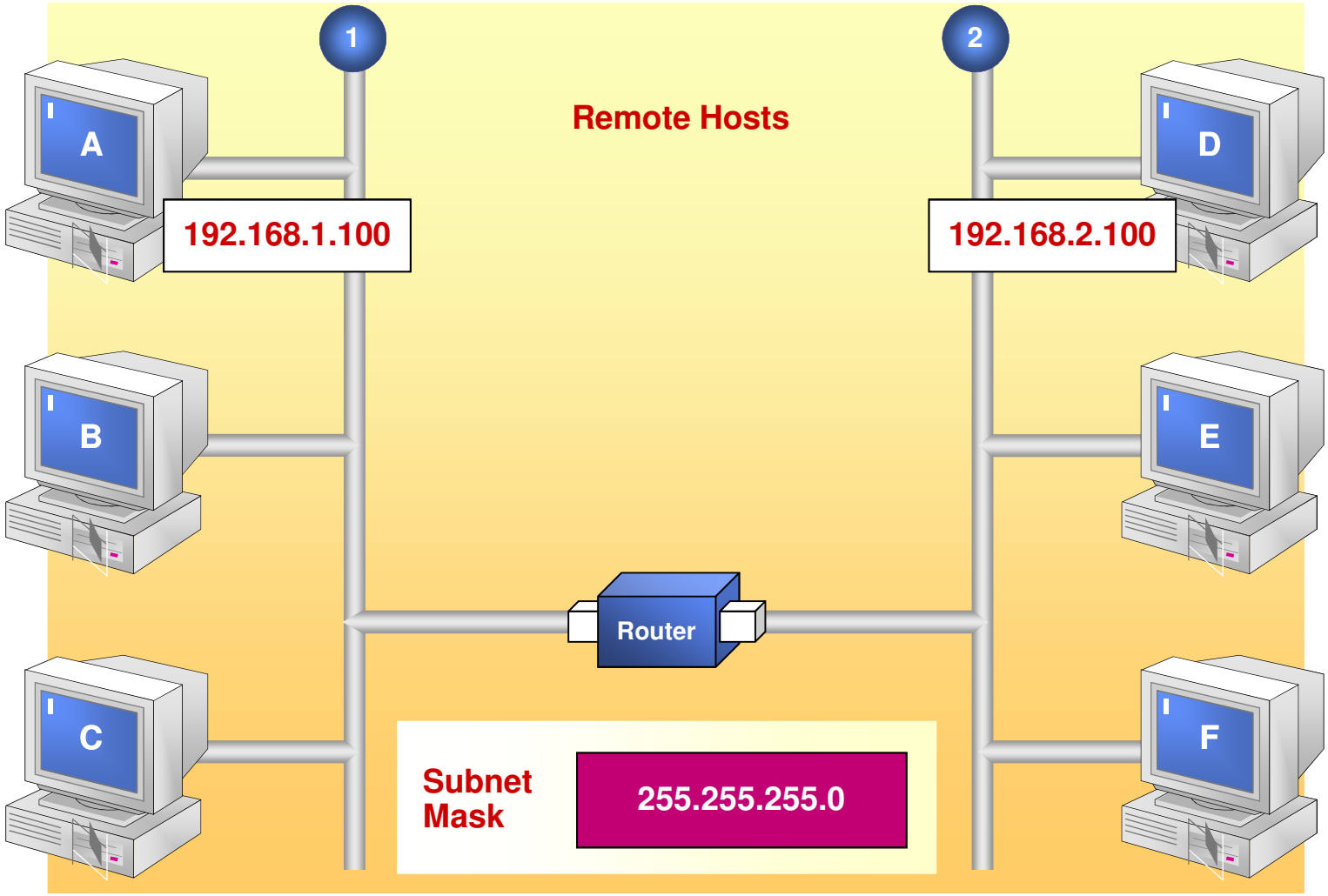
Subnets



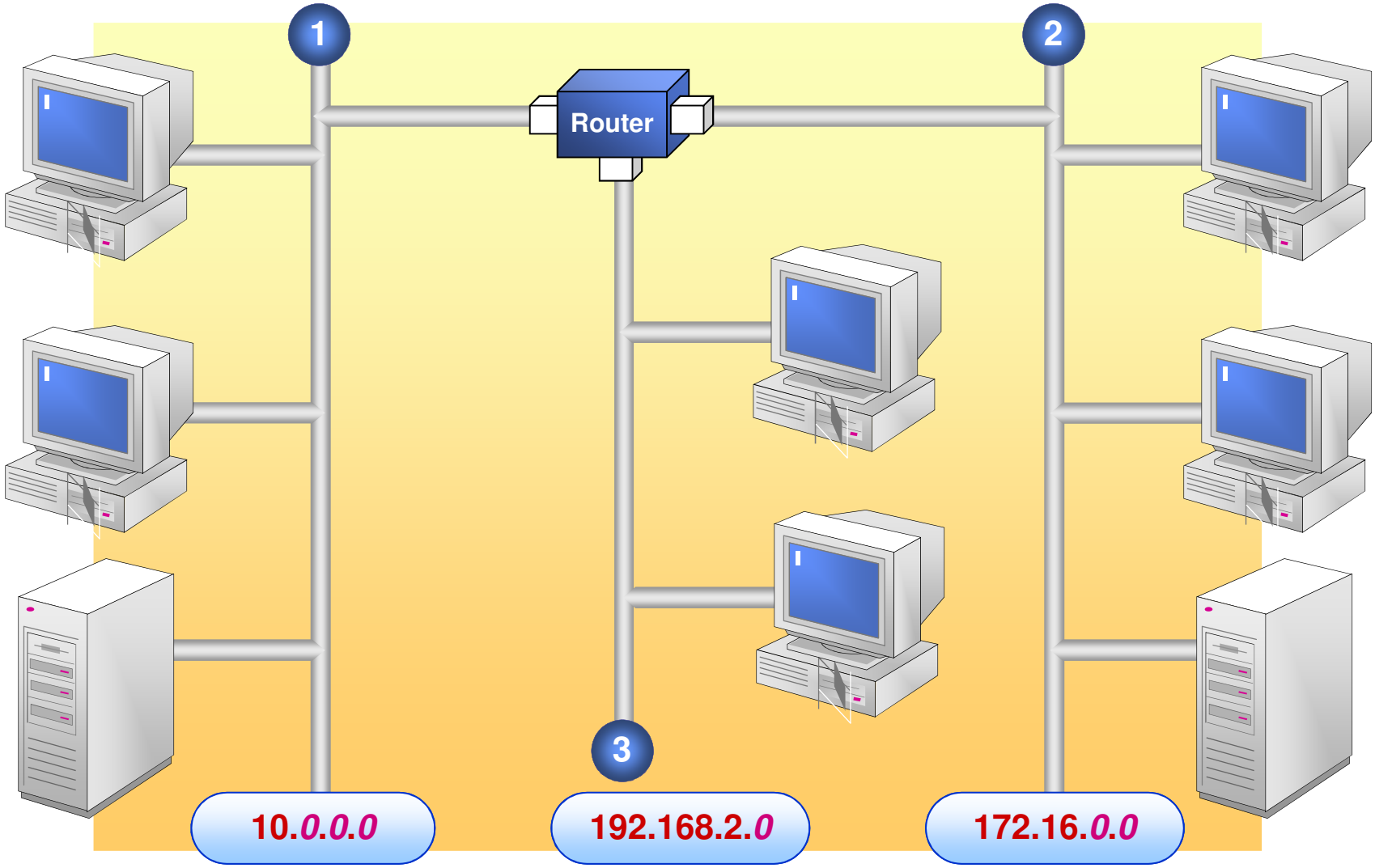
Subnet Masks

IP Address	10.50.100.	200
Subnet Mask	255.255.255.	0
Network ID	10.50.100.	0

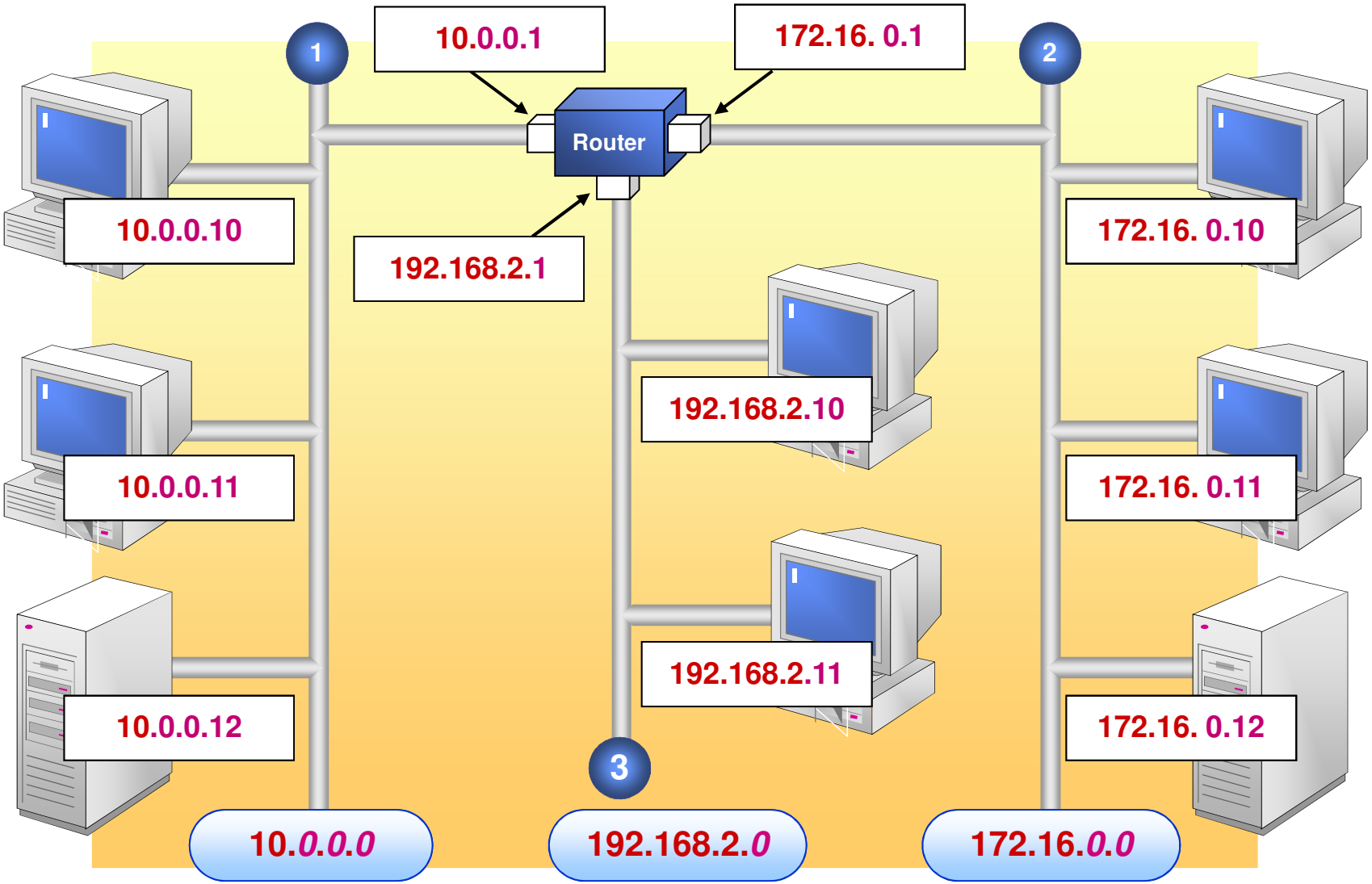
Local and Remote Hosts



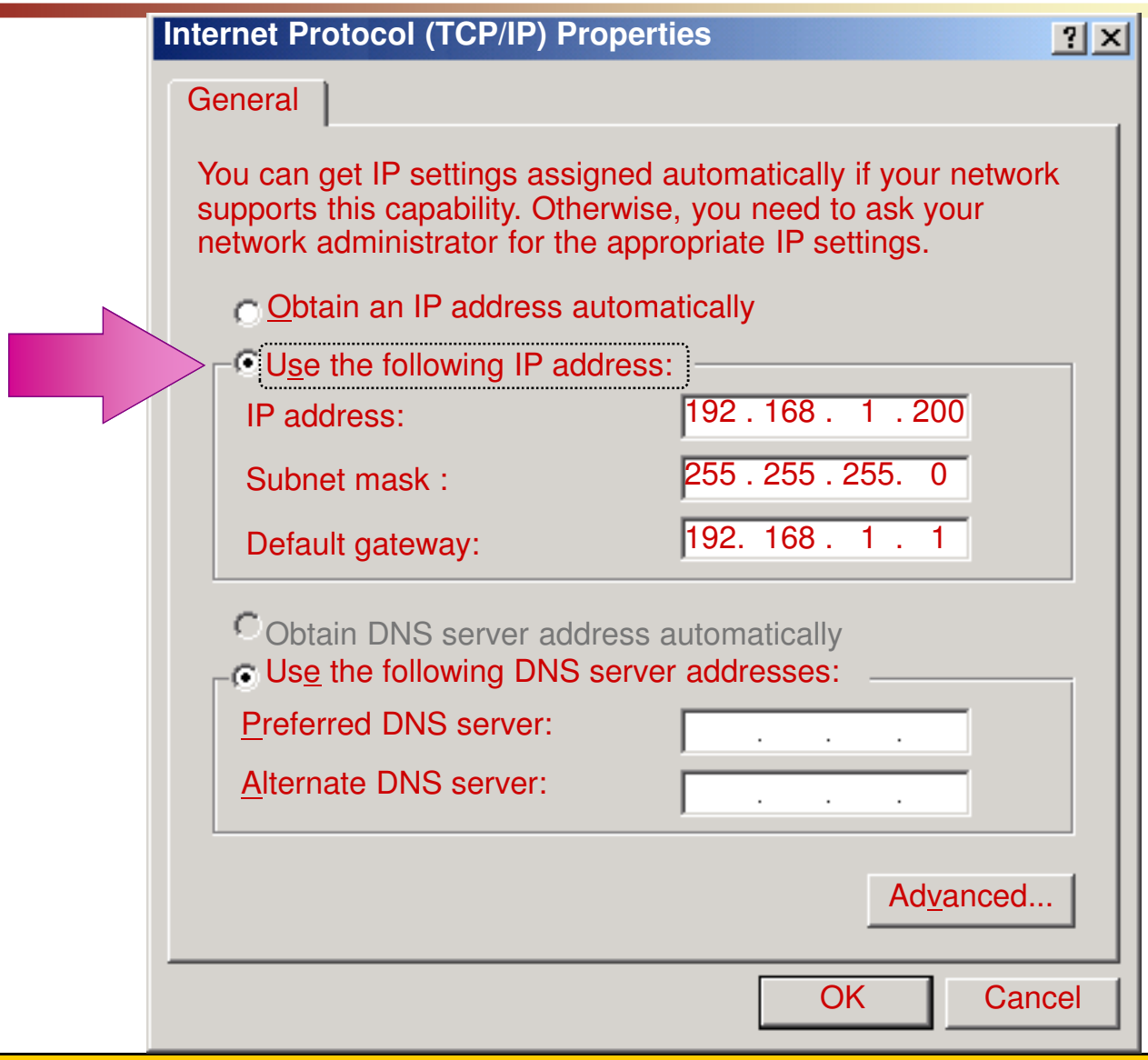
Assigning Network IDs



Assigning Host IDs



Static IP Addressing

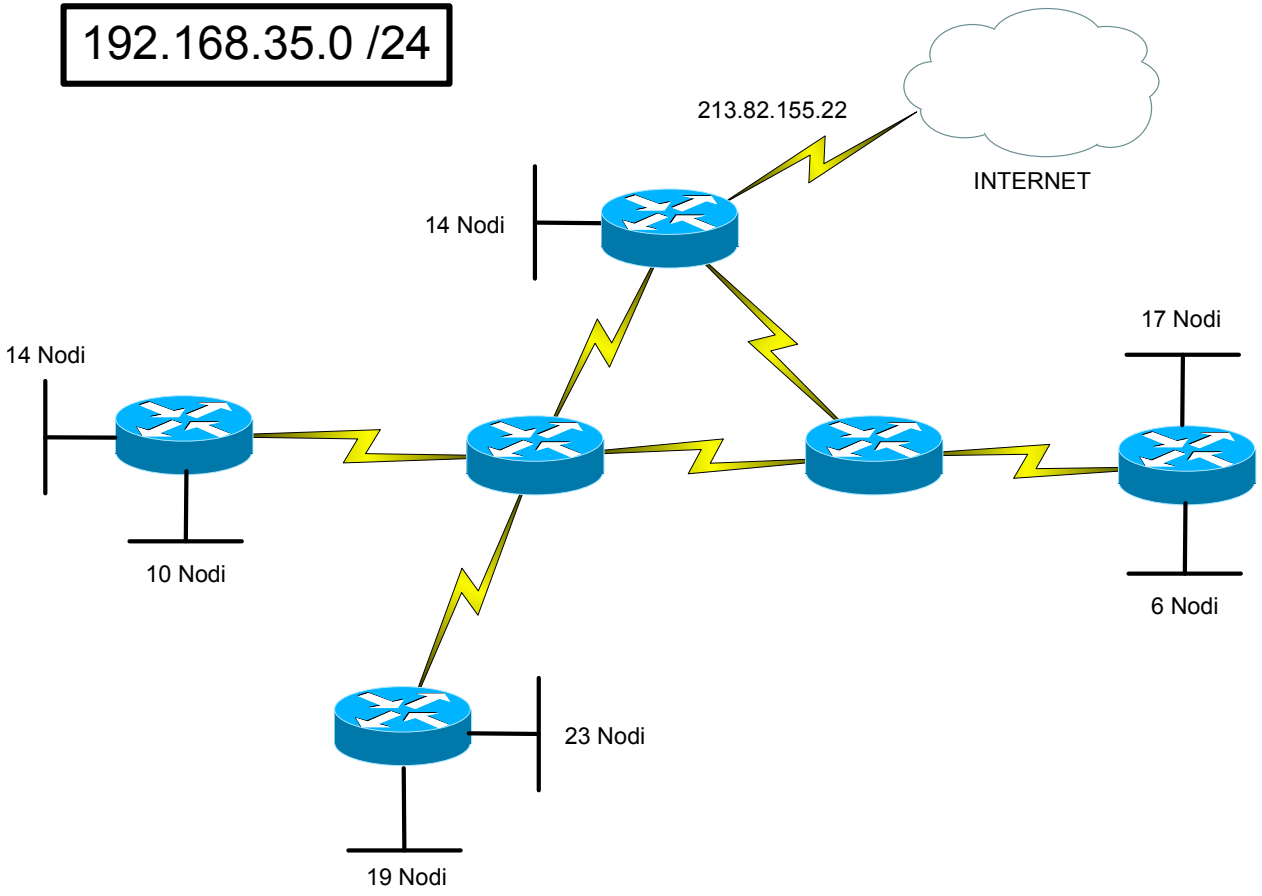


Viewing TCP/IP Configuration Using Ipconfig

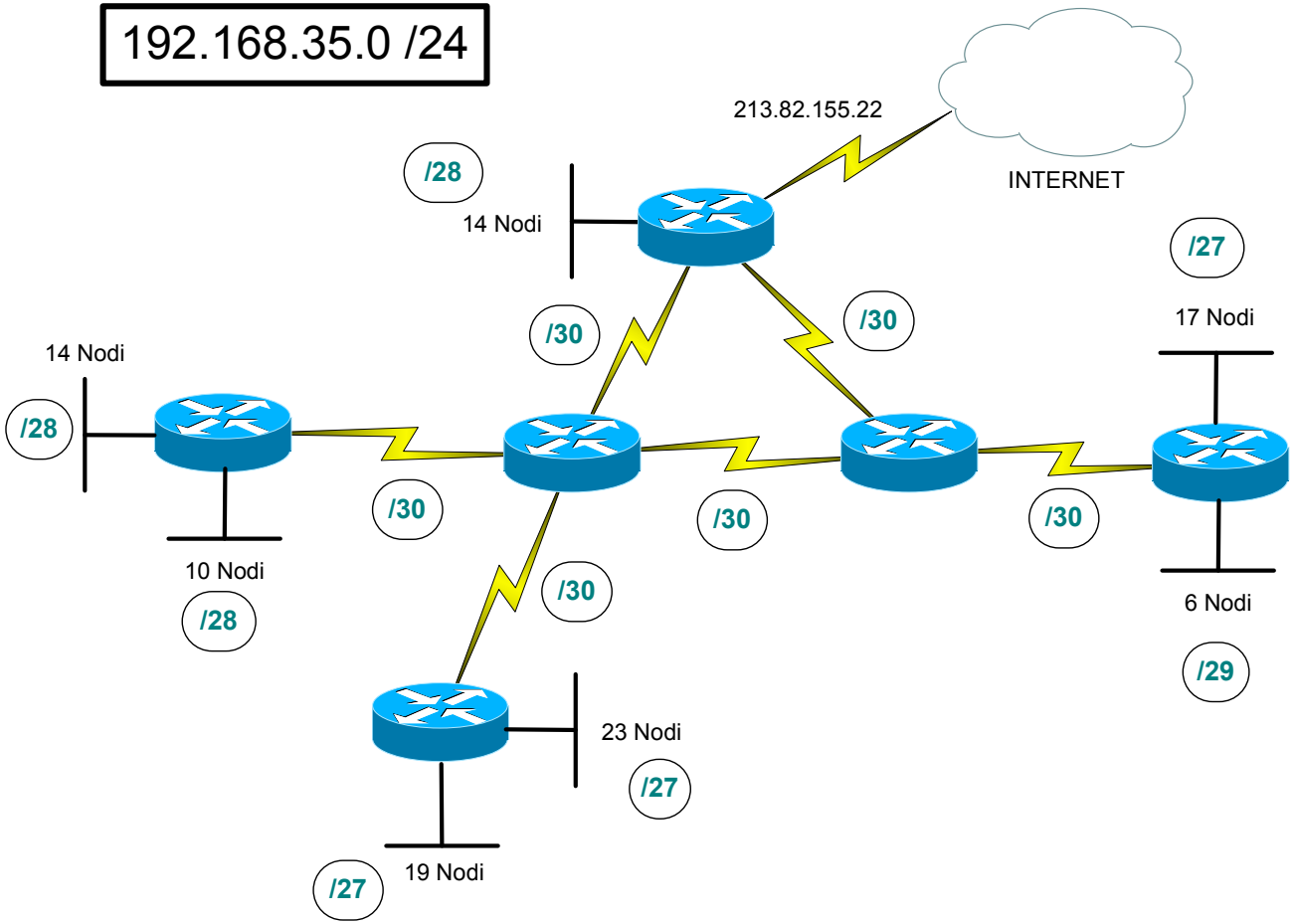
Command Prompt

```
Microsoft Windows 2000 [version 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.

C:\>ipconfig

Windows 2000 IP Configuration

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 192.168.1.200
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.1.1

C:\>_
```

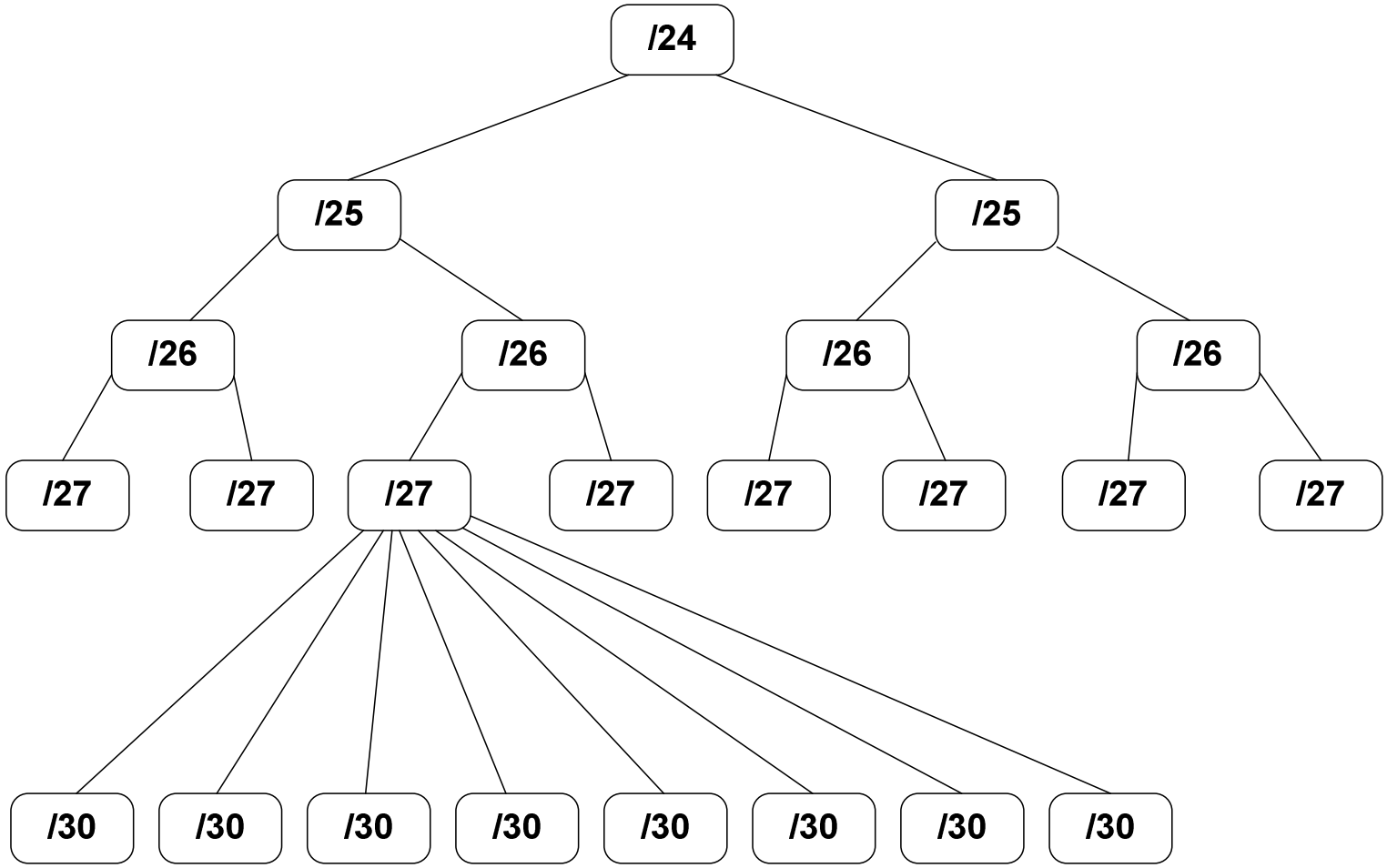
Esempio Pratico



Esempio Pratico



Subnetting Options (cont.)



Subnetting Examples

SubnetMask	SubnetMask	# Hosts
255.255.255.0	/24	256 (254)
255.255.255.128	/25	128 (126)
255.255.255.192	/26	64 (62)
255.255.255.224	/27	32 (30)
255.255.255.240	/28	16 (14)
255.255.255.248	/29	8 (6)
255.255.255.252	/30	4 (2)

Classless Inter-Domain Routing (CIDR)

Usare due numeri a 32-bit per rappresentare la rete.
Network number = IP address + Mask

IP Address : 12.4.0.0 IP Mask: 255.254.0.0

Address	00001100	00000100	00000000	00000000
Mask	11111111	11111110	00000000	00000000

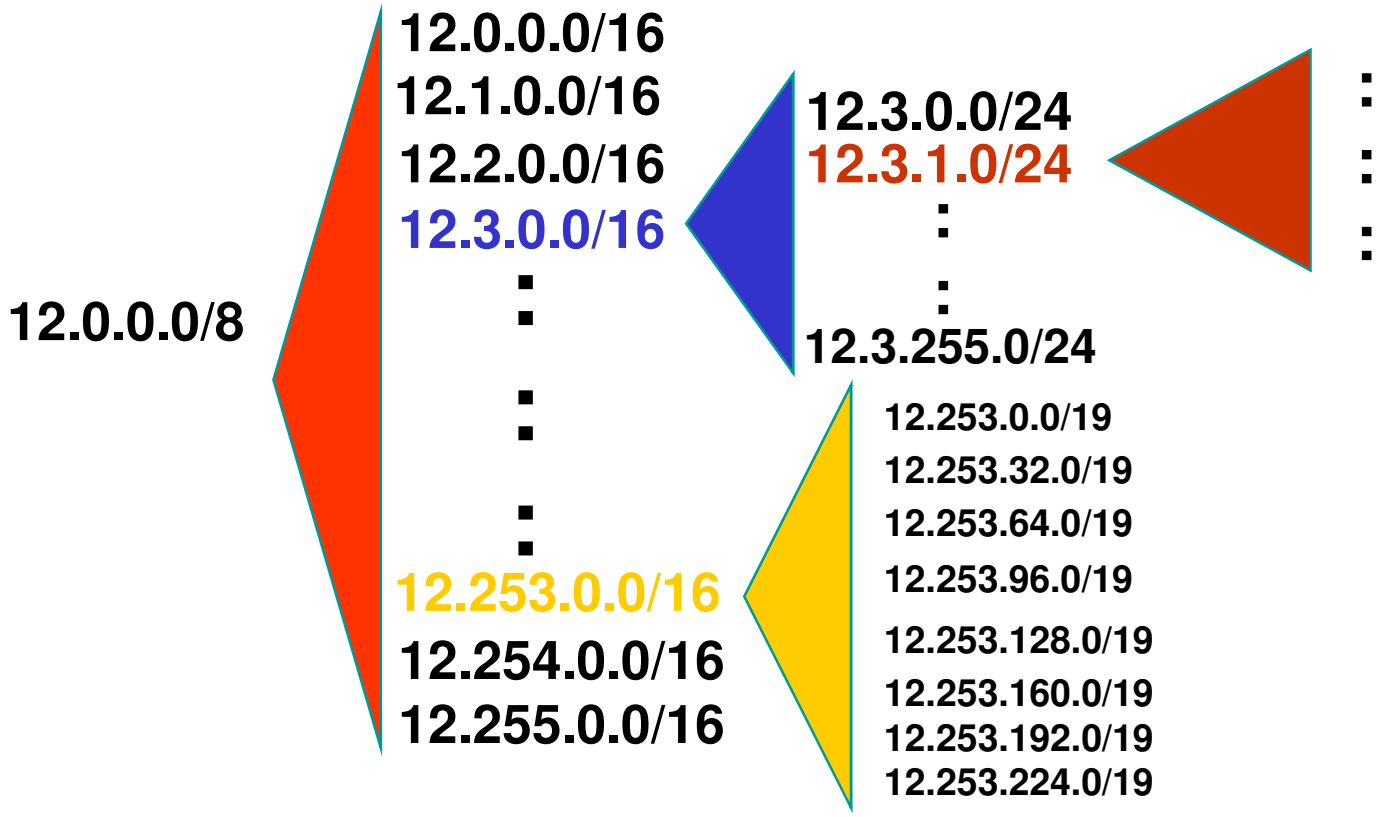
← **Network Prefix** → ← **hosts** →

Scritto come 12.4.0.0/15

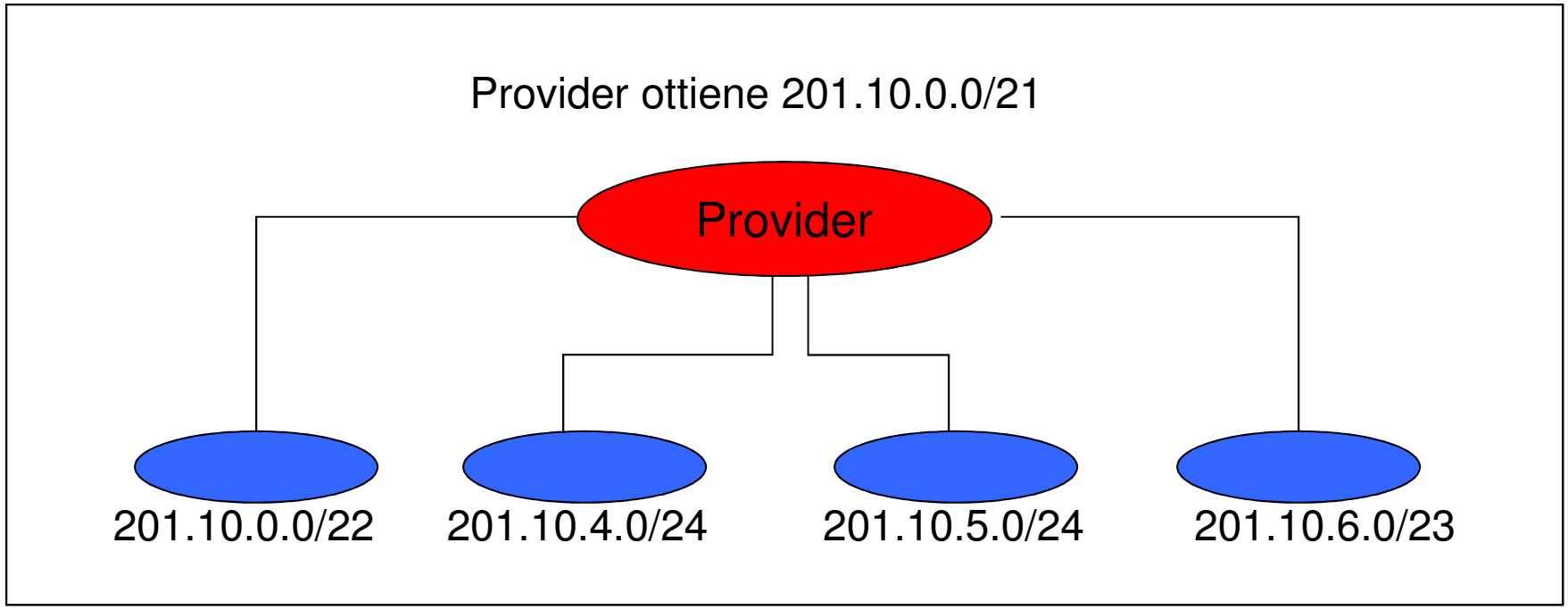
CIDR: Indirizzamento gerarchico

Prefissi sono la chiave della scalabilità

- Gli indirizzi sono allocati in blocchi contigui (chunks)
- I protocolli di routing e il packet forwarding sono basati sui prefissi
- Oggi, una routing table contiene ~150.000 ➔ 200.000 prefissi.



Scalabilità: Address Aggregation



Routers nel resto di Internet devono solo sapere come raggiungere 201.10.0.0/21. Il provider può dirigere i pacchetti IP al customer appropriato.

IP Address e 21-bit Subnet Mask

Address	**201**	**10**	**0**	**0**
	11001001	00001010	00000000	00000000
/21	11111111	11111111	11111000	00000000
Mask	**255**	**255**	**248**	**0**

Divisione dello spazio in due Subnet /22

Address	**201**	**10**	**0**	**0**
	11001001	00001010	00000000	00000000
/22	11111111	11111111	11111100	00000000
Mask	**255**	**255**	**252**	**0**

1

Divisione dello spazio in due Subnet /22

	201	10	0	0
S-1	11001001	00001010	00000000	00000000
	201	10	4	0
S-2	11001001	00001010	00000100	00000000
/22	11111111	11111111	11111100	00000000
Mask	255	255	252	0

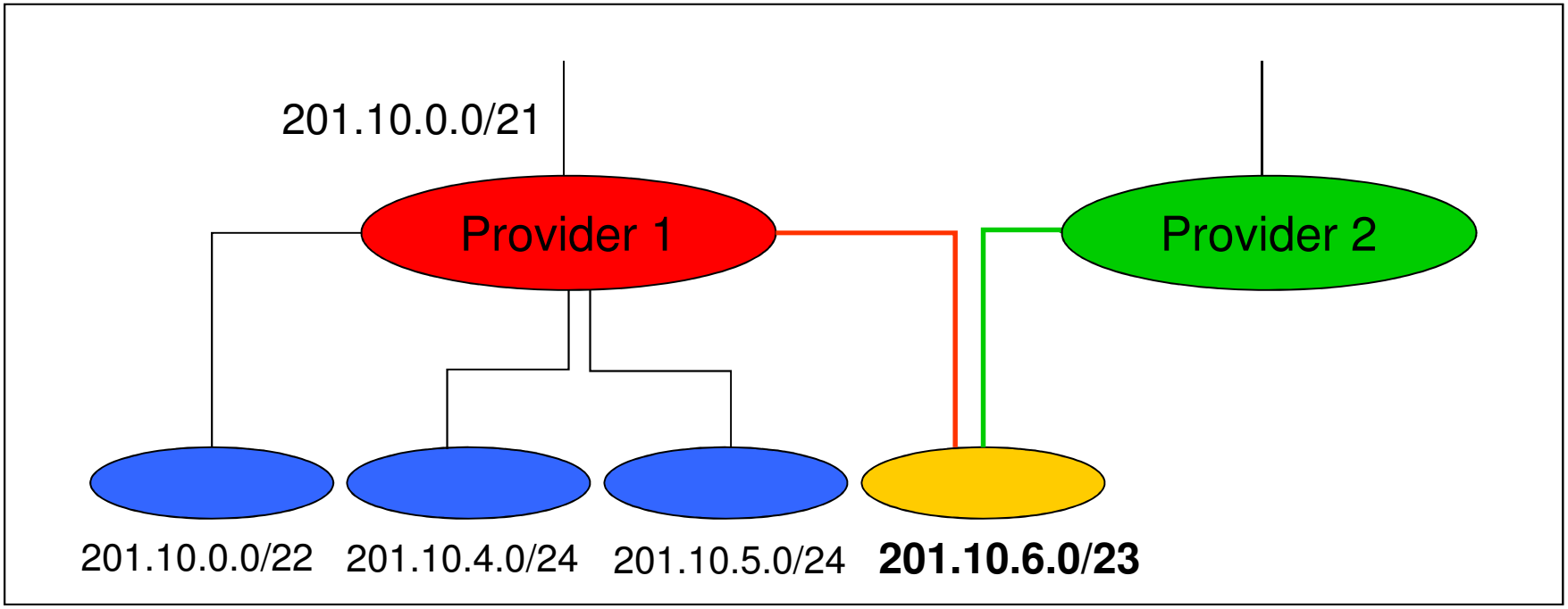
Divisione di S2 in due Subnet /23

	201	10	4	0
S-2-1	11001001	00001010	00000100	00000000
	201	10	6	0
S-2-2	11001001	00001010	00000110	00000000
/23	11111111	11111111	11111110	00000000
Mask	255	255	254	0

Divisione di S2-1 in due Subnet /24

	201	10	4	0
S-2-1-1	11001001	00001010	00000100	00000000
	201	10	5	0
S-2-1-2	11001001	00001010	00000101	00000000
/24	11111111	11111111	11111111	00000000
Mask	255	255	255	0

L'aggregazione non è sempre possibile



Multi-homed **customer con 201.10.6.0/23 ha due providers. Internet deve sapere come raggiungere la destinazione attraverso entrambi i providers.**

Subnet 192.168.1.65/24 25

C:\WINDOWS\system32\cmd.exe

```
D:\ity10250\Desktop\UnivAq>subnet.exe 192.168.1.65/24 25
Showing         : 192.168.1.65/24      11000000.10101000.00000001.01000001

Subnet Mask     : 255.255.255.0        11111111.11111111.11111111.00000000
Wildcard Mask   : 0.0.0.255            00000000.00000000.00000000.11111111
Host Count      : 254

Network         : 192.168.1.0          11000000.10101000.00000001.00000000
Minimum Host    : 192.168.1.1          11000000.10101000.00000001.00000001
Maximum Host    : 192.168.1.254        11000000.10101000.00000001.11111110
Broadcast       : 192.168.1.255        11000000.10101000.00000001.11111111

/25 Subnets     : 2
Subnet Mask     : 255.255.255.128      11111111.11111111.11111111.10000000
Wildcard Mask   : 0.0.0.127            00000000.00000000.00000000.01111111
Host Count      : 126                  (Per Subnet)

Network         : 192.168.1.0          11000000.10101000.00000001.00000000
Minimum Host    : 192.168.1.1          11000000.10101000.00000001.00000001
Maximum Host    : 192.168.1.126        11000000.10101000.00000001.01111110
Broadcast       : 192.168.1.127        11000000.10101000.00000001.01111111

Network         : 192.168.1.128        11000000.10101000.00000001.10000000
Minimum Host    : 192.168.1.129        11000000.10101000.00000001.10000001
Maximum Host    : 192.168.1.254        11000000.10101000.00000001.11111110
Broadcast       : 192.168.1.255        11000000.10101000.00000001.11111111
```

Ottenere un blocco di indirizzi

- **Separazione del controllo**
  - Prefisso: assegnato ad una istituzione
  - Indirizzi: assegnati da una istituzione ai suoi nodi
- **Chi assegna i prefissi ?**
  - Internet Corporation for Assigned Names and Numbers (ICANN)
    - Alloca larghi blocchi di indirizzi ai Regional Internet Registries
  - Regional Internet Registries (RIRs)
    - E.g., ARIN (American Registry for Internet Numbers)
    - Allocano blocchi di indirizzi all'interno delle loro regioni
    - Assegnati a Internet Service Providers e Istituzioni
  - Internet Service Providers (ISPs)
    - Allocano blocchi di indirizzi ai propri customers
    - Questi possono allocare gli indirizzi ai propri customers…

Chi possiede un Address

- **Address registries**
  - Public record dell'allocazione degli indirizzi
  - Internet Service Providers (ISPs) dovrebbero farne l'update quando assegnano indirizzi ai propri customers
  - Tuttavia, i records sono notoriamente out-of-date
- **Queries**
  - UNIX: "whois –h whois.arin.net univaq.it"
  - http://www.arin.net/whois/
  - http://www.geektools.com/whois.php
  - …

Whois Record

```
Domain:          univaq.it
Status:          ACTIVE
Created:         1996-01-29 00:00:00
Last Update:     2007-01-30 00:36:31
Expire Date:     2008-01-29

Registrant  Name:        Universita' degli Studi - L'Aquila
ContactID:               UNIV412-ITNIC
Address:                 Via Forcella 10
                         L'Aquila 67100  AQ    IT
Technical Contacts
Name:            ********************
ContactID:       MP1325-ITNIC
Address:         Universita' degli Studi dell'Aquila
                 Via vetoio snc
                 Coppito (67010) AQ   IT
Created:         1999-09-02 00:00:00
Last Update:     2007-03-01 07:48:00

Registrar Organization: Consortium GARR
Name:            GARR-MNT

Nameservers
ns.univaq.it
ns1.univaq.it
```

32-bit sono abbastanza ?

- **Non ci sono poi tantissimi indirizzi**
  - 2^{32} = 4,294,967,296
  - In più, diversi sono riservati per scopi speciali
  - e, gli indirizzi sono assegnati in blocchi
- **Diversi device hanno esigenza di un IP address**
  - Computers, PDAs, routers, tanks, toasters, …
- **Soluzione a lungo-termine: un address space più grande**
  - IPv6 ha 128-bit addresses ($2^{128} = 3.403 \times 10^{38}$)
  - Quantificando con un esempio, per ogni metro quadrato di superficie terrestre, ci sono 666.000.000.000.000.000.000.000 indirizzi IPv6 unici (cioè 666 mila miliardi di miliardi), ma solo 0,000007 IPv4 (cioè solo 7 IPv4 ogni km quadrato).
- **Soluzione a breve: arrangiarsi con IPv4**
  - Private addresses
  - Dynamically-assigned addresses (DHCP)
  - Network address translation (NAT)

DHCP: Dynamic Host Configuration Protocol

Protocollo Dinamico di Configurazione degli Host

Luigi Vetrano

TechnoLabs

Definizione e riferimenti

- **Dynamic Host Configuration Protocol: protocollo di assegnazione automatica degli host (client) di più settaggi di rete (IP, mask, DNS, router…)**

- **riferimenti :**
  - RFC 2131
  - RFC 3397
  - RFC 2132 - DHCP Options and BOOTP Vendor Extensions
  - http://www.networksorcery.com/enp/protocol/bootp/options.htm

Quando ? (Ambiti di utilizzo)

- Non sappiamo a priori quanti client ci siano in rete
- Si hanno portatili sulla rete
- Si vogliono fornire dinamicamente anche altre info (DNS/Gateway/lpr …)
- Si può dedicare una macchina a tale servizio

Perchè ?

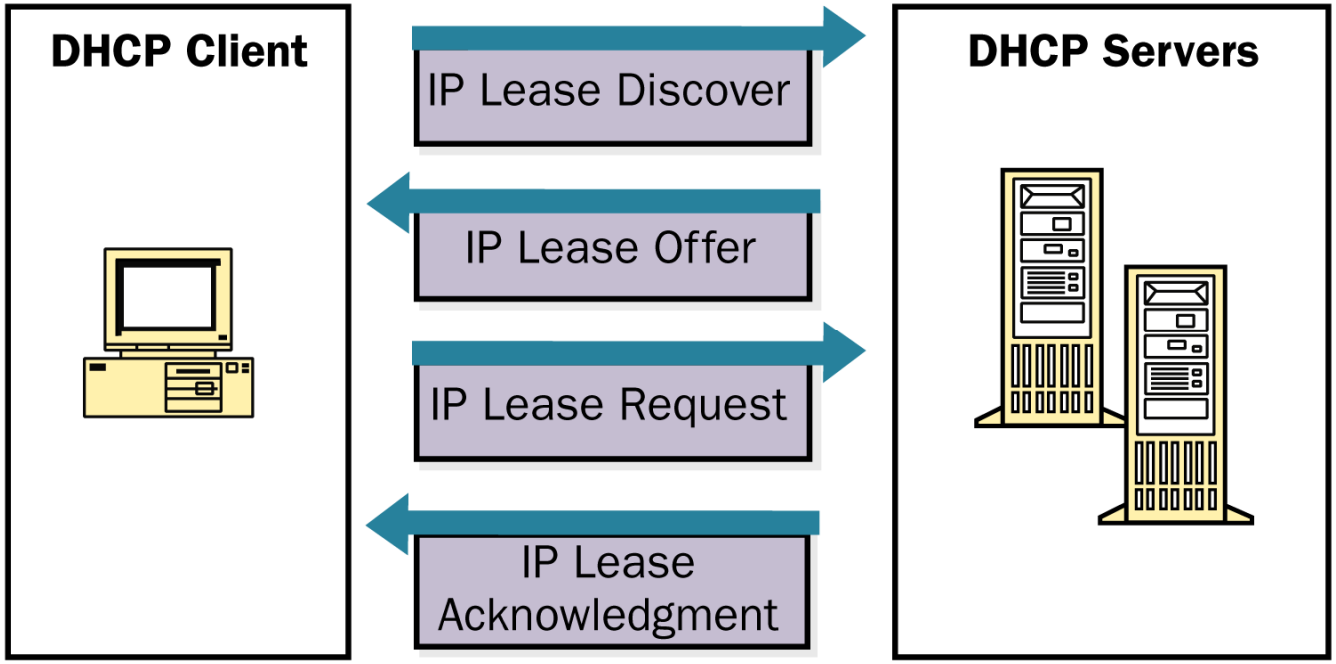
- Fornisce automaticamente settaggi altrimenti da implementare manualmente
- I client sono molti (> 100)
- Ogni nuovo client è subito operativo (non deve essere settato dall'Amministratore)
- Fornisce molti altri servizi (DNS, Gateway LPR, WINS…)

Come ?

- **Pianificazione ACCURATA del range e delle opzioni**
- **Scelta di un opportuno server (anche con poca processing power)**
- **Assegnazione di un IP FISSO al server DHCP**
- **Unix: dhcpd sviluppato da ISC (Internet Software Consortium - http://www.isc.org).**

  La configurazione del demone dhcpd è contenuta in un file di testo: /etc/dhcpd.conf.

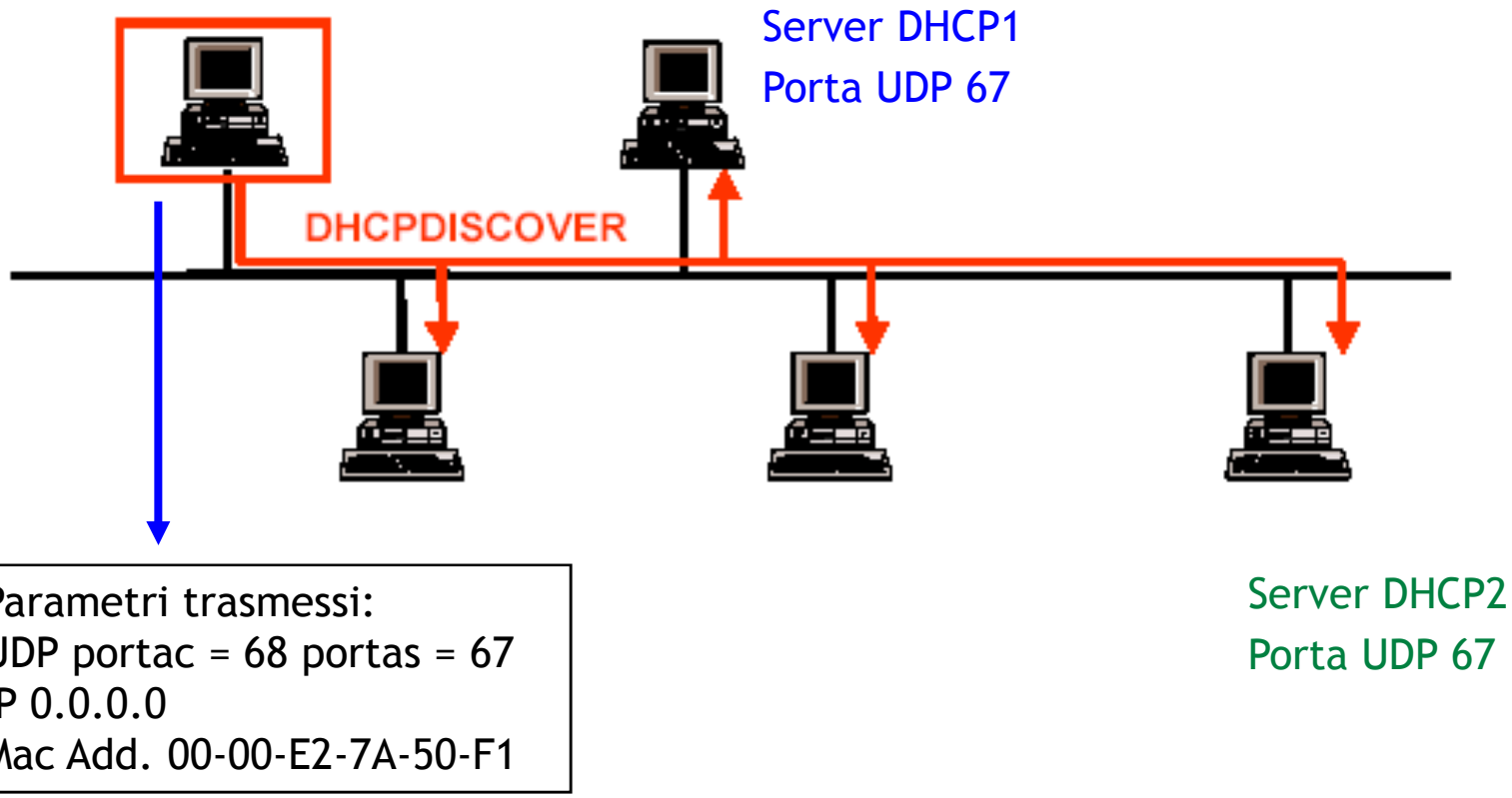
IP Lease Discover & Offer



Principio di funzionamento (1)

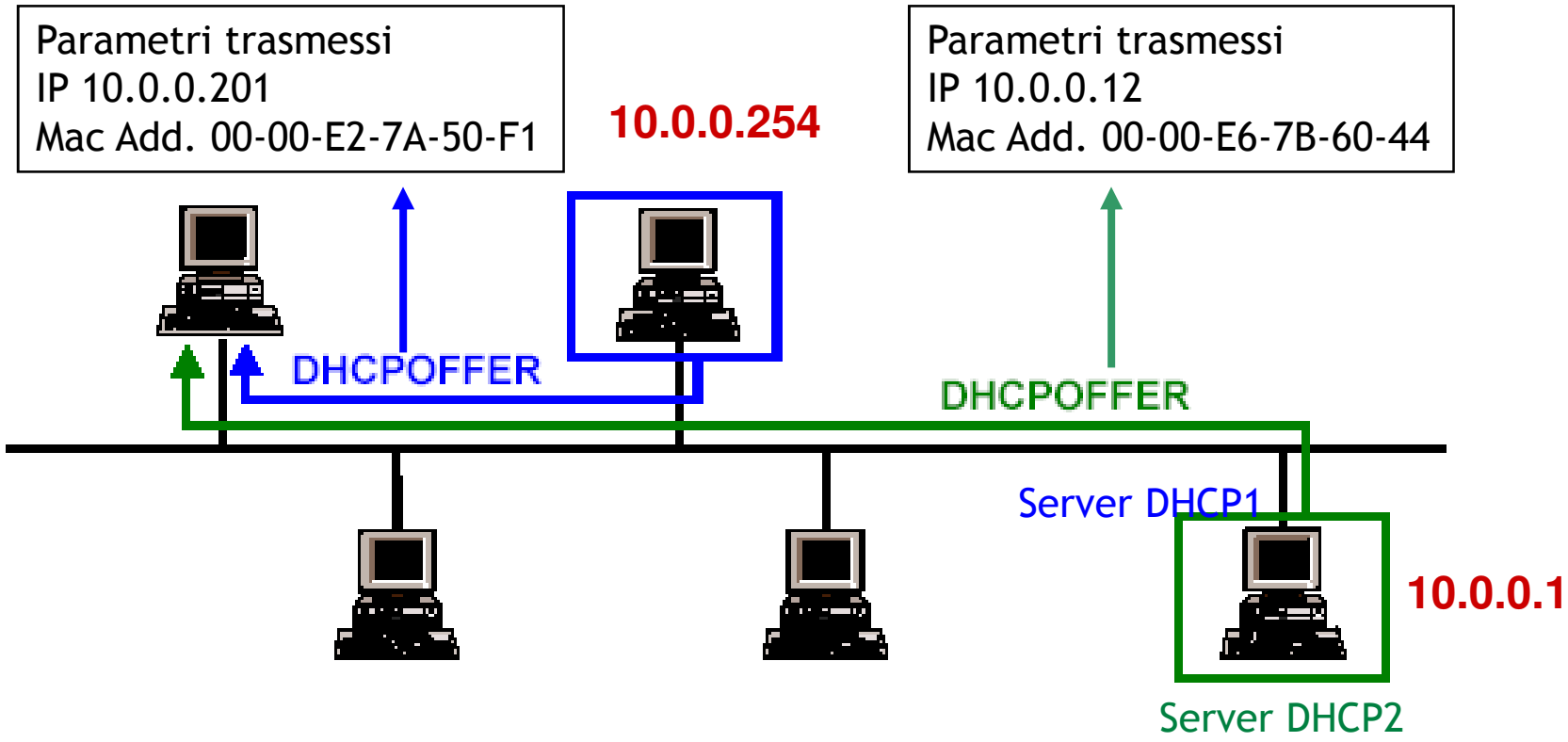
- **Il client invia** DHCPDISCOVER

  (messaggio broadcast utilizzato dal client per richiedere i parametri di configurazione ad un server DHCP)



Principio di funzionamento (2a)

- **Ogni server risponde con** DHCPOFFER
  (messaggio utilizzato dai server DHCP per offrire indirizzi IP ai client che li richiedono)



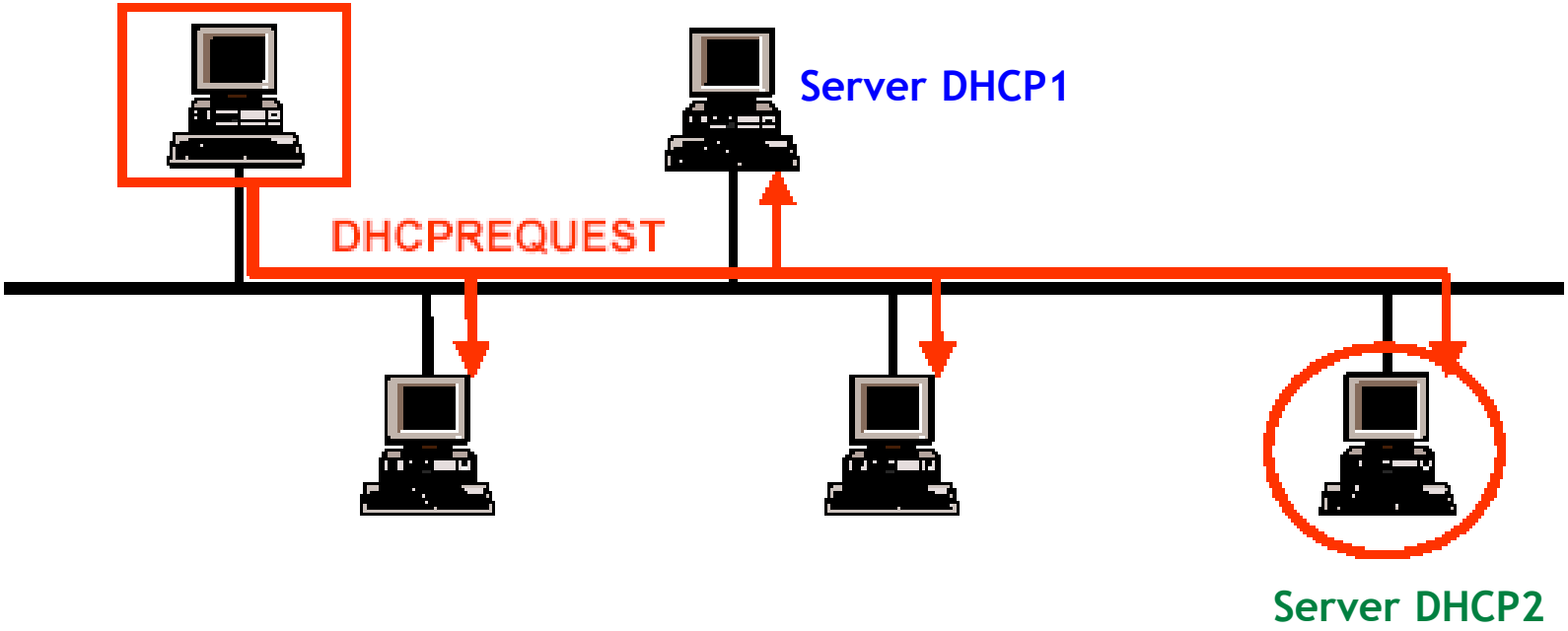
Principio di funzionamento (2b)

Note:

- **Ogni server invia il suo Mac Address poichè è l'unico identificativo VALIDO in questa fase**
- **Utilizzando il Mac Address ricevuto da ogni server il client può rispondere al server**
- **Ogni server DHCP ha un suo range di address assegnabili (Non sovrapposti)**

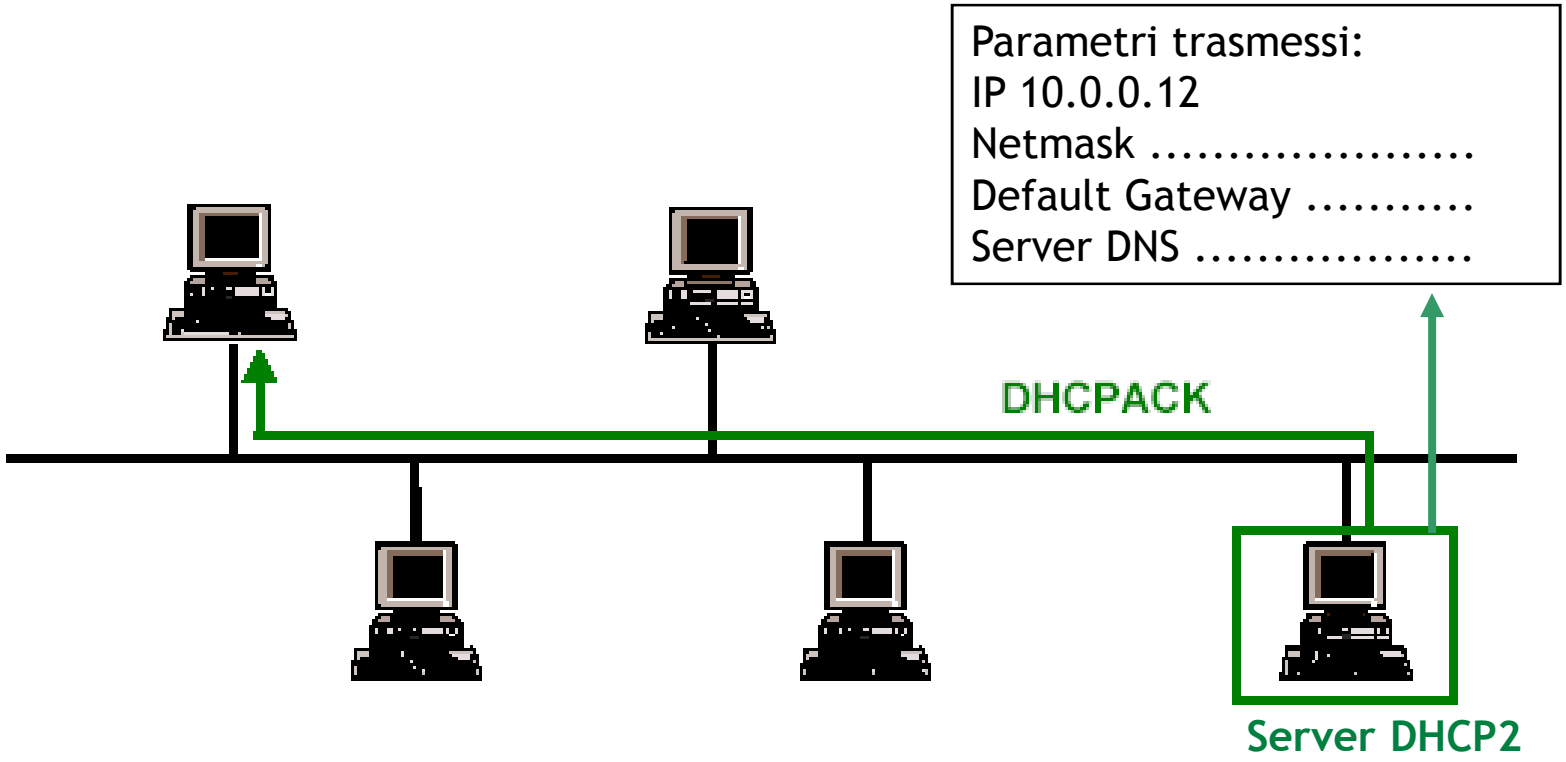
Principio di funzionamento (3)

- **Il client invia** DHCPREQUEST
  (messaggio utilizzato dal client per accettare o rinnovare l'assegnazione di un indirizzo IP, specificando il DHCP server)



Principio di funzionamento (4)

- DHCPACK
  messaggio utilizzato dal server DHCP per riconoscere l'accettazione da parte di un client di un indirizzo IP offerto specificando i parametri di configurazione

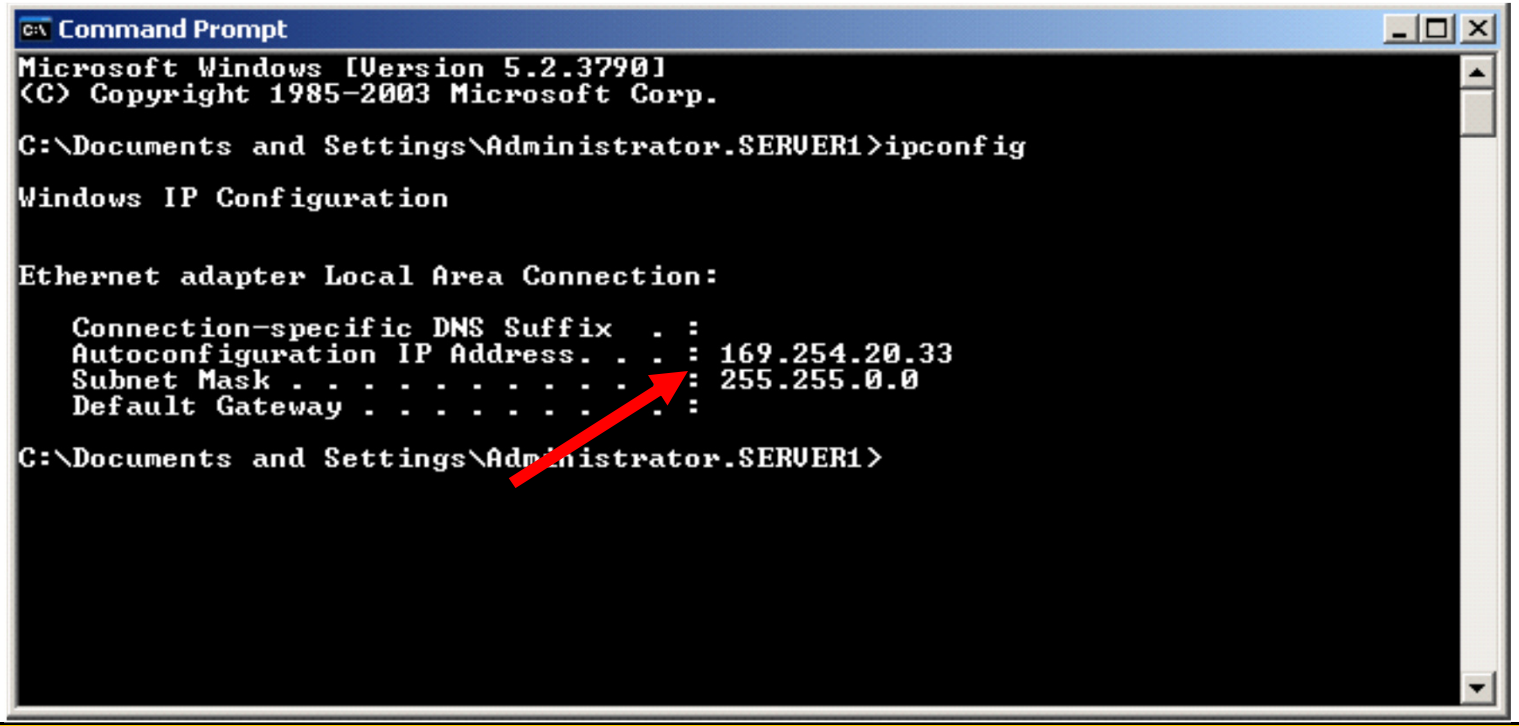


DHCP: implementazioni

- **Windows NT 4.0 - 2000 server - 2003 server:**
  - è un servizio di rete
  - una opzione sui server
  - coesiste con altri servizi
- **Unix:**
  - dhcpd è un demone
- **Apparecchiature “ibride”:**
  - Router ADSL
  - Router ISDN
  - Web sharing
  - etc...

Automatic Private IP Addressing (APIPA)

- Se non c'è un DHCP Server nel segmento di rete, un indirizzo APIPA viene assegnato ad ogni DHCP client, in modo tale che possano almeno lavorare su quel segmento (stessa subnet mask 255.255.0.0)



APIPA aka ZEROCONF

- **Il meccanismo APIPA interviene quando un DHCP_Client invia una richiesta, e nessun DHCP_Server risponde**
- **Autoconfigurazione di un indirizzo di Classe B.**
  - 169.254.0.0, con subnet mask 255.255.0.0.
  - APIPA – genera un indirizzo IP valido su questa rete
- **I clienti testano l'esistenza di eventuali conflitti (i.e. indirizzo IP già in uso)**
  - Se c'è un conflitto, ripete l'ultimo step finchè I conflitti non sono risolti
- **Il client riprova a farsi ridare un IP dal DHCP server ogni 5 minuti.**
- **Scopo – permette a piccole LANs di essere funzionanti senza l'intervento di operatori esperti**

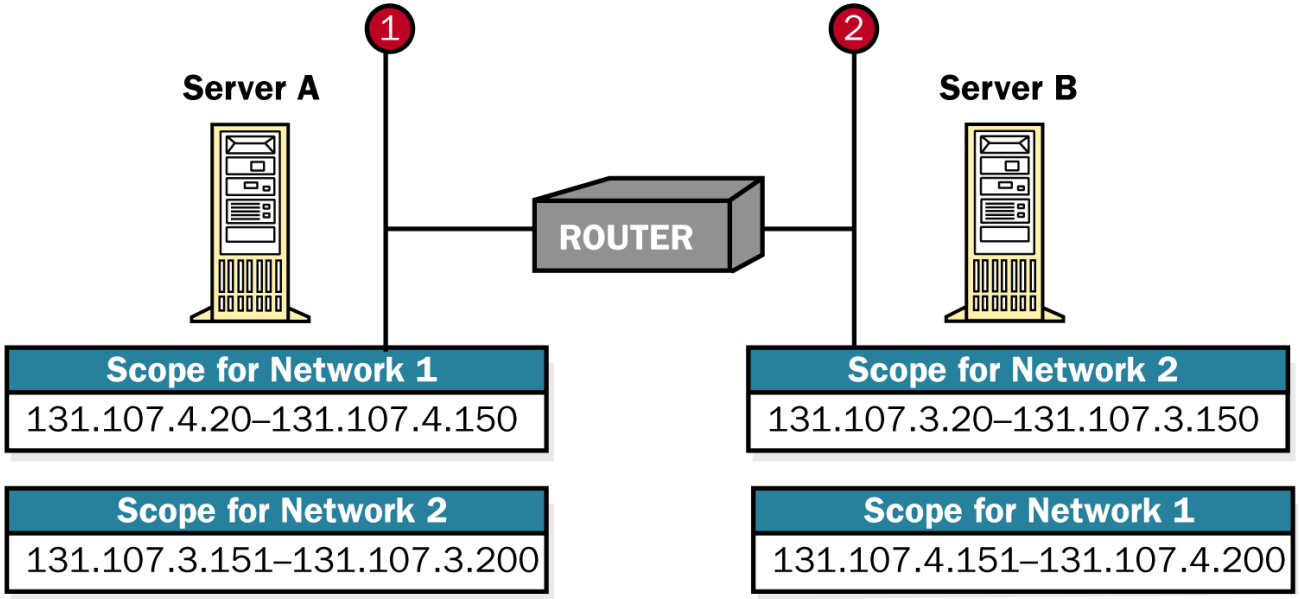
Ipconfig options

/?	Display help message
/all	Display full configuration information.
/release	Release the IP address for the specified adapter.
/renew	Renew the IP address for the specified adapter.
/flushdns	Purges the DNS Resolver cache.
/registerdns	Refreshes all DHCP leases and re-registers DNS names
/displaydns	Display the contents of the DNS Resolver Cache.

DHCP Relay Agent

- **DHCP broadcasts may not, by default, be forwarded to other subnets by routers.**
- **Many routers can be configured to pass DHCP/BOOTP messages to other segments**
  - referred to as BOOTP Relay
  - Allows for centralization of DHCP services in one place or on selected subnets
- **Windows NT/2000/2003 can be configured as a DHCP Relay Agent**
  - Forwards DHCP messages between clients and servers on subnets, useful when routers won't do BOOTP Relay
- **If neither of the above, need to set up separate DHCP servers for each subnet**

Scope and IP Address Ranges for Server A and Server B



Network Address Translation

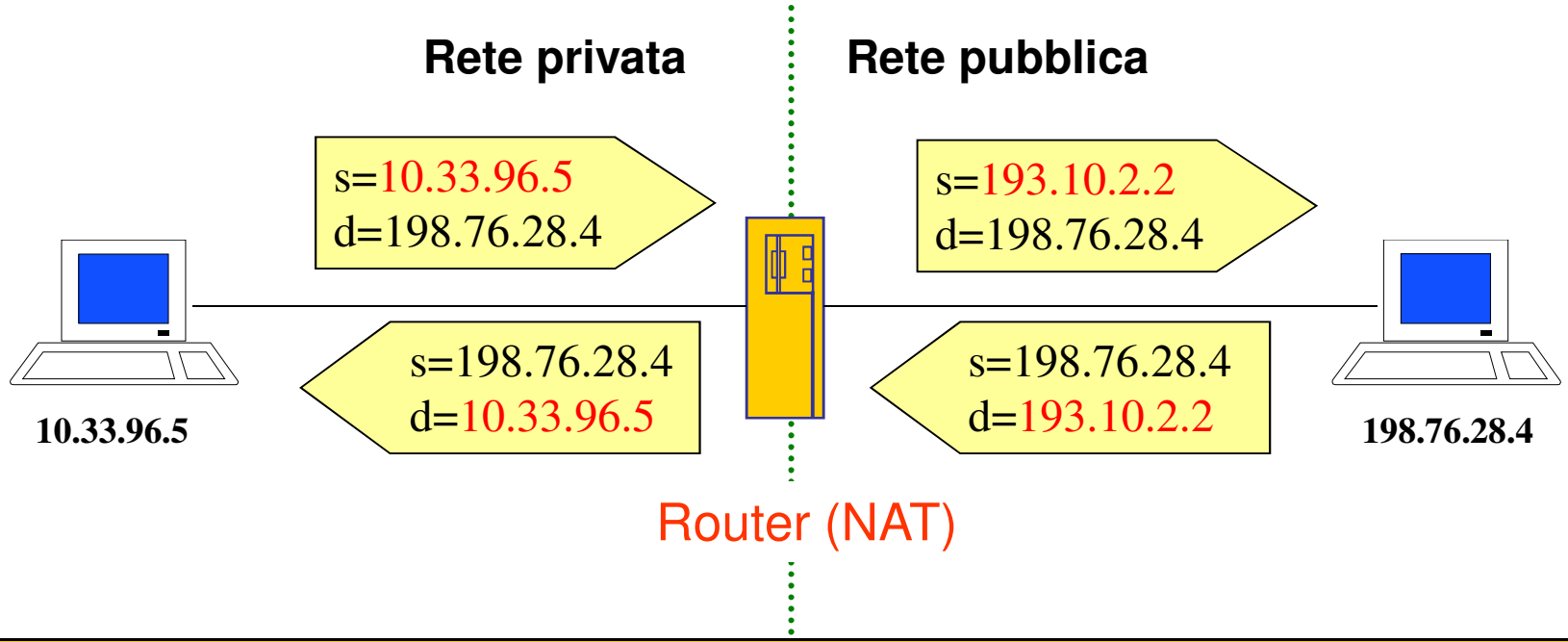
NAT

Luigi Vetrano

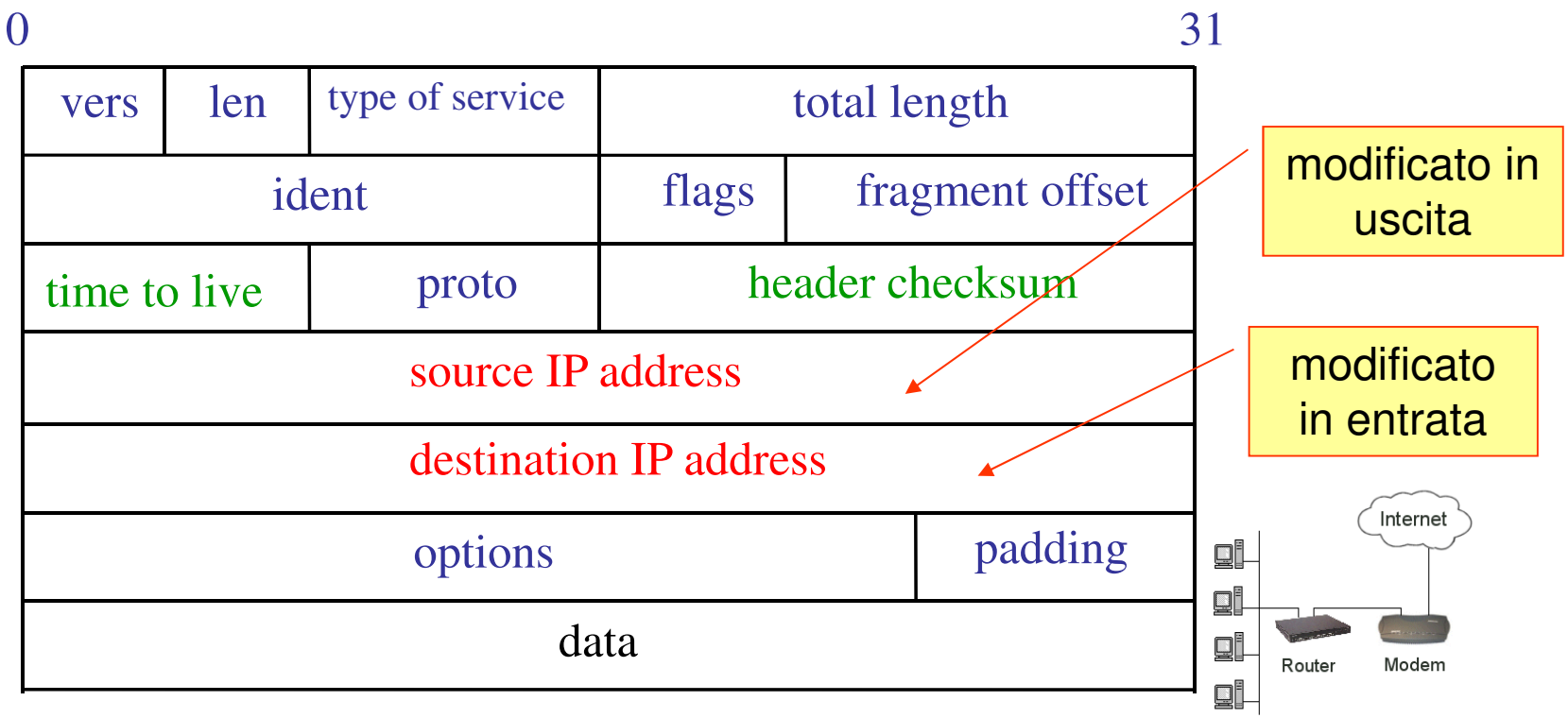
TechnoLabs

Network Address Translation (NAT)

- Una rete locale ha un numero limitato di indirizzi IP pubblici assegnati dal provider.
- Tutte le macchine possono comunicare simultaneamente su Internet sfruttando un indirizzo IP pubblico grazie al metodo del Network Address Translation (NAT).
- Il funzionamento del NAT risiede in un dispositivo di routing, che traduce gli indirizzi IP sorgenti dei pacchetti interni; l'IP interno viene cioè scartato ed al suo posto viene inserito l'indirizzo IP pubblico dell'interfaccia esterna del router.
- In questo modo i pacchetti possono essere inoltrati esternamente e gestiti dai sistemi di destinazione; il server remoto avrà così la possibilità di recapitare a destinazione le risposte.



Modifica dei pacchetti IP



Indirizzi IP pubblici

Gli indirizzi IP pubblici vengono, in genere assegnati in modo dinamico, cioè cambiano ad ogni connessione. In alcuni casi, si assegnano IP statici: si tratta, in sostanza, di IP che non variano ad ogni connessione. In genere, questi IP **vengono offerti sotto forma di servizio a pagamento.**

Un computer necessita di un indirizzo IP **pubblico statico nel caso in cui si debba offrire un servizio Internet, ad esempio una macchina su cui è installato un Server Web.**

- TECHNOLABS 88.52.22.22
- MSN 213.199.154.47
- CISCO 198.133.219.25
- INFOSTRADA 193.76.212.93
- TIN 62.211.64.8

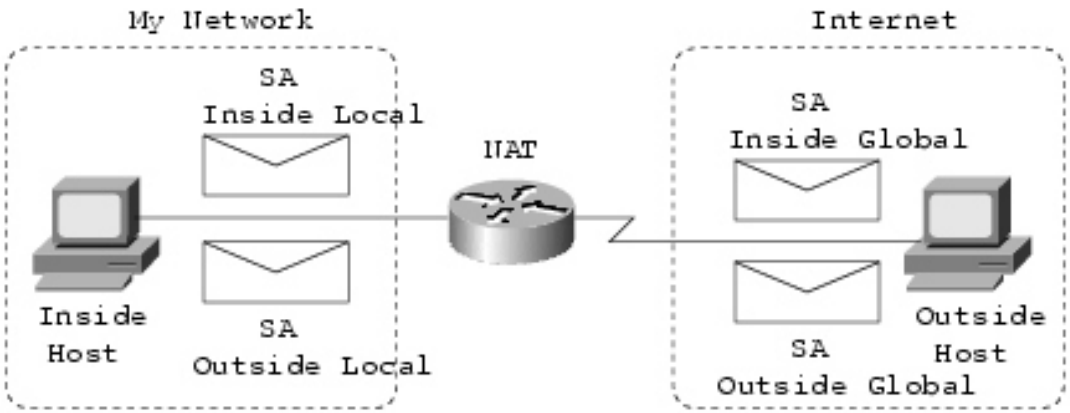
Indirizzi IP privati

Numero di indirizzi	dall'indirizzo	all'indirizzo
16 milioni	10.0.0.0	10.255.255.255
65 mila	192.168.0.0	192.168.255.255
4096	172.16.0.0	172.31.0.255

Questi numeri possono essere assegnati in piena autonomia alle macchine che fanno parte di una rete privata e permetteranno a tutte le macchine di utilizzare localmente i protocolli TCP/IP, ma calcolatori a cui sono assegnati questi indirizzi saranno "nascosti" ad Internet e non potranno essere visti se non dalle macchine che si trovano sulla stessa rete privata.

NAT statico e dinamico (1)

Poiché solo due calcolatori che hanno un indirizzo IP univoco e non mascherato possono scambiare dati tra loro si pone il problema di far sì che anche i calcolatori "nascosti" nella rete privata possano accedere all'esterno. Questo è stato risolto attraverso sistemi che svolgono la funzione di "Network Address Translator", sistemi cioè che si pongono fisicamente tra la rete privata e la rete pubblica e possiedono sia un indirizzo IP della rete nascosta che un indirizzo IP fisso della rete pubblica.



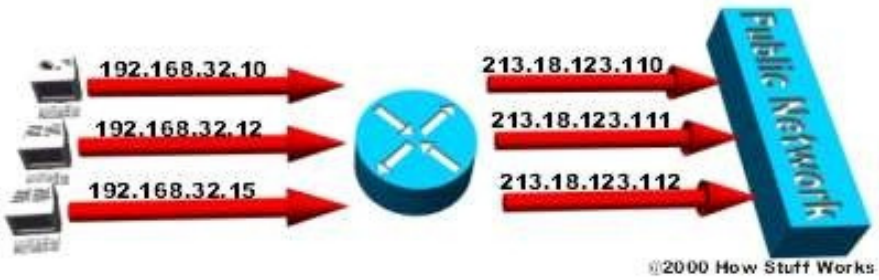
- An IP address is either local or global
- Local IP addresses are seen in the inside network
- Global IP addresses are seen in the Outside network

NAT statico e dinamico (2)

NAT statico

Il NAT **statico consiste in una mappatura biunivoca (1:1) tra indirizzi** IP **non registrati e indirizzi** IP **registrati.**

Risulta particolarmente utile quando è necessario rendere accessibile un dispositivo alla rete esterna.



NAT dinamico

Il NAT **dinamico mappa un indirizzo** IP **non registrato in un indirizzo** IP **registrato preso in un gruppo di indirizzi IP registrati e disponibili.**

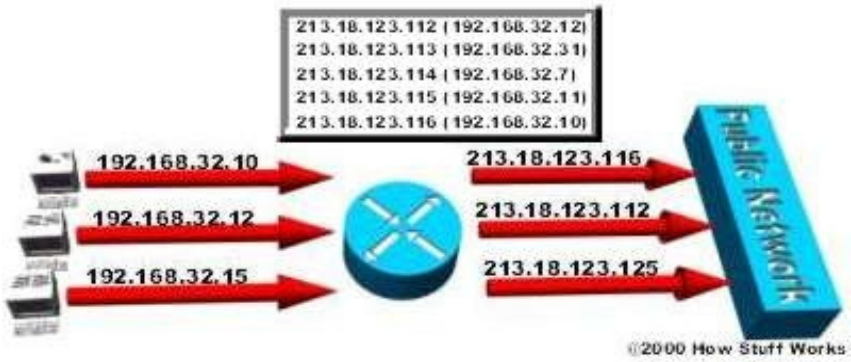
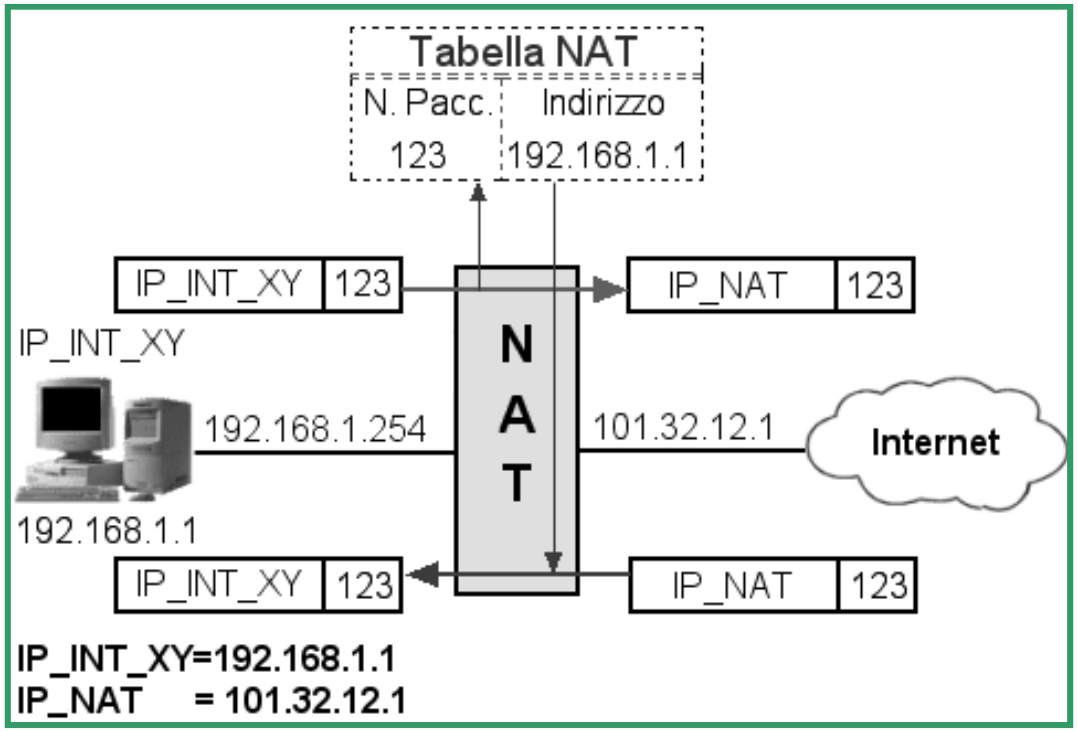


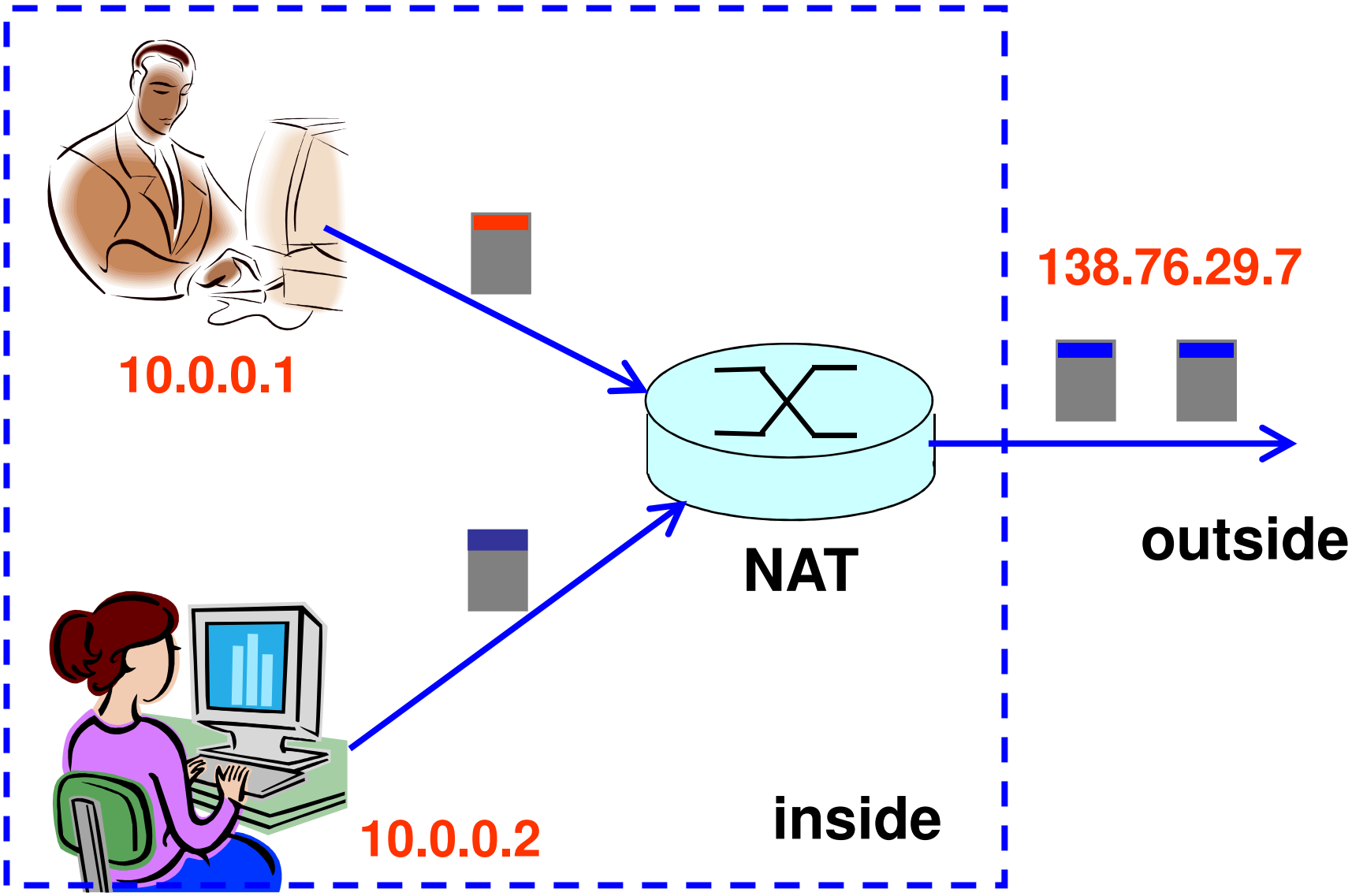
Tabelle di Natting

Quando un pacchetto di dati, proveniente dalla rete privata, viene inviato verso l'esterno è costretto ad attraversare il NAT dove il pacchetto viene modificato assumendo l'indirizzo del NAT stesso e contemporaneamente viene aggiornata una tabella di attraversamento.

Quando giunge il messaggio di risposta, relativo al pacchetto inviato, viene consultata la tabella di attraversamento e, in base a questa, il NAT individua il calcolatore della rete privata che ha fatto l'interrogazione.



Usare un Singolo Source Address



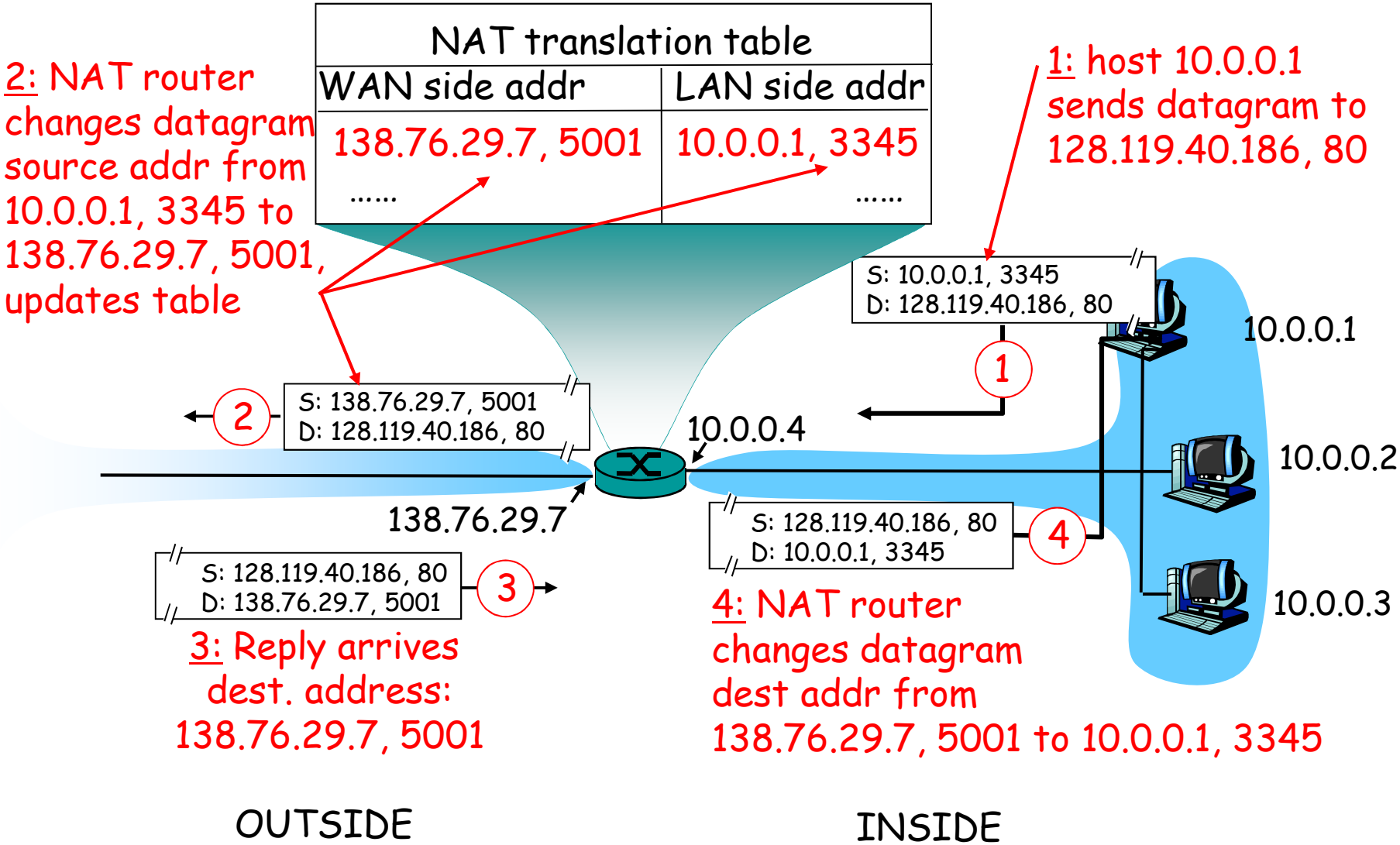
Cosa succede se entrambi contattano lo stesso Sito ?

- **Supponiamo che i due hosts contattino la stessa destinazione**
  - E.g., entrambi aprono una socket con local port 3345 alla destinazione 128.119.40.186 sulla porta 80
- **Il router-NAT assegna ai packets lo stesso source address**
  - Quindi tutti i packets hanno source address 138.76.29.7
- **Problemi**
  - Può la destinazione discriminare tra i senders ?
  - Può ritornare il traffico all'host corretto ?

Port-Translating NAT

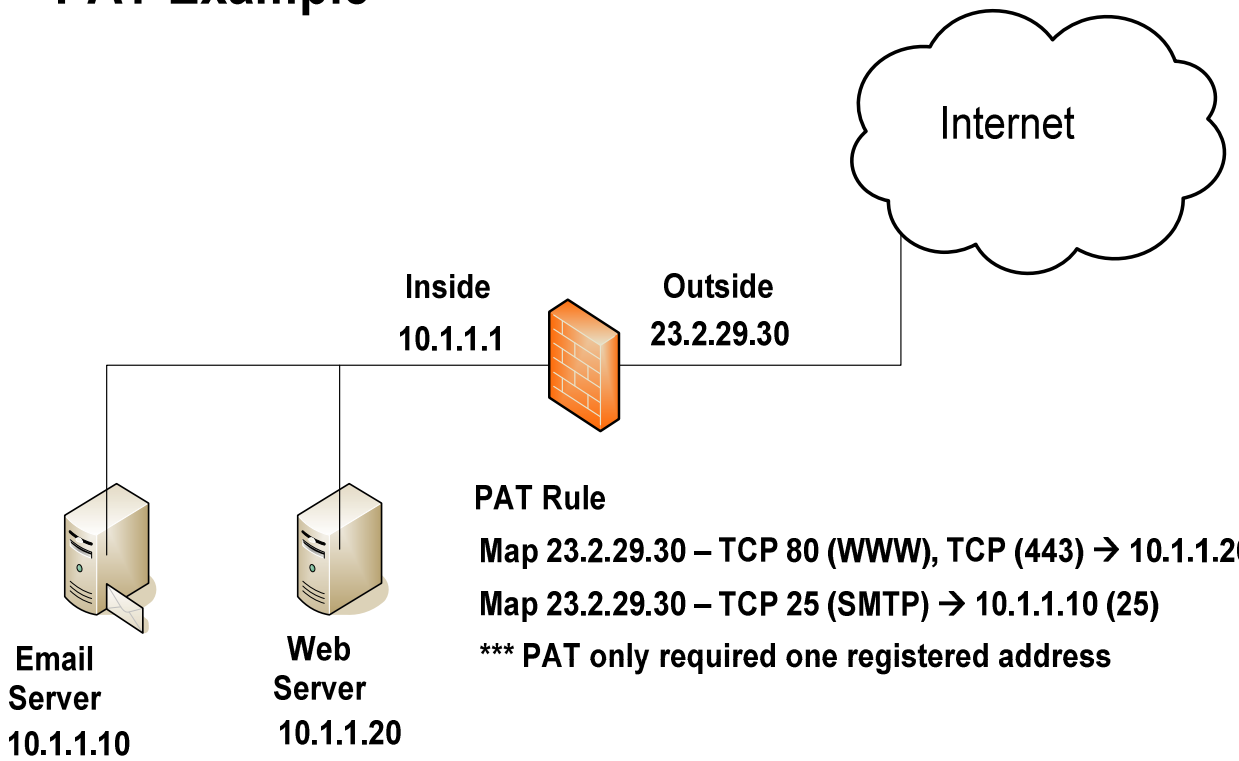
- **Mappaggio dei pacchetti in uscita**
  - Rimpiazzare source address con **NAT address**
  - Rimpiazzare source port number con **new port number**
  - Remote host risponde usando (NAT address, new port #)
- **Mantenere una translation table**
  - (source address, port #) → (NAT address, new port #)
- **Mappaggio dei pacchetti in ingresso**
  - Consultare la translation table
  - Sostituire destination address e port number
  - L'Host locale riceve il pacchetto correttamente

Esempio: Network Address Translation



Port Address Translation (PAT)

PAT Example



PAT Rule

Map 23.2.29.30 – TCP 80 (WWW), TCP (443) → 10.1.1.20

Map 23.2.29.30 – TCP 25 (SMTP) → 10.1.1.10 (25)

***** PAT only required one registered address**

Vantaggi e svantaggi del Natting (1)

Vantaggi

- Sicurezza: Crea automaticamente un firewall tra la rete interna e quella esterna
- Accesso a Internet: è sufficiente un solo indirizzo IP non mascherato per "far vedere la rete" a decine di macchine. Non fa alcuna differenza in questo caso che l'indirizzo IP sia fisso o assegnato dinamicamente dal provider, in quanto la tabella di instradamento viene azzerata e ricostruita ad ogni nuova connessione alla rete

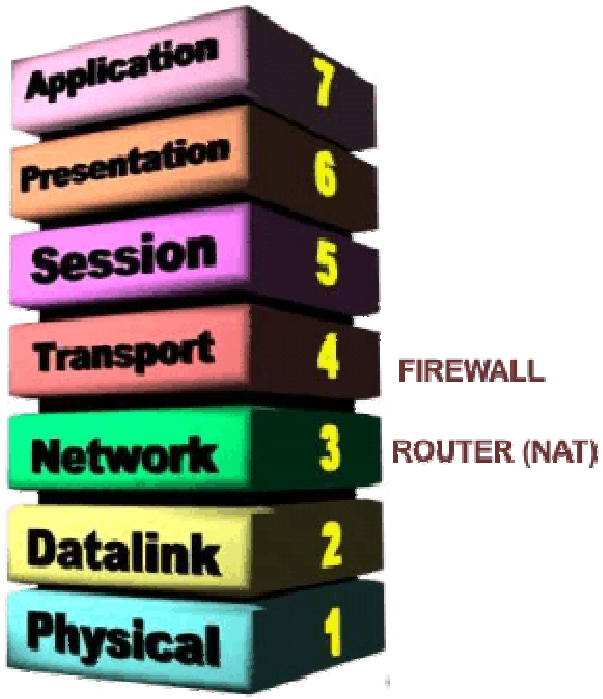
Svantaggi

- Nessun tracking delle informazioni relative ai protocolli utilizzati se non il numero di porta.
- Non si ha un reale controllo su chi può far partire le connessioni dall'interno della rete verso l'esterno, se non definendo nelle access list quali host possono eseguire queste operazioni di connessione

Vantaggi e svantaggi del Natting (2)

Considerazioni

- Il NAT è talvolta confuso con il Proxy Server; il NAT è trasparente sia per il computer sorgente che per quello di destinazione, un Proxy Server invece non è trasparente, il computer sorgente sa che sta inoltrando una richiesta al Proxy Server e deve essere configurato opportunamente.
- Inoltre i Proxy Server lavorano normalmente a partire dal livello 4 del modello di riferimento OSI mentre il NAT lavora a livello 3.
- Il Proxy Server lavorando a livelli OSI più alti rispetto al NAT risulta generalmente più lento



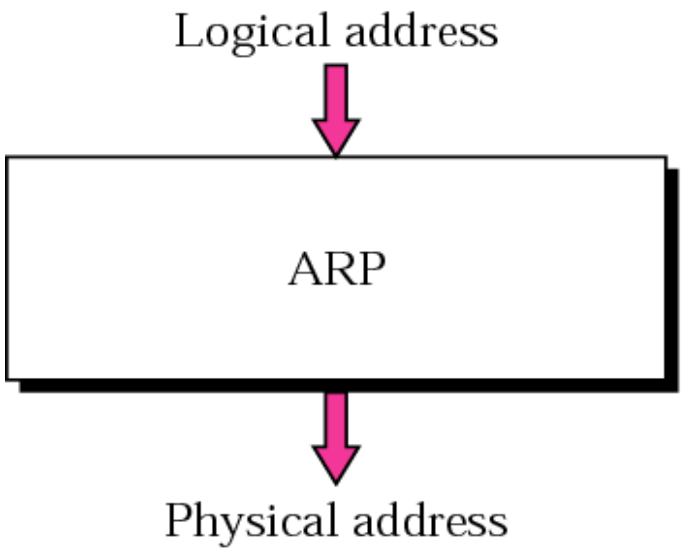
ARP and RARP

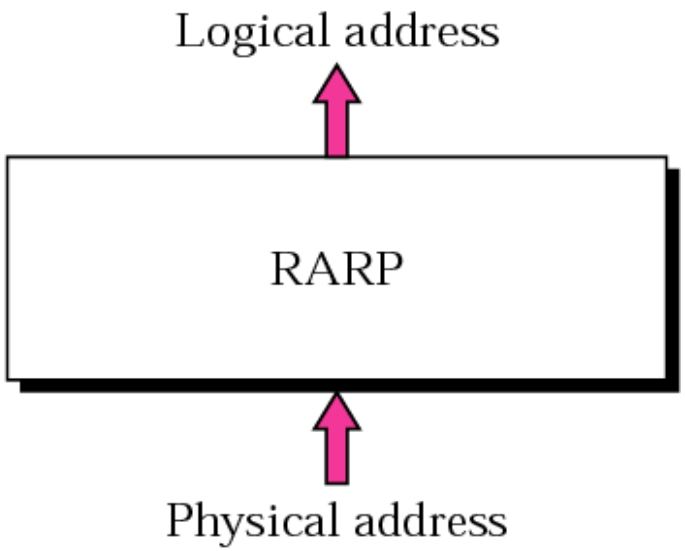
1. *Understand the need for ARP*
2. *Understand the cases in which ARP is used*
3. *Understand the components and interactions in an ARP package*
4. *Understand the need for RARP*

Luigi Vetrano

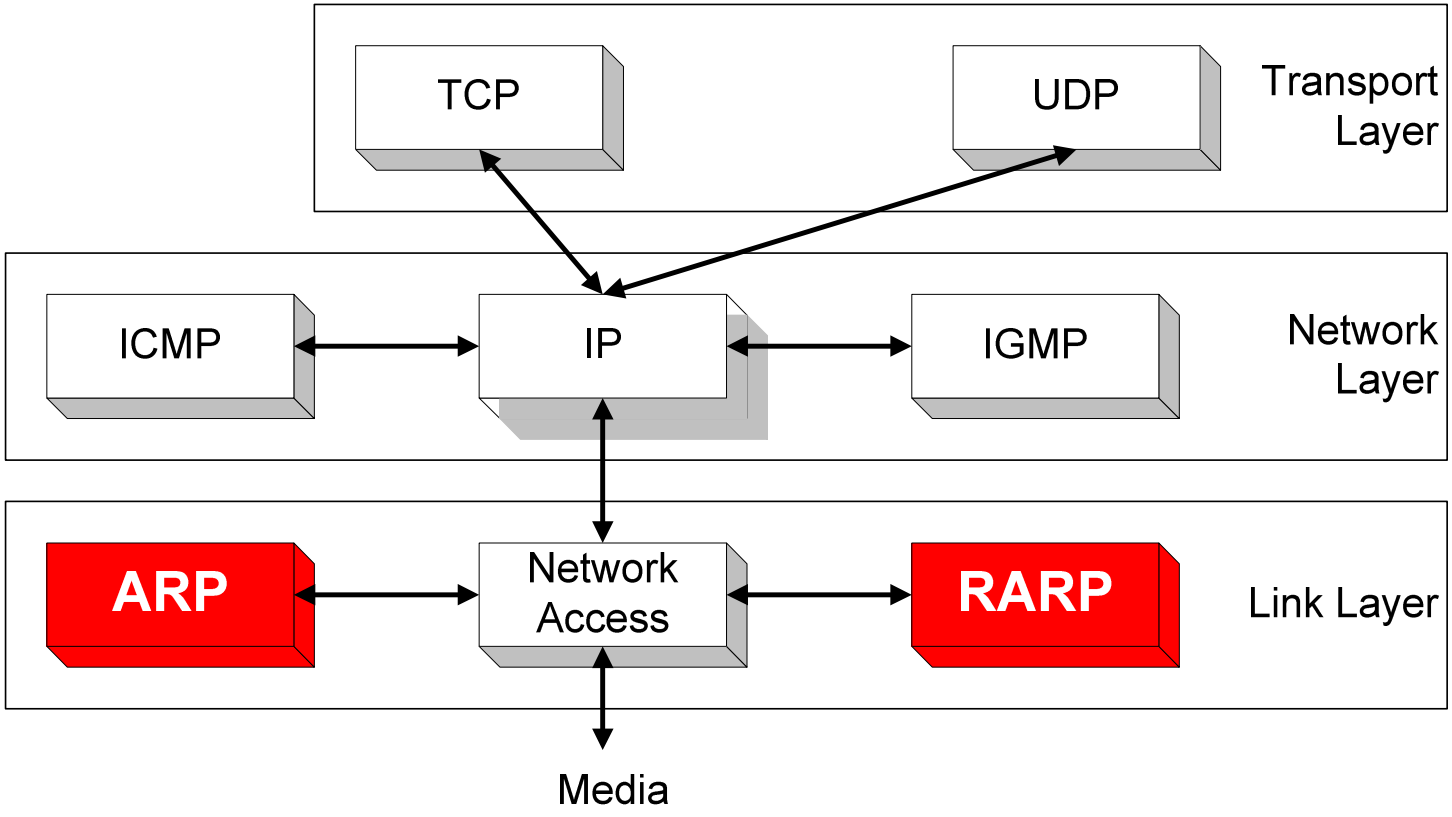
TechnoLabs

ARP and RARP

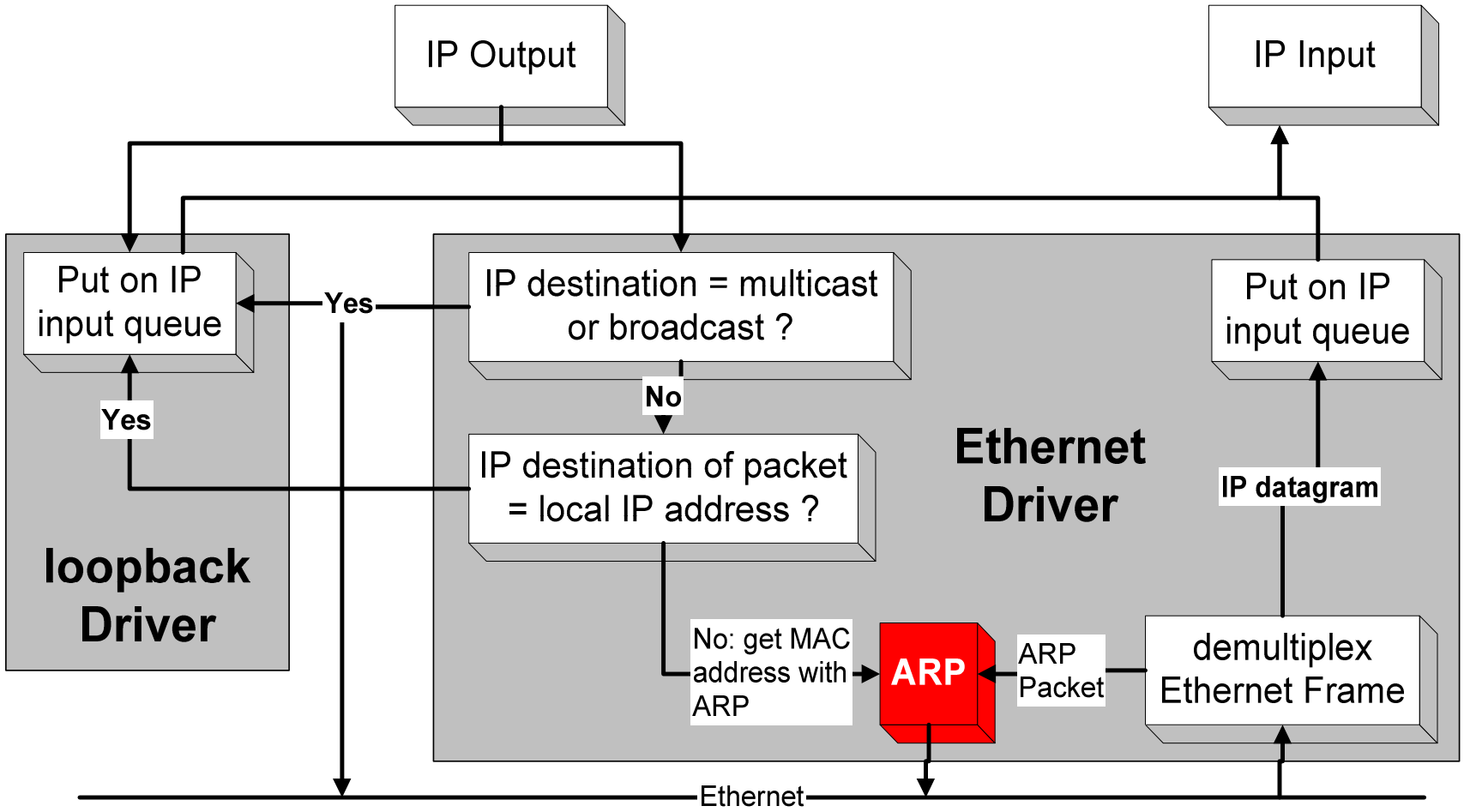

Logical address
ARP
Physical address


Logical address
RARP
Physical address

Position in TCP/IP suite

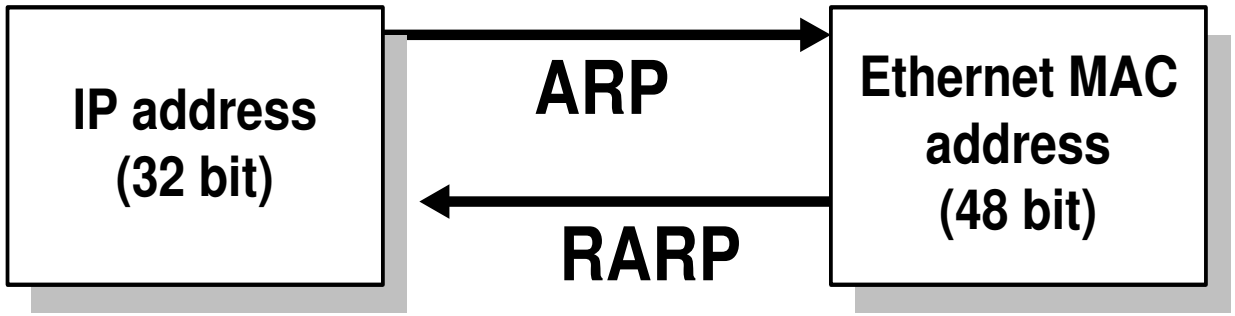


Processing of IP packets by network device drivers



ARP and RARP

- **Note:**
  - The Internet is based on IP addresses
  - Data link protocols (Ethernet, FDDI, ATM) may have different (MAC) addresses
- **The ARP and RARP protocols perform the translation between IP addresses and MAC layer addresses**
- **We will discuss ARP for broadcast LANs, particularly Ethernet LANs**

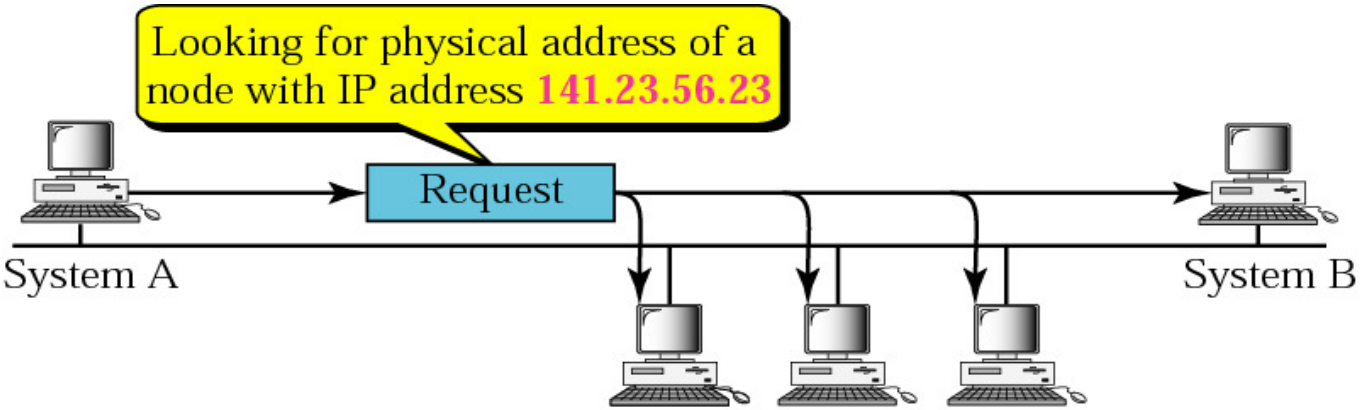


ARP

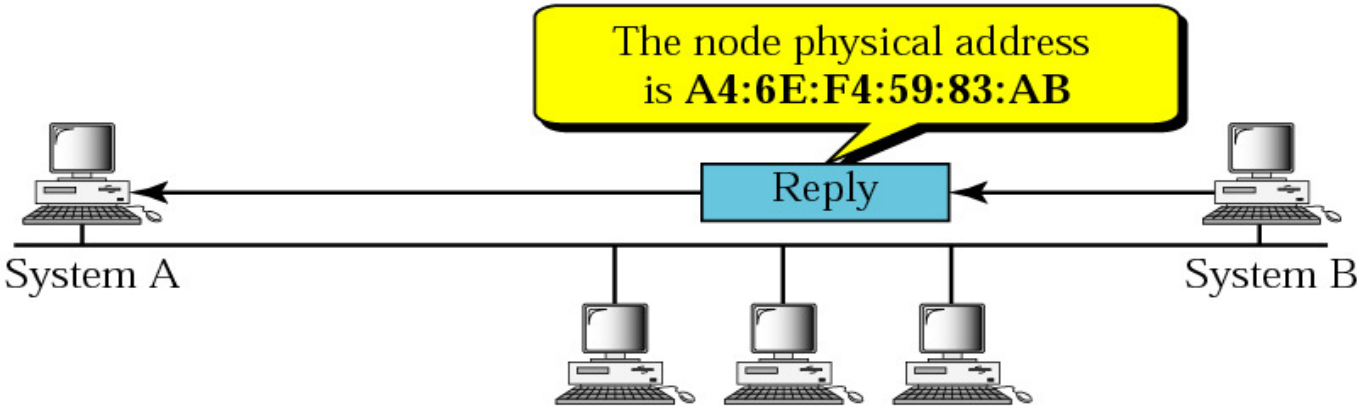
ARP associates an IP address with its physical address. On a typical physical network, such as a LAN, each device on a link is identified by a physical or station address that is usually imprinted on the NIC.

Logical address to physical address translation can be done statically (not practical) or dynamically (with ARP).

ARP operation



a. ARP request is broadcast



b. ARP reply is unicast

ARP packet

Hardware Type		Protocol Type
Hardware length	Protocol length	Operation Request 1, Reply 2
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled in a request)		
Target protocol address (For example, 4 bytes for IP)		

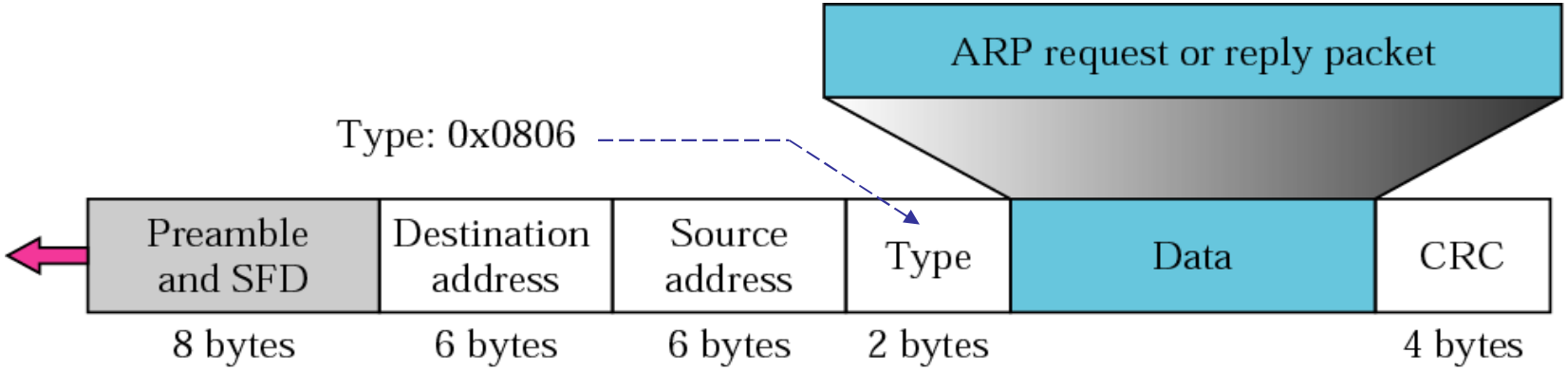
Hardware Type - Ethernet is type 1

Protocol Type-IPv4=x0800

Hardware Length:length of Ethernet Address (6)

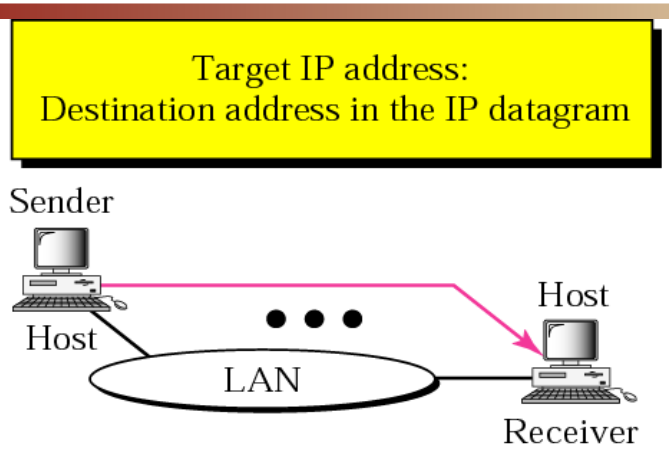
Protocol Length:length of IPv4 address (4)

Encapsulation of ARP packet

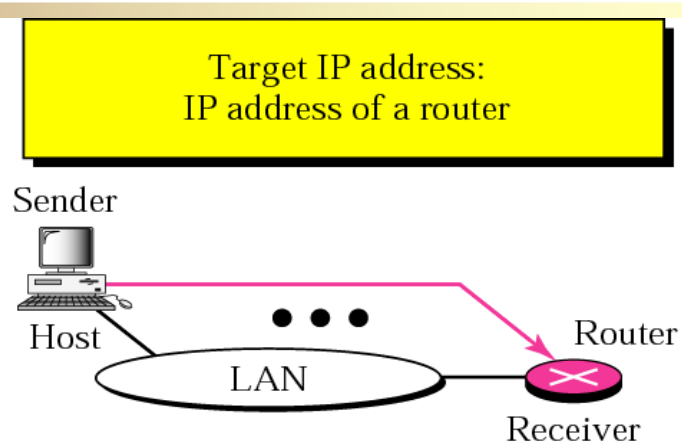


The ARP packet is encapsulated within an Ethernet packet.
Note: Type field for Ethernet is x0806

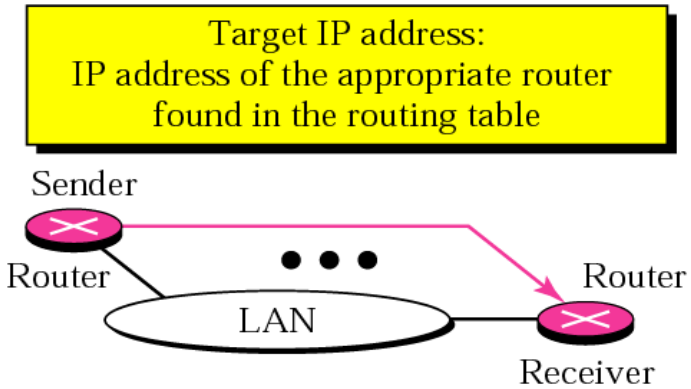
Four cases using ARP



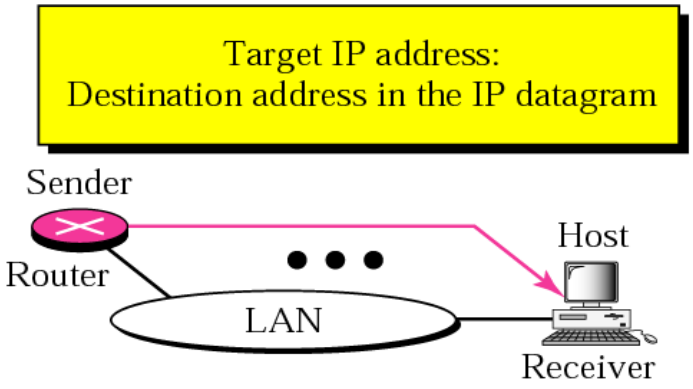
Case 1. A host has a packet to send to another host on the same network.



Case 2. A host wants to send a packet to another host on another network.
It must first be delivered to a router.



Case 3. A router receives a packet to be sent to a host on another network.
It must first be delivered to the appropriate router.



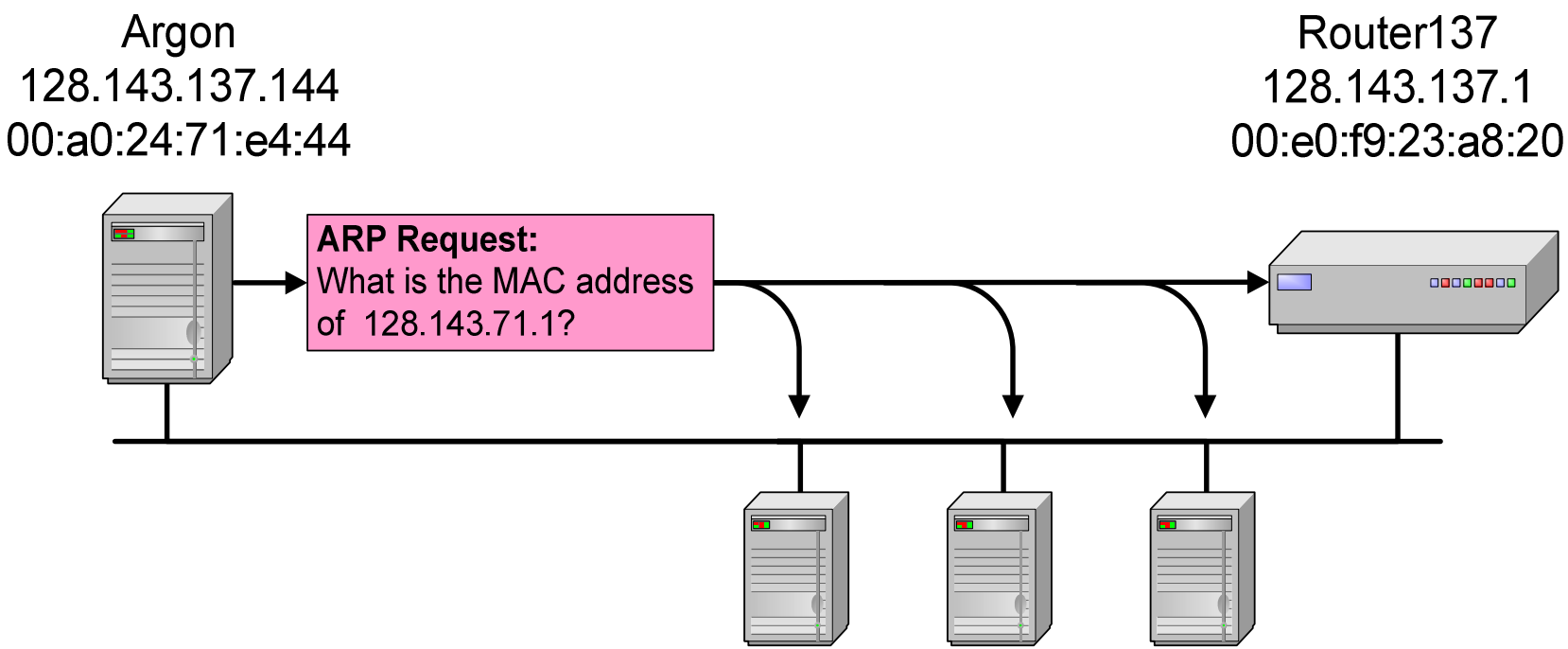
Case 4. A router receives a packet to be sent to a host on the same network.

Address Translation con ARP

ARP Request:

Argon manda in broadcast una ARP request a tutte le stazioni sulla rete LAN:

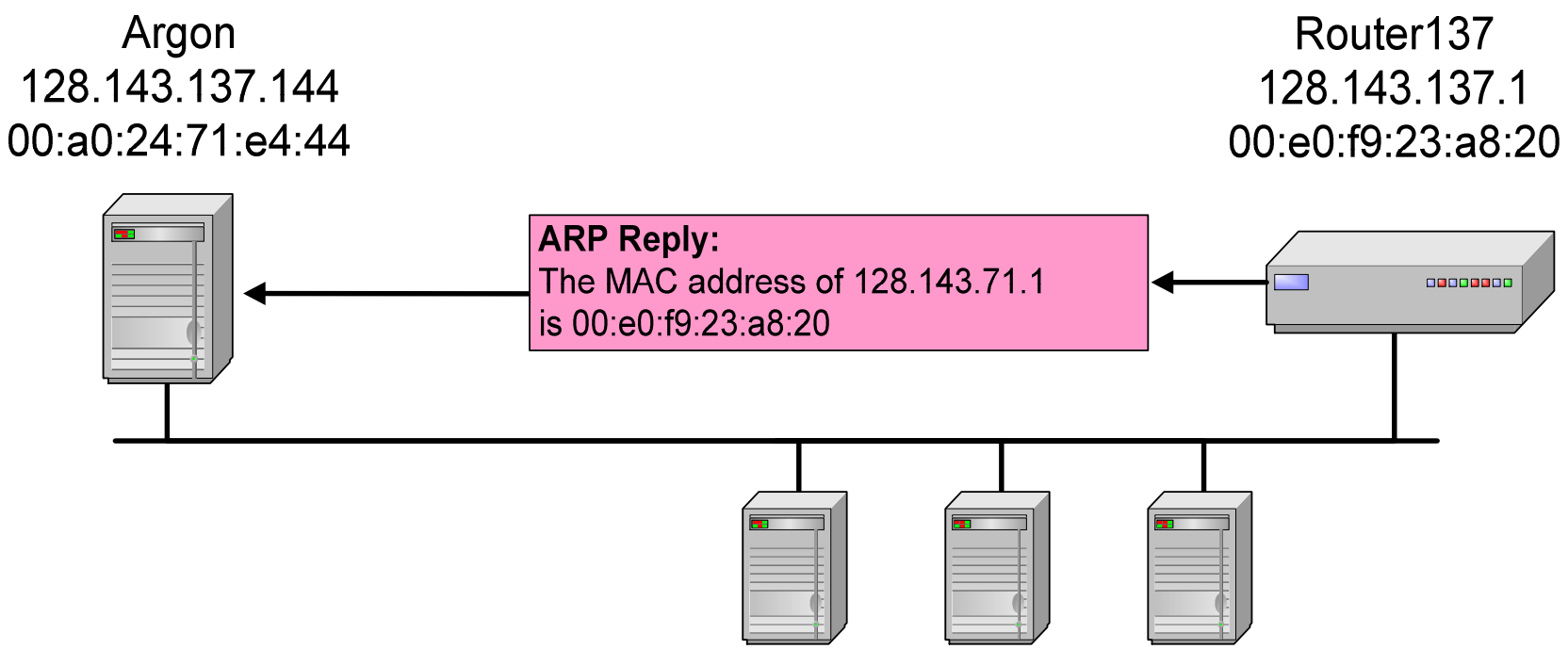
"What is the hardware address of 128.143.137.1 ?"



Address Translation with ARP

ARP Reply**:**

Router 137 responds with an ARP Reply which contains the hardware address



La ARP Table

- Ogni qualvolta ARP risolve un indirizzo IP e restituisce un indirizzo MAC, questa informazione viene memorizzata in una cache dove viene conservata per un tempo prefissato

- Ogni nuova richiesta per lo stesso IP address, verrà inoltrata prima alla cache

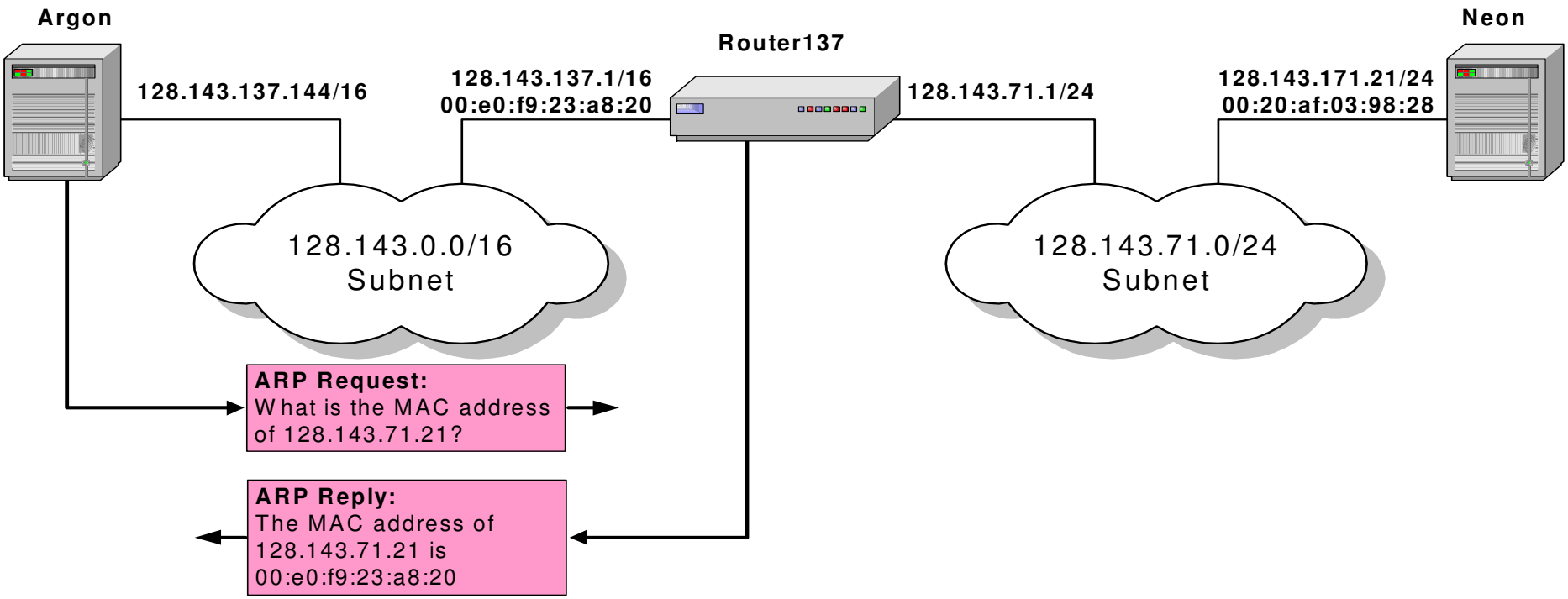
Cache Table

State	*Queue*	*Attempt*	*Time-Out*	*Protocol Addr.*	*Hardware Addr.*
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
F					
R	9		60	19.1.7.82	4573E3242ACA
P	18	3		188.11.8.71	

State: FREE, PENDING, RESOLVED

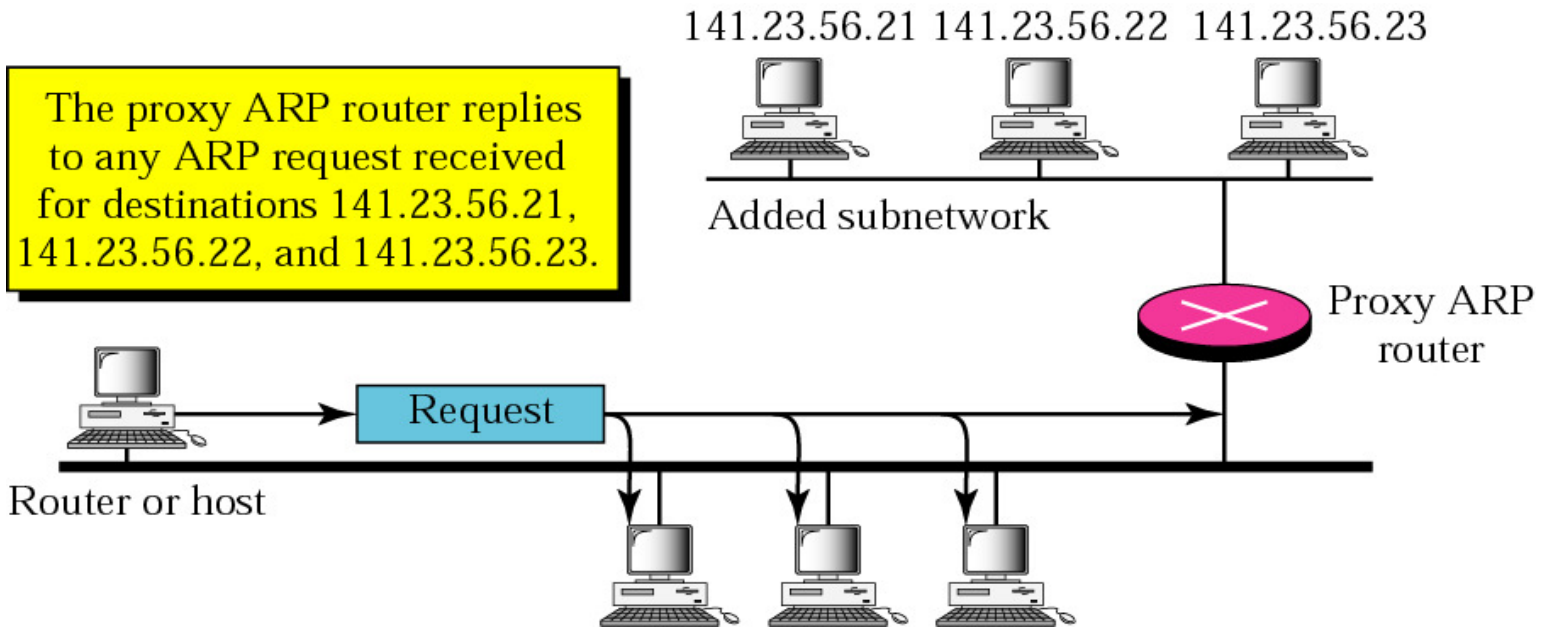
Proxy ARP

- **Host o router che risponde alle ARP Request che gli arrivano da una delle sue reti direttamente connesse per un host che è su un'altra rete direttamente connessa.**

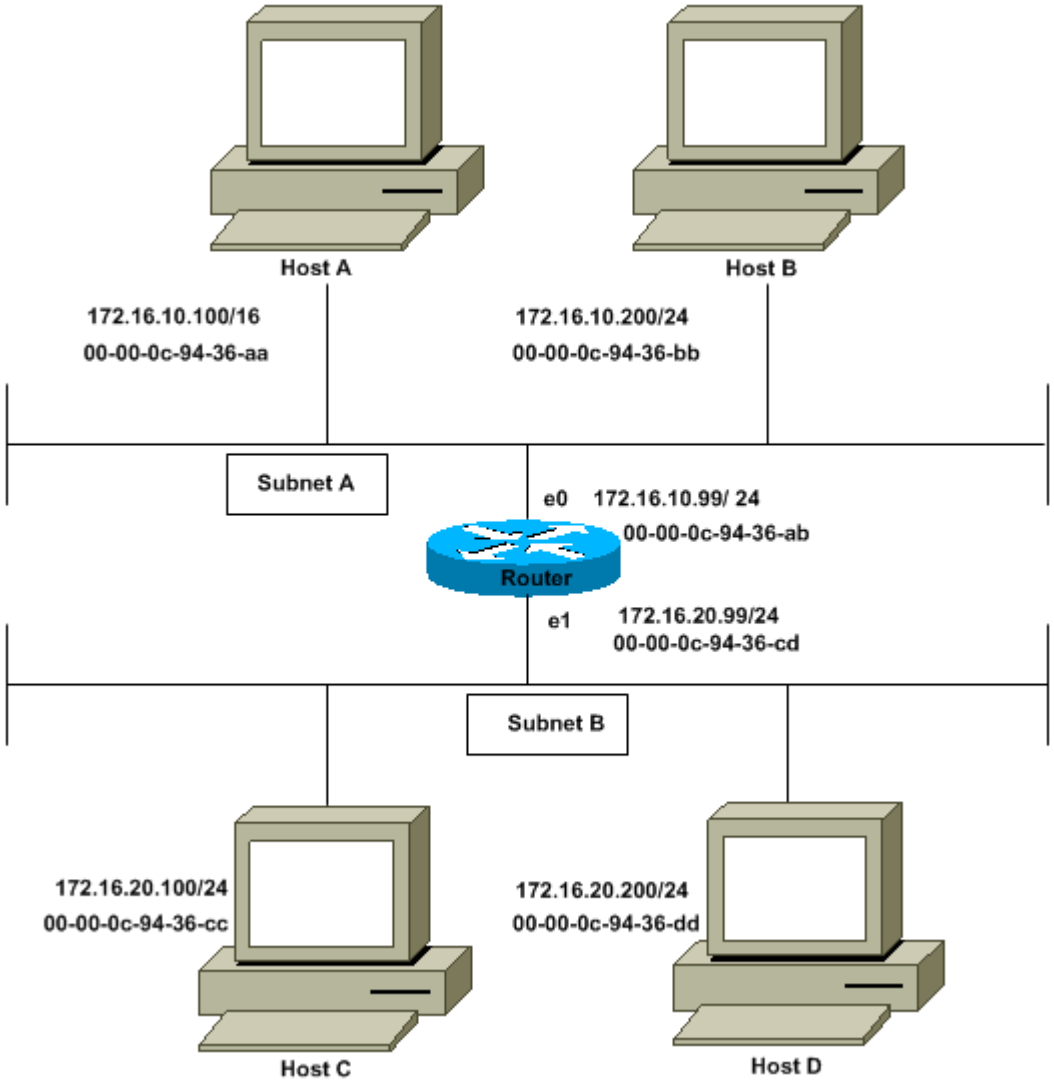


Proxy ARP

1. **Un proxy ARP, in un router, può rispondere a una richiesta ARP diretta ad uno qualunque degli host che lui 'protegge'.**
2. **Il proxy ARP risponde con il proprio MAC address.**
3. **Quando il pacchetto arriva, il router lo consegna all' host appropriato.**



Proxy ARP scenario



Cose da sapere su ARP

- **Cosa succede se c'è una richiesta ARP per un host inesistente ?**

- **Su diversi sistemi (incluso Linux) un host invia periodicamente Richieste ARP per tutti gli indirizzi listati nella sua ARP cache. Questo aggiorna il contenuto della ARP cache, ma ovviamente introduce traffico.**

- **Gratuitous ARP Requests: Un host invia una ARP request per il suo stesso IP address:**
  - Utile per determinare se un indirizzo IP sia già stato assegnato.

Vulnerabilità di ARP

1. Dal momento che ARP non autentica le richieste o le risposte, queste possono essere contraffatte
2. ARP è stateless: Una risposte ARP può essere inviata senza una corrispondente richiesta ARP(gratuitous ARP)
3. Secondo le specifiche del protocollo ARP, un nodo che riceve un pacchetto ARP (richiesta o la risposta) deve aggiornare la sua cache locale ARP con le informazioni che sovrascrivono quelle esistenti se il nodo ricevente ha già una voce per l'indirizzo IP nella sua ARP cache.

Sfruttamento tipico di queste vulnerabilità:

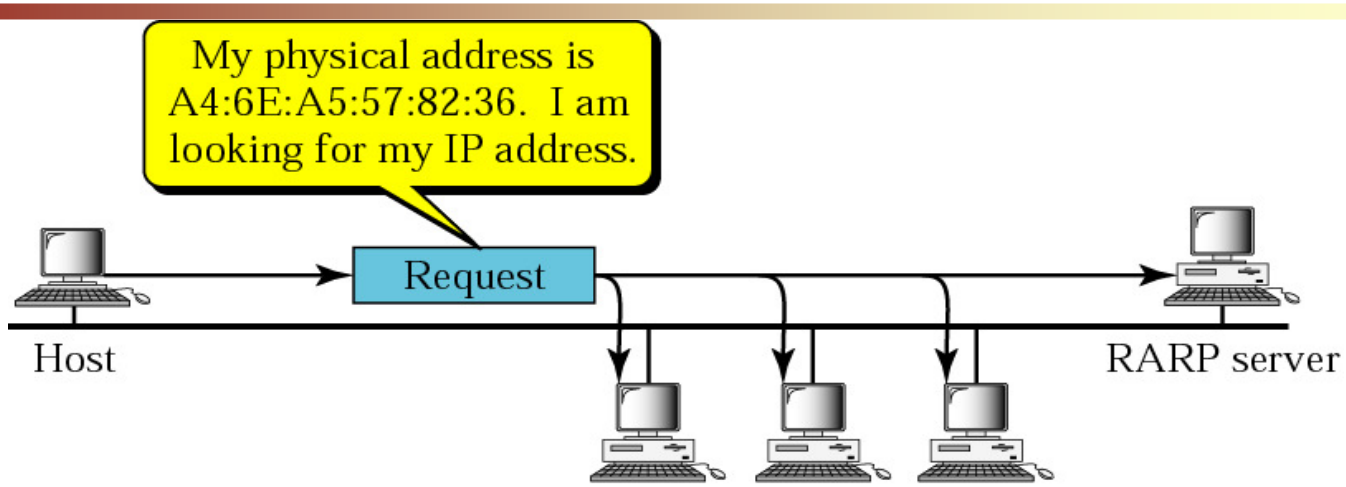
Una richiesta o risposta ARP contraffatta può essere utilizzata per aggiornare la cache ARP di un sistema remoto con una voce ad hoc (ARP Poisoning)

Questo può essere utilizzato per reindirizzare il traffico IP ad altri host

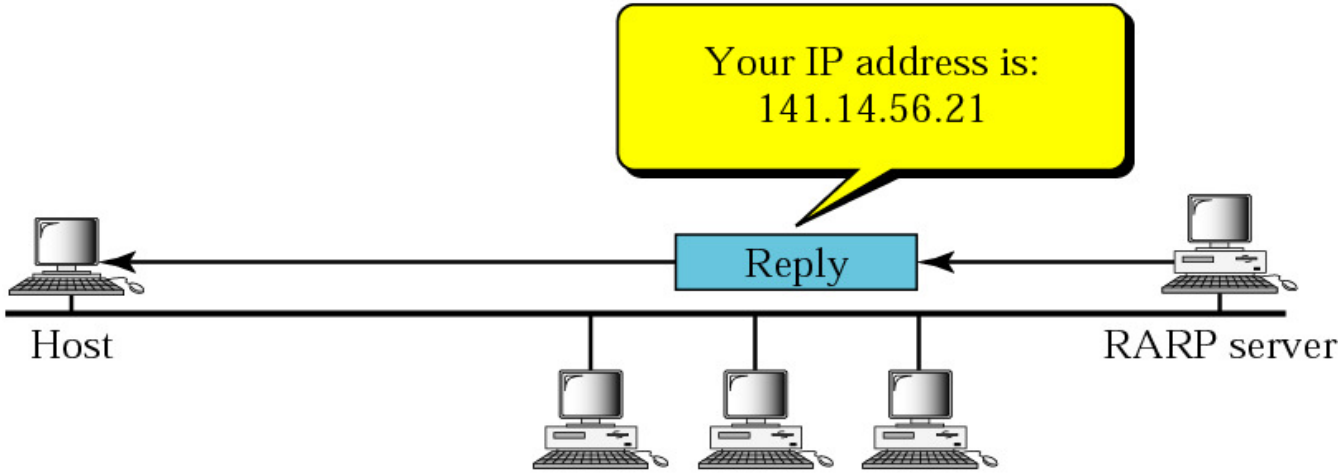
RARP

- **RARP trova l'indirizzo logico per una macchina di cui si conosce solo il suo indirizzo fisico.**
- **Questo protocollo è spesso usato su thin client workstation**
  - macchine diskless che all'avvio hanno bisogno di sapere l'indirizzo IP (non si può bruciare nella ROM).
- **Le richieste RARP sono trasmesse in broadcast mentre le risposte RARP sono unicast.**
- **Se una thin-client workstation deve conoscere il proprio indirizzo IP, probabilmente ha anche bisogno di conoscere la sua maschera di sottorete, indirizzo del router, indirizzo DNS, ecc.**
  - Quindi abbiamo bisogno di qualcosa di più di RARP.
  - BOOTP prima e DHCP poi hanno sostituito RARP.

RARP operation



a. RARP request is broadcast



b. RARP reply is unicast