



*Corso di*  
***SISTEMI TELEMATICI***  
*a.a. 2011-2012*

*Frammentazione, aggregazione e  
indirizzamento nelle reti INTERNET*



# Frammentazione e aggregazione

- Le sottoreti possono avere diverse limitazioni circa la lunghezza massima delle loro unità dati
  - ✗ per una LAN Ethernet la lunghezza max è 1500 byte; per una MAN FDDI è 4470 byte
- La dimensione max dell'unità dati di una sotto-rete è detta, in TCP/IP, Maximum Transfer Unit (MTU)
- Tipicamente la dimensione del datagramma IP è inferiore al suo valore max di 65536 byte e pari al valore della MTU della sottorete alla quale è connesso il sistema mittente
  - ✗ questa è resa nota all'entità IP mittente dal software che interfaccia IP alla sotto-rete (tale software è detto driver)
  - ✗ se la quantità di dati da trasmettere è inferiore alla MTU, il datagramma avrà dimensione minore della MTU stessa



# Frammentazione e aggregazione

Se il datagramma incontra una sottorete con una MTU di dimensione inferiore a quella scelta per il datagramma, quest'ultimo viene segmentato in più parti (frammenti)

- ✗ i frammenti non devono necessariamente essere tutti delle stesse dimensioni; almeno l'ultimo frammento ha dimensione minore degli altri
- Anche il frammento può essere frammentato a sua volta se incontra sottoreti con MTU ancora più piccole

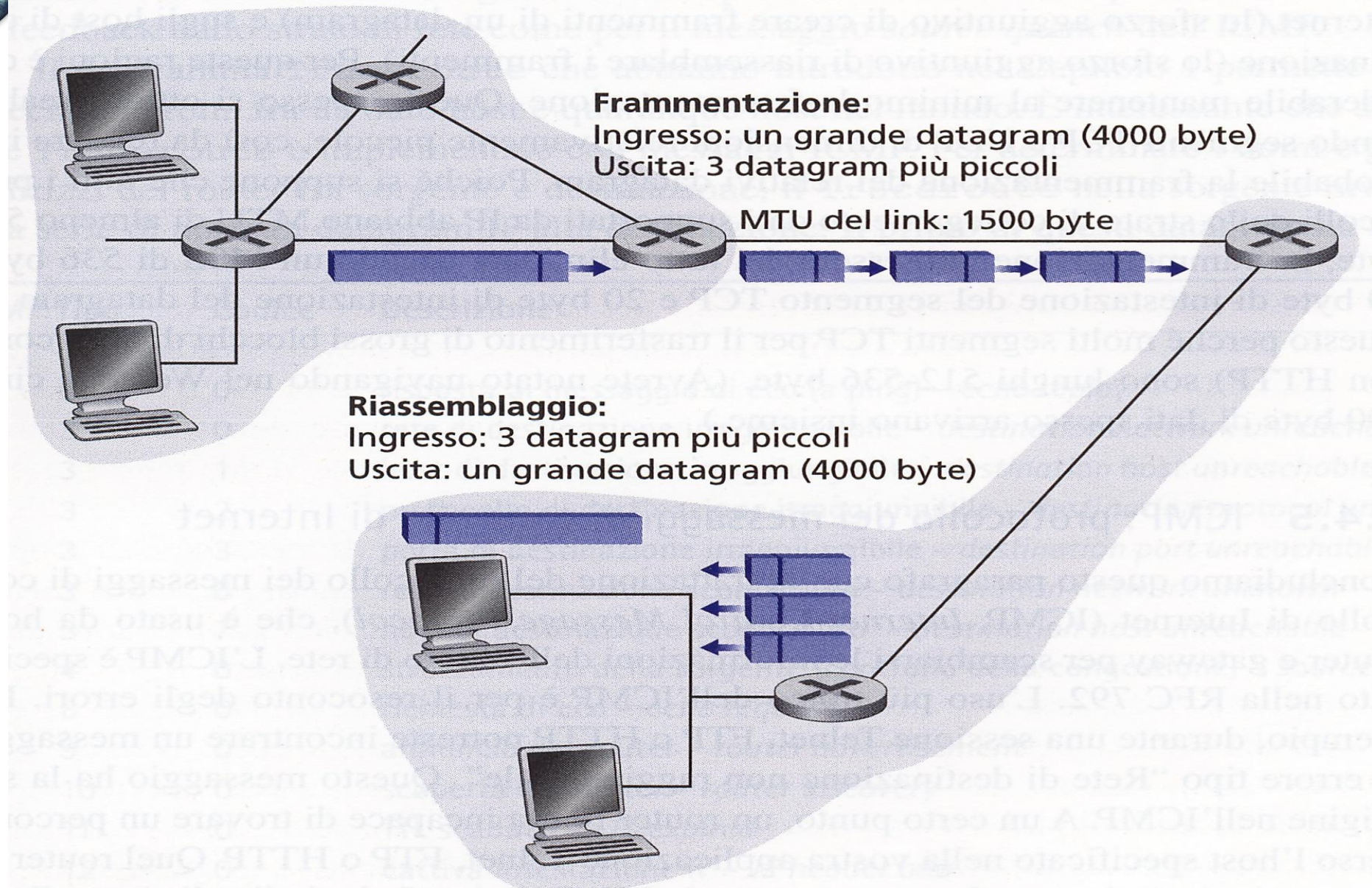


# Frammentazione e aggregazione

I soli vincoli imposti da IP sono

- ✗ i router devono accettare datagrammi di dimensioni pari a quelli delle MTU delle sottoreti interconnesse
- ✗ tutti i sistemi (host e router) devono comunque accettare datagrammi di dimensioni almeno pari a 576 byte (frammentati o non)
- ✗ I sistemi devono poter inoltrare datagrammi di 68 byte senza ulteriore frammentazione (60 byte di header al massimo + 8 byte di frammento, che è il minimo)





Esempio di frammentazione e riassemblaggio



# Frammentazione e aggregazione

- I datagrammi IP non segmentabili (bit DF=1) non vengono frammentati; ciò comporta la perdita del datagramma se incompatibile con la capacità di trasporto della rete
  - ✗ in tal caso verrà generato un messaggio ICMP
- La segmentazione/aggregazione serve a frammentare il datagramma originario in un numero arbitrario di unità e a ricomporle, a destinazione, nella forma originaria
  - ✗ il destinatario usa il campo Identification di ogni frammento che è assegnato dall'host sorgente al datagramma in modo univoco per tutti i processi che operano in un dato momento tra entità sorgenti ed entità remote
  - ✗ il campo Fragment Offset assegnato ad ogni frammento permette al destinatario di risalire alla posizione (in multipli di 8 byte) del frammento nel datagramma originario



# Frammentazione e aggregazione

L'informazione presente nell'intestazione del datagramma viene “copiata” nell'intestazione di ogni frammento; tranne il campo Total Length che viene modificato con la lunghezza del frammento in questione

- ✗ I campi Options dell'header possono venir copiati o non, o possono venir copiati solo nel primo frammento (es. RRO)
- ✗ ogni frammento diventa a sua volta un datagramma e può a sua volta essere ulteriormente frammentato





# Frammentazione e aggregazione

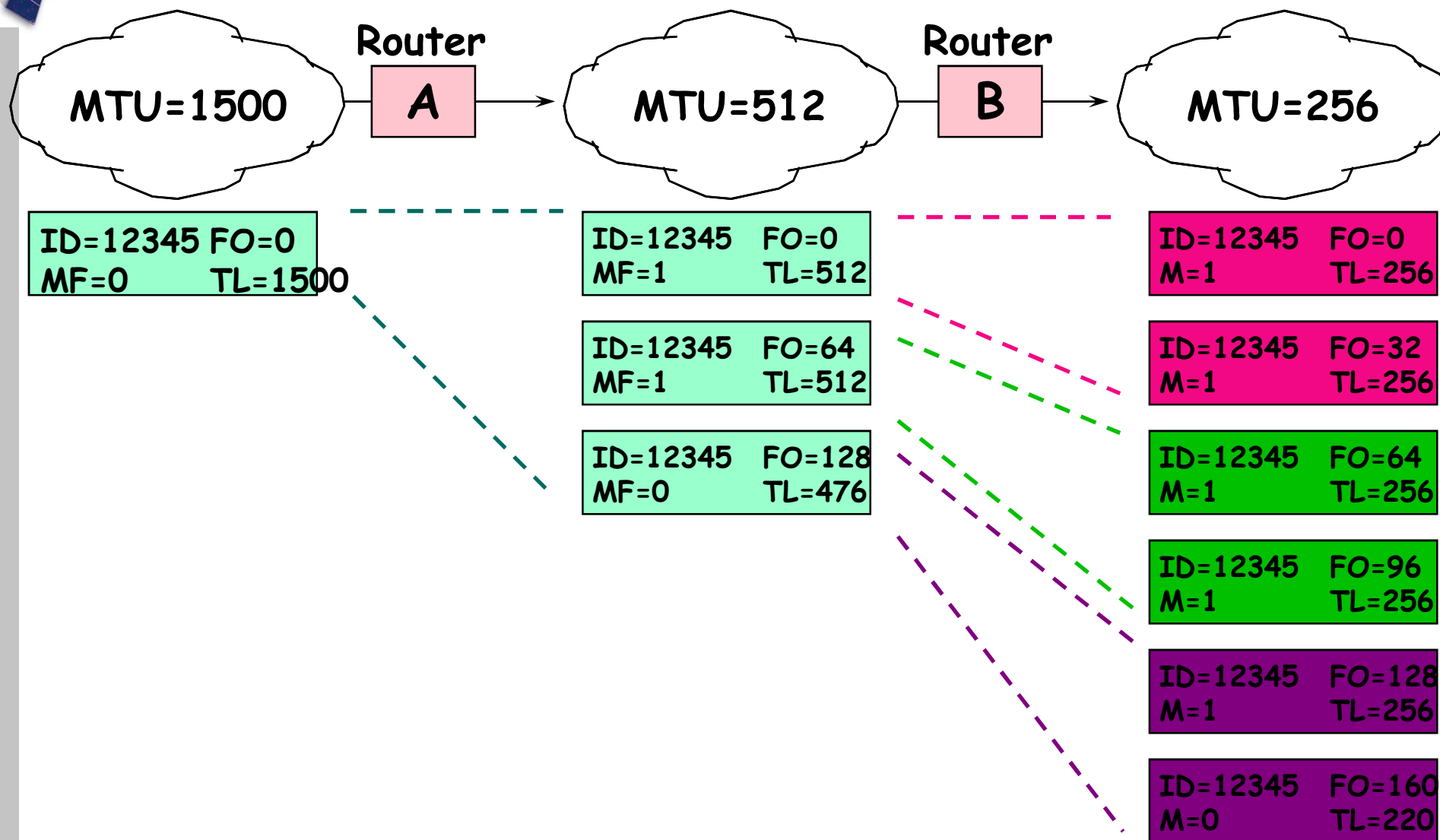
• Il datagramma originale viene ricostruito solo a destinazione

- ✗ la destinazione combina i frammenti con gli stessi valori dei campi Identification, Source e Destination Address e Protocol; la parte dati di ogni frammento è inserita nel datagramma nella posizione indicata dal campo Fragment Offset
- ✗ se uno o più frammenti vengono persi, i restanti che arrivano a destinazione vengono anch'essi scartati (dopo un time-out)
- ✗ se si utilizza TCP come protocollo di trasporto, l'intero datagramma viene ritrasmesso dalla sorgente





# Frammentazione di datagrammi IP





# Procedura di frammentazione

Notazione del pseudo-codice: " $\leq$ " significa "minore o uguale", "#" significa "diverso", "=" significa "uguale", " $\leftarrow$ " significa "assegnazione di un valore". Inoltre, "x to y" include x ed esclude y; per es. "4 to 7" include 4, 5, e 6 (non 7)

- FO - Fragment Offset
- IHL - Internet Header Length
- DF - Don't Fragment flag
- MF - More Fragments flag
- TL - Total Length
- OFO - Old Fragment Offset
- OIHL - Old Internet Header Length
- OMF - Old More Fragments flag
- OTL - Old Total Length
- NFB - Number of Fragment Blocks
- MTU - Maximum Transmission Unit



# Procedura di frammentazione

IF Total Length  $\leq$  MTU THEN submit this datagram to the next step in datagram processing

ELSE IF DF = 1 THEN discard the datagram ELSE

To produce the first fragment:

- (1) Copy the original internet header;
- (2) OldIHL  $\leftarrow$  IHL; OldTL  $\leftarrow$  TL; OldFO  $\leftarrow$  FO;  
OldMF  $\leftarrow$  MF;
- (3) Number of Fragment Blocks  $\leftarrow$  (MTU-IHL\*4)/8;
- (4) Attach the first NFB\*8 data octets;
- (5) Correct the header:  
MF  $\leftarrow$  1; TL  $\leftarrow$  (IHL\*4)+(NFB\*8);  
Recompute Checksum;
- (6) Submit this fragment to the next step in datagram processing;



# Procedura di frammentazione

To produce the second fragment:

- (7) Selectively copy the internet header (some options are not copied);
- (8) Append the remaining data;
- (9) Correct the header:  
$$\text{IHL} \leftarrow (((\text{OIHL} * 4) - (\text{length of options not copied})) + 3) / 4;$$
$$\text{TL} \leftarrow \text{OTL} - \text{NFB} * 8 - (\text{OIHL} - \text{IHL}) * 4;$$
$$\text{FO} \leftarrow \text{OFO} + \text{NFB}; \text{MF} \leftarrow \text{OMF};$$
$$\text{Recompute Checksum};$$
- (10) Submit this fragment to the fragmentation test;

DONE.



# Procedura di aggregazione

## Notazione

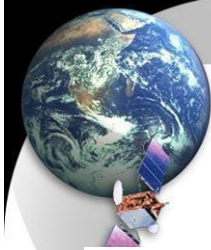
- FO** - Fragment Offset
- IHL** - Internet Header Length
- MF** - More Fragments flag
- TTL** - Time to Live
- NFB** - Number of Fragment Blocks
- TL** - Total Length
- TDL** - Total Data Length
- BUFID** - Buffer Identifier
- RCVBT** - Fragment Received Bit Table
- TLB** - Timer Lower Bound





# Procedura di aggregazione

- (1) BUFID <- source | destination | protocol | identification;
- (2) IF FO = 0 AND MF = 0
- (3) THEN IF buffer with BUFID is allocated
- (4) THEN flush all reassembly for this BUFID;
- (5) Submit datagram to next step; DONE.
- (6) ELSE IF no buffer with BUFID is allocated
- (7) THEN allocate reassembly resources with BUFID;  
TIMER <- TLB; TDL <- 0;
- (8) put data from fragment into data buffer with  
BUFID from octet FO\*8 to octet  
(TL-(IHL\*4))+FO\*8;



# Procedura di aggregazione

- (9) set RCVBT bits from FO to  $FO + ((TL - (IHL * 4) + 7) / 8)$ ;
- (10) IF MF = 0 THEN  $TDL \leftarrow TL - (IHL * 4) + (FO * 8)$
- (11) IF FO = 0 THEN put header in header buffer
- (12) IF TDL  $\neq$  0
- (13)     AND all RCVBT bits from 0 to  $(TDL + 7) / 8$  are set
- (14)     THEN  $TL \leftarrow TDL + (IHL * 4)$
- (15)     Submit datagram to next step;
- (16)     free all reassembly resources for this BUFID;
- DONE.
- (17)  $TIMER \leftarrow \text{MAX}(TIMER, TTL)$ ;
- (18) give up until next fragment or timer expires;
- (19) timer expires: flush all reassembly with this BUFID;
- DONE.



# Il protocollo IP

## L'indirizzamento

Architettura ISO/OSI

Architettura TCP/IP

⋮	⋮
Livello Trasporto	TCP/UDP
Livello Rete	Internet Protocol (IP)
Livello DataLink	Livello Fisico
Livello Fisico	



# Indirizzamento

- Per consentire a tutti i sistemi connessi a Internet di comunicare tra loro è necessario stabilire un metodo globalmente accettato per identificare ed indirizzare in modo *univoco* tutti i sistemi
- Ciò ha richiesto di definire un nuovo schema di indirizzamento, dato che le sotto-reti pre-esistenti avevano schemi di indirizzamento propri, diversi tra loro in generale, e quindi non univoci globalmente (es. indirizzi Ethernet, indirizzi X.25, numeri telefonici etc.)

# Indirizzamento



Lo schema di indirizzamento in Internet è definito all'interno degli strati IP e TCP/UDP

- Un indirizzo IP identifica un sistema (host o router) e non uno specifico utente (un generico sistema può essere usato da più utenti o processi)
- L'identificazione di un utente (in senso OSI) all'interno di un sistema è affidata ai protocolli di strato superiore (TCP o UDP)
- Un indirizzo completo è quindi composto da 2 parti, una definita nello strato IP e una nello strato TCP (o UDP)





# Indirizzamento IP

- Gli indirizzi devono essere unici in tutta la inter-rete (è possibile attribuire indirizzi arbitrari ad una rete TCP/IP solo se questa non è connessa con altre reti o con Internet)
- Lo schema di indirizzamento IP è stato progettato per facilitare le operazioni di instradamento, perciò ha una struttura gerarchica
  - ✕ un indirizzo IP identifica prima una porzione di inter-rete a cui un sistema è connesso e poi il sistema all'interno di quella porzione
  - ✕ si può fare un parallelo con il sistema di numerazione telefonica



# Indirizzamento IP

- Un indirizzo IP è costituito da una stringa di 32 bit
  - ✗ possono esistere teoricamente  $2^{32}$  (=4 294 967 296) possibili indirizzi
- L'IP address è formato da due parti:

$$\text{IP\_Address} = \text{Net\_Id}.\text{Host\_Id}$$

- ✗ **Net-ID:** identifica una porzione di inter-rete, che può coincidere con una sotto-rete fisica, o comprendere più sotto-reti fisiche o essere un sottoinsieme di una sotto-rete fisica; nel seguito indicheremo la porzione individuata da Net-ID col nome di “rete logica”
  - ✗ **Host-ID:** identifica il sistema (host o router) nella rete logica
- La divisione dei bit tra Net\_Id e Host\_Id non è fissa



# Indirizzamento IP

notazione  
“numerica”

10100100 01100001 00011001 01110010

160.97.25.114

notazione “puntata”  
(dotted) o decimale

- L'indirizzo di 32 bit (quattro byte) può essere espresso scrivendo i valori decimali di ciascun byte separati dal carattere “punto”
  - ✗ il passaggio da notazione dotted a indirizzo di 32 bit è banale in quanto implica una semplice conversione decimale-binario



# Indirizzamento IP

Classe	Bit Iniziali	Net_Id	Host_Id	“reti logiche” disponibili	“indirizzi” disponibili
A	0	7 bit	24 bit	128	16 777 216
B	10	14 bit	16 bit	16 384	65 536
C	110	21 bit	8 bit	2 097 152	256
D	1110	indirizzi multicast: 28 bit indirizzi possibili: $2^{32}/16=268\,435\,456$			
E	11110	riservati per usi futuri e ricerca: 27 bit indirizzi possibili: $2^{32}/32=134\,217\,728$			



# Indirizzi di classe A



- Esempio

15.10.10.90



Net ID

Host ID

- In notazione decimale il primo numero è compreso tra 0 e 127
- (0.0.0.0 – 127.255.255.255), 127.0.0.0 è riservato
- Numero totale di indirizzi di classe A =  $(2^{32})/2$

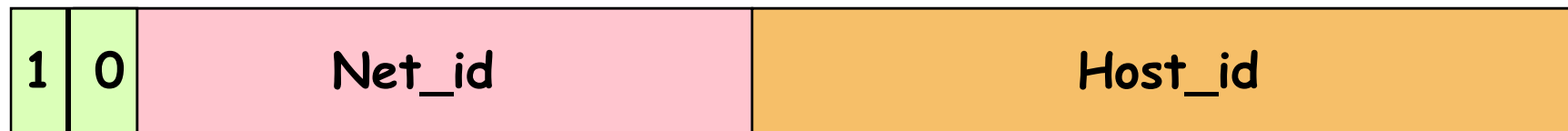




# Indirizzi di classe B

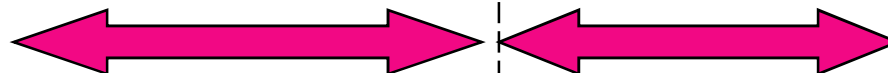
14 bit

16 bit



- Esempio

130.20.18.62



Net ID

Host ID

- In notazione decimale il primo numero è compreso tra 128 e 191
- (128.0.0.0 – 191.255.255.255)
- Numero totale di indirizzi di classe B =  $(2^{32})/4$



# Indirizzi di classe C



- Esempio

195.31.235.10



Net ID

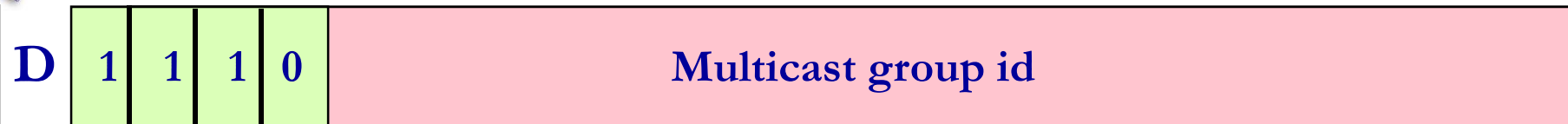
Host ID

- In notazione decimale il primo numero è compreso tra 192 e 223
- (192.0.0.0 – 223.255.255.255)
- Numero totale di indirizzi di classe C =  $(2^{32})/8$



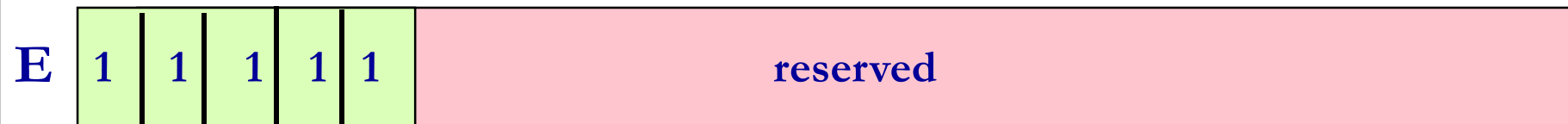
# Indirizzi di classe D ed E

28 bit



- (224.0.0.0 – 239.255.255.255)
- Un gruppo multicast corrisponde a una pluralità di host; si usa un protocollo aggiuntivo IGMP (Internet Group Management Protocol)
- Numero totale di indirizzi di classe D =  $(2^{32})/16$

27 bit



- (240.0.0.0 – 255.255.255.254)
- Numero totale di indirizzi di classe E =  $(2^{32})/32$



# Indirizzi IP speciali

Tutti "0"		Questo host <sup>1</sup>
Tutti "0"	Host_ID	Questa rete a cui appartiene Host <sup>1</sup>
Tutti "1"		Tutti gli host (limited broadcast) <sup>2</sup>
Net_ID	Tutti "1"	Tutti gli host della rete Net (directed broadcast) <sup>2</sup>
127 <sup>4</sup>	Anything (spesso 1)	Loopback <sup>3</sup> o rinvio a sé stesso

<sup>1</sup> Permessso solo al bootstrap ed è usabile solo come indirizzo sorgente

<sup>2</sup> Può essere usato solo come indirizzo destinazione

<sup>3</sup> Non deve essere propagato dai router sulla rete

<sup>4</sup> Notazione binaria: 01111111

# Indirizzi IP speciali



## Esempi:

- 255.255.255.255 indirizza tutti gli host di Internet
- 160.97.255.255 indirizza tutti gli host della rete logica
- 120.255.255.255 indirizza tutti gli host della rete 120
- 0.0.0.0 si riferisce all'host che origina la richiesta
- 0.0.0.95 si riferisce alla rete logica a cui e' connesso l'host 95



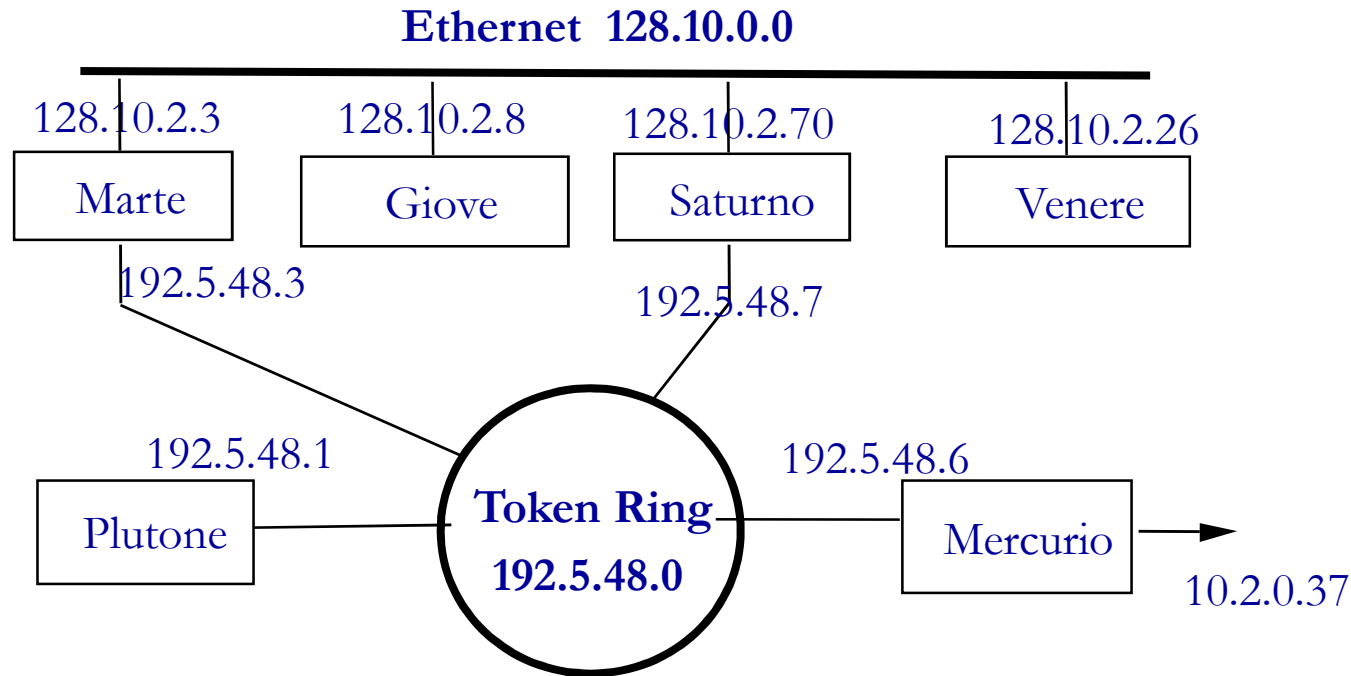


# Router e indirizzi IP

- Occorre evidenziare che **non sono i nodi** ad avere un indirizzo IP, bensì le interfacce verso le reti. Quindi se un nodo ha  $N$  interfacce (ad es. un router), esso ha  $N$  indirizzi IP (oltre ai corrispondenti indirizzi locali)
- Inoltre, l'indirizzo IP **non identifica l'host** in quanto tale, ma la connessione di un host alla relativa rete. Di conseguenza, se un host viene spostato in un'altra rete IP, il suo indirizzo deve essere cambiato
- Vediamo un esempio in cui le reti logiche coincidono con le sotto-reti fisiche



# Router IP



- “Marte” e “Saturno” sono router connessi sia alla Token Ring che a Ethernet; hanno quindi due indirizzi.
- Il router “Mercurio” interconnette la Token Ring a un'altra rete (ARPANET) e quindi anch'esso ha due indirizzi .
- Siccome gli indirizzi sono utilizzati per l'instradamento, se ci riferiamo a “Marte” con l'indirizzo 192.5.48.3 il messaggio arriva tramite la Token Ring; se invece usiamo 128.10.2.3, il messaggio arriva tramite l'Ethernet.



# Assegnazione degli Indirizzi

- Al fine di assicurare l'univocità degli indirizzi di Internet è stata costituita un'autorità centrale con il compito di assegnare gli indirizzi, l'INTERnet Network Information Center (INTER-NIC)
- L'INTER-NIC è controllato a sua volta dall'autorità amministrativa IANA
- INTER-NIC assegna solo la parte Net\_Id, delegando l'organizzazione richiedente per l'assegnazione della restante parte dell'indirizzo, l'Host\_Id
- Così un utente che ha bisogno di un indirizzo IP non deve rivolgersi ad INTER-NIC, ma può fare riferimento ad un'autorità locale
- L'organizzazione che si vede assegnare una Net\_Id potrà poi suddividere la parte di Host\_Id per creare al suo interno delle sotto-reti logiche, ognuna delle quali avrà la stessa Net\_Id; il processo può essere iterativo

# Subnetting



• La struttura di indirizzamento a due livelli gerarchici era sufficiente nella fase iniziale di Internet

- ✗ Internet non ha una topologia gerarchica, ma l'interpretazione degli indirizzi è gerarchica

- Alcune organizzazioni (università, compagnie con estensioni vaste) spesso devono gestire più di una sotto-rete fisica e non conviene richiedere indirizzi diversi per ogni sotto-rete

- ✗ esplosione nelle dimensioni delle tabelle di routing

- Al crescere delle dimensioni di Internet, si decise di rendere più flessibile l'indirizzamento aggiungendo (RFC 950, nel 1985) un terzo livello gerarchico: il livello di sotto-rete (Subnet)

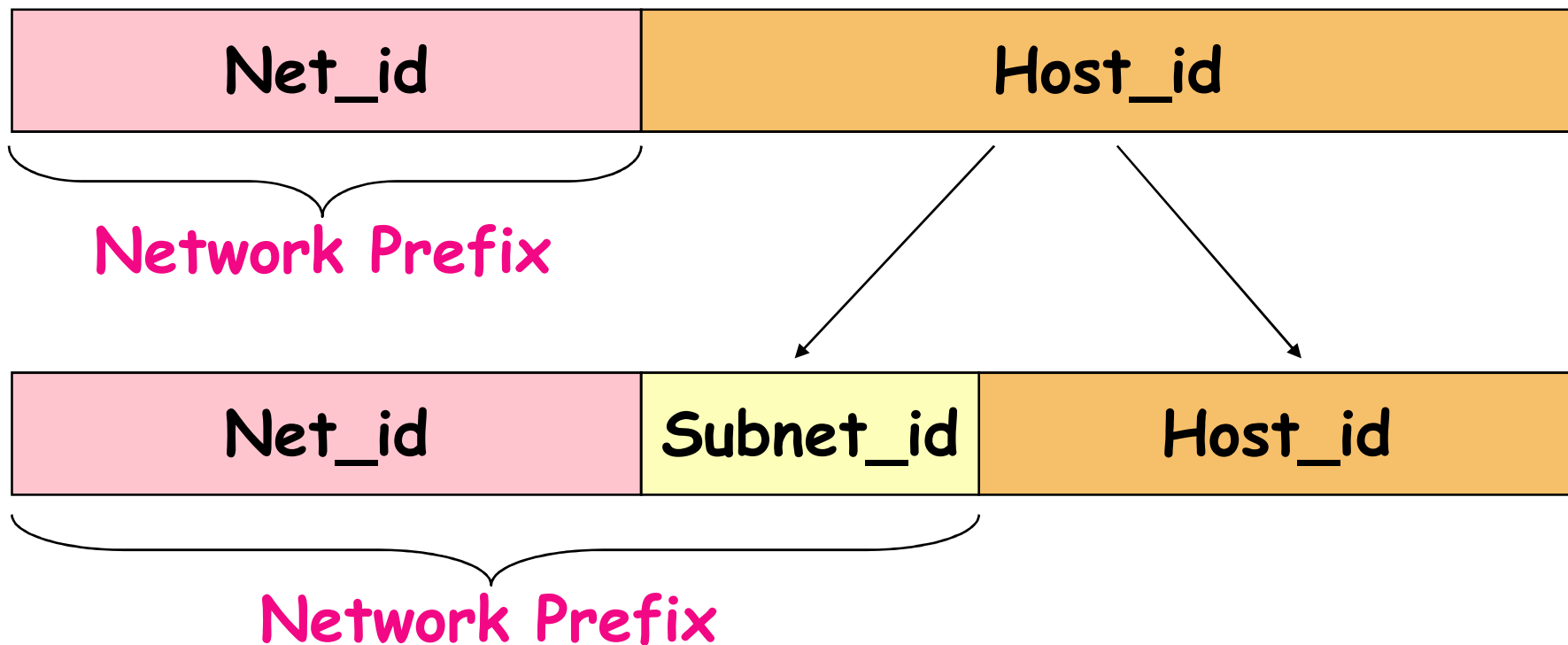
# Subnetting



- Il subnetting è la divisione di una singola rete di classe A, B o C in sottoreti di dimensioni minori
- Si utilizzano alcuni bit dell'Host\_Id per codificare Subnet\_Id
- Un generico indirizzo IP è quindi costituito da:

$\text{IPaddress} = \text{Net\_id}.\text{Host\_id} = \text{Net\_id} + \text{Subnet\_id} + \text{Host\_id}$

# Subnetting





- Il campo Subnet\_Id è identificato da una maschera di bit denominata “Subnet Mask”
- Una Subnet Mask è una parola di 32 bit in cui
  - i bit “1” identificano i bit del Net\_Id e del Subnet\_Id
  - i bit “0” identificano i bit dell’Host\_Id

21 bit

8 bit

Classe C

1

1

0

Network

Subnet

Host

193

205

102

36

1

1

0

0

0

0

0

1

1

1

0

0

1

1

0

1

1

0

0

1

1

1

1

0

0

0

1

0

0

1

0

0

Subnet Mask

255

255

255

248

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

0

0

0



# Subnetting: vantaggi



- Il subnetting risolve il problema della continua richiesta di indirizzi IP, assegnando ad ogni organizzazione uno (o alcuni) indirizzi di rete
  - ✗ l'amministratore della rete logica è libero di decidere come frazionare il campo Host\_id assegnare diversi Subnet\_id a ciascuna delle sue sotto-reti logiche interne
- Una sotto-rete logica può coincidere con una sotto-rete fisica o una parte di una sotto-rete fisica, ma non può includere più sotto-reti fisiche

# Subnetting: vantaggi



- Il subnetting riduce le dimensioni delle tabelle di routing facendo in modo che le sottoreti di una rete logica non siano visibili all'esterno della rete stessa
- Il Subnet\_Id è preso in esame solo quando il datagramma arriva nella rete logica di destinazione
  - ✗ solo i router locali all'interno della rete logica conoscono e distinguono le singole sottoreti e tengono traccia degli host nella propria sotto-rete
  - ✗ i router esterni ad una rete logica si basano solo sulla componente Net\_id per instradare i datagrammi (hanno un'unica entry nella tabella di routing che individua tutte le sottoreti logiche di una rete logica) e non devono conoscere la collocazione di tutti gli host delle altre sotto-reti



# Subnetting: tabelle di routing

- Ogni router effettua un **AND** bit a bit tra l'indirizzo IP di destinazione e la sua subnet mask e ottiene l'indirizzo della subnet a cui è collegato l'host
- Esempio:
  - ✗ un datagramma indirizzato a 130.50.15.6 arriva a un router della subnet 5 di una rete di classe B (130.50.0.0) con subnet mask 11111111.11111111.11111000.00000000 (255.255.252.0)
  - ✗ il router effettua l'**AND** logico tra l'indirizzo IP e la mask e ottiene l'indirizzo 130.50.12.0 che individua la quarta subnet.



# Netmask: valori leciti

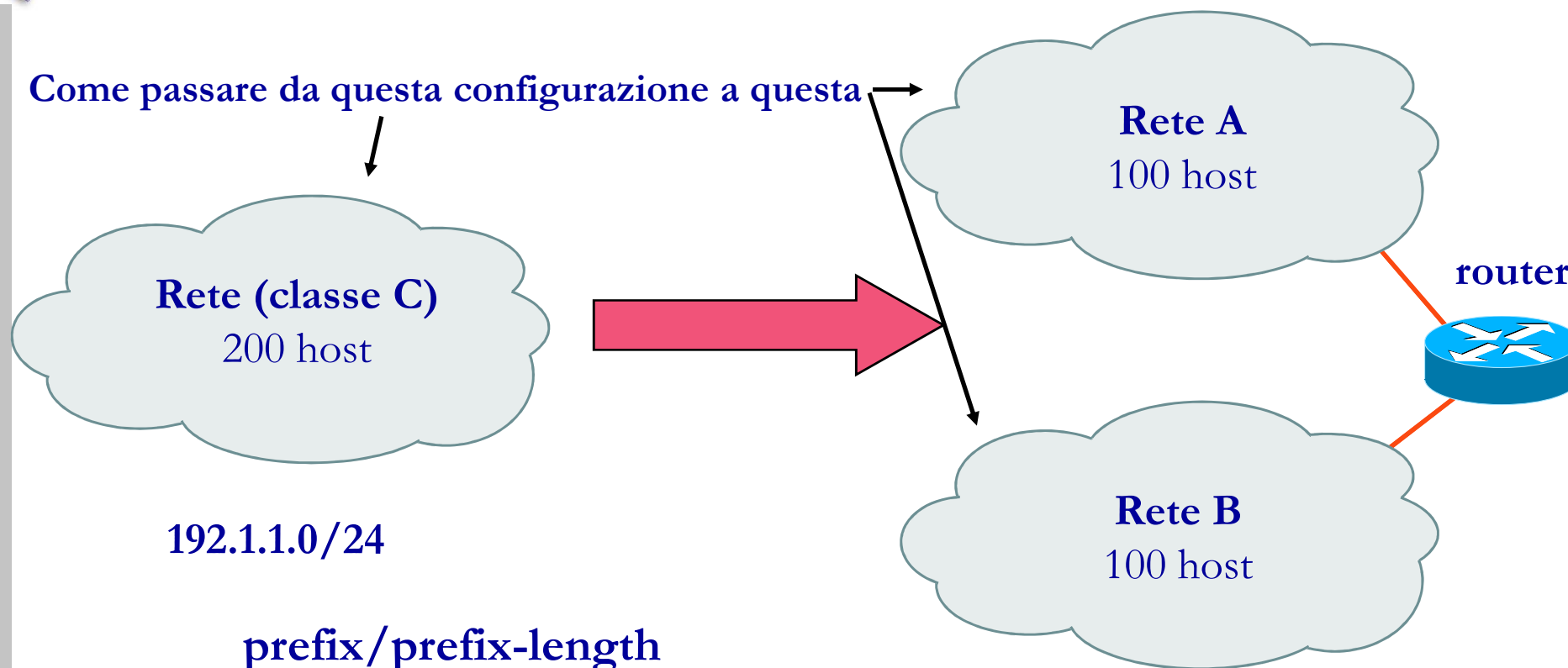
- I valori decimali leciti nei 4 byte che costituiscono la netmask sono:

128	1000 0000	(128)
192	1100 0000	(64)
224	1110 0000	(32)
240	1111 0000	(16)
248	1111 1000	(8)
252	1111 1100	(4)
254	1111 1110	(2)
255	1111 1111	(1)



# Subnetting: esempio

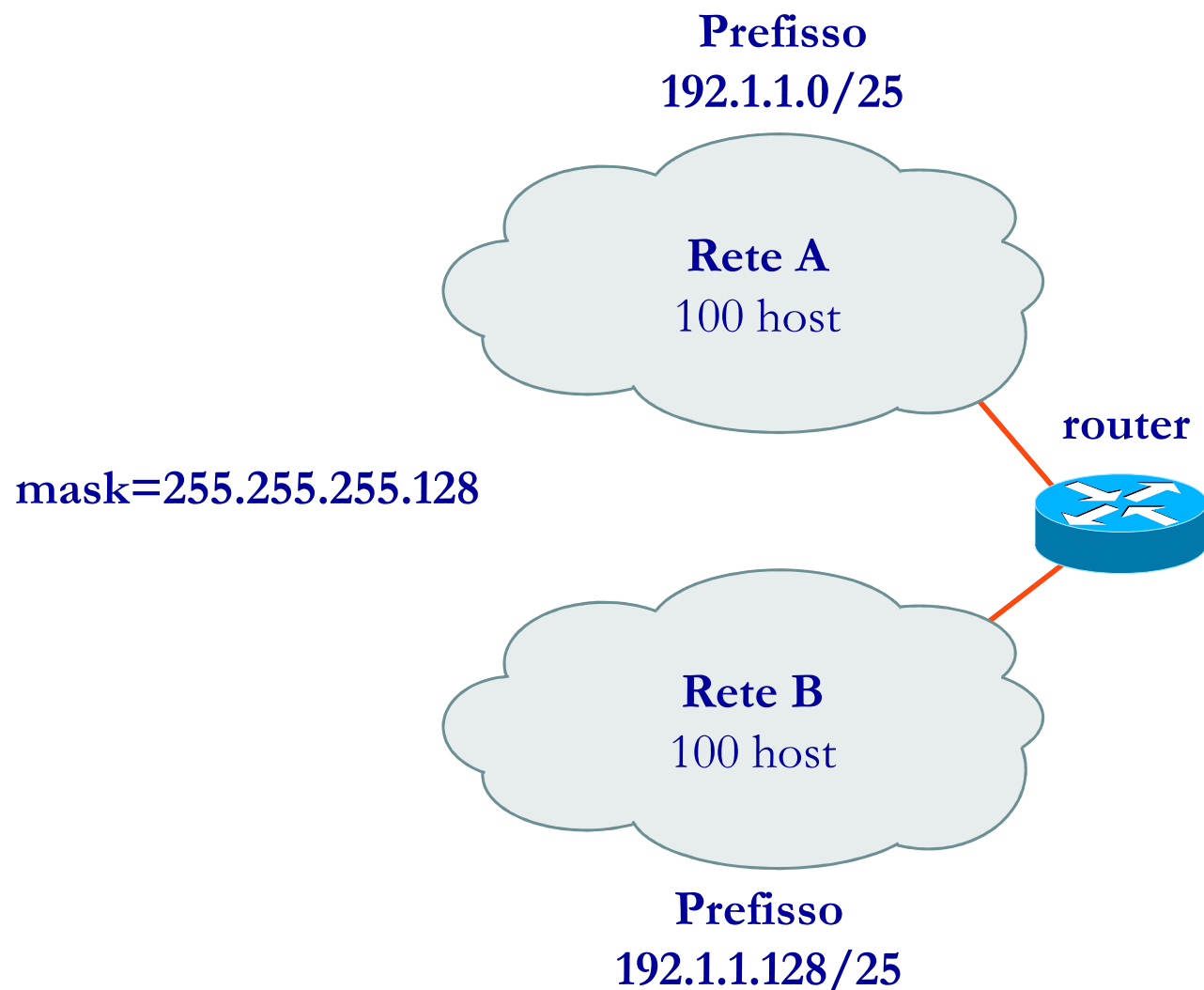
Come passare da questa configurazione a questa



prefix-length è un valore decimale che specifica quanti dei bit contigui più significativi specificano il prefisso (net+subnet)



# Subnetting: esempio





# Subnetting: esempio

**Prefisso**  
**192.1.1.0/25**



**Prefisso**  
**192.1.1.128/25**

router



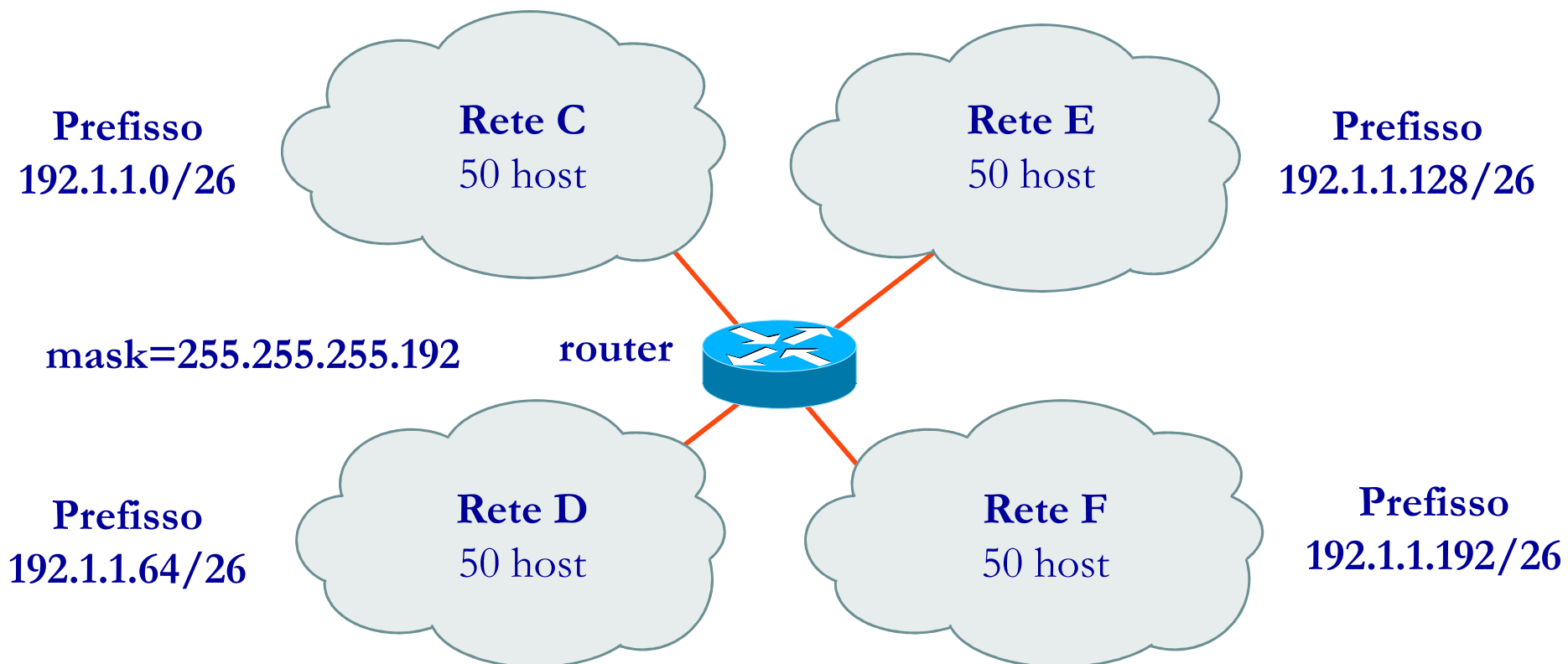
router







# Subnetting: esempio



# Subnet mask

Come fa un host/router a conoscere la subnet mask usata dalla sottorete a cui è connesso?

- Il protocollo ICMP include due messaggi ad hoc:
- Address Mask Request
  - ✗ Un host/router, al boot, manda in broadcast un messaggio di Address Mask Request
- Address Mask Reply
  - ✗ Un router che riceve la richiesta, riempie il campo subnet mask del messaggio ICMP con le informazioni della sotto-rete dalla quale è pervenuta la richiesta
- Un router connesso a più sottoreti deve conoscere la subnet mask di ciascuna di esse



# Address Mask Request/Reply

## Esempio:

- L'host richiedente è connesso a una rete di classe B 128.99.0.0 e ha indirizzo IP 128.99.4.123
- L'indirizzo IP del router connesso alla rete di classe B è 128.99.4.62 ed usa un campo Subnet di 6 bit, cioè una subnet mask 255.255.252.0
- L'host invia un Address Mask Request in broadcast, cioè usa i parametri:
  - × Source address: 128.99.4.123
  - × Destination address: 255.255.255.255
  - × Protocol: ICMP=1
  - × Type: Address Mask Request (AM1)
  - × Code: 0
  - × Mask: 0



# Address Mask Request/Reply

## Esempio (continua):

- Il router risponde con un Address Mask Reply, cioè usa i parametri:
  - × Source address: 128.99.4.62
  - × Destination address: 128.99.4.123
  - × Protocol: ICMP=1
  - × Type: Address Mask Reply (AM2)
  - × Code: 0
  - × Mask: 255.255.252.0
- Soluzioni alternative
  - × BOOTP, al boot la workstation diskless acquisisce tutte le informazioni utili (IP address, subnet mask, etc.)



# Subnet: esercizi

## Quesito B

Si consideri un router IP con indirizzo di classe B a cui sono connesse 4 sottoreti A, B, C, D, ciascuna comprendente il seguente numero di host:  $N_A=12$ ,  $N_B=9$ ,  $N_C=18$ ,  $N_D=6$ .

Si chiede di a) determinare la subnet mask che possa essere utilizzata per individuare le 4 sottoreti e che minimizzi il numero di bit utilizzati per indirizzare gli host di ciascuna sottorete; b) determinare il numero di sottoreti indirizzabili con la maschera definita al punto precedente.

## Soluzione

a) Per indirizzare 4 sottoreti sono sufficienti 2 bit, ma per minimizzare i bit dell'host-id di ciascuna bastano 5 bit; la subnet mask è 255.255.255.224

b)  $2^{11}=2048$