



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Internet e IP

Franco CALLEGATI

Dipartimento di Informatica: Scienza e Ingegneria

A.A. 2018-2019



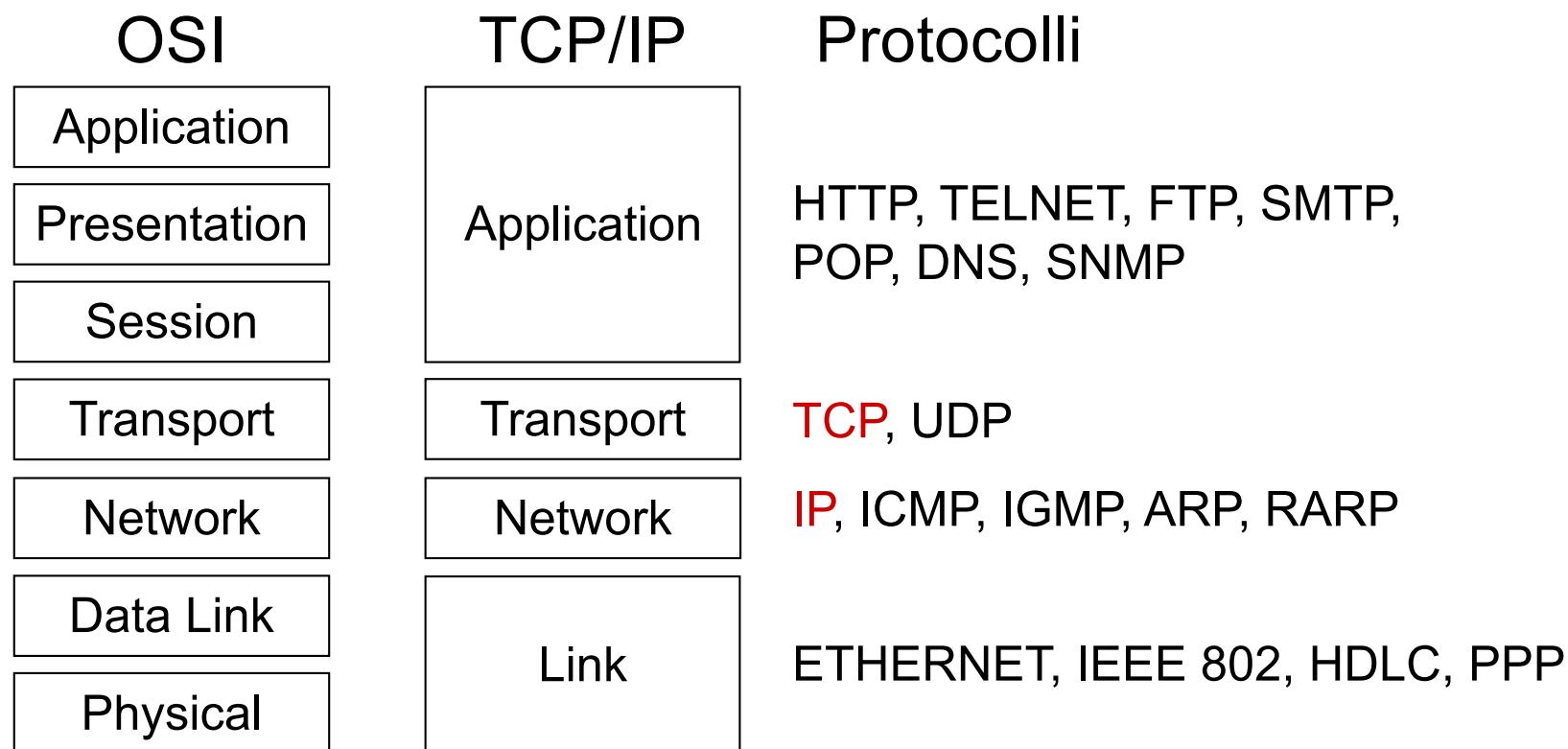
ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

I protocollli di Internet



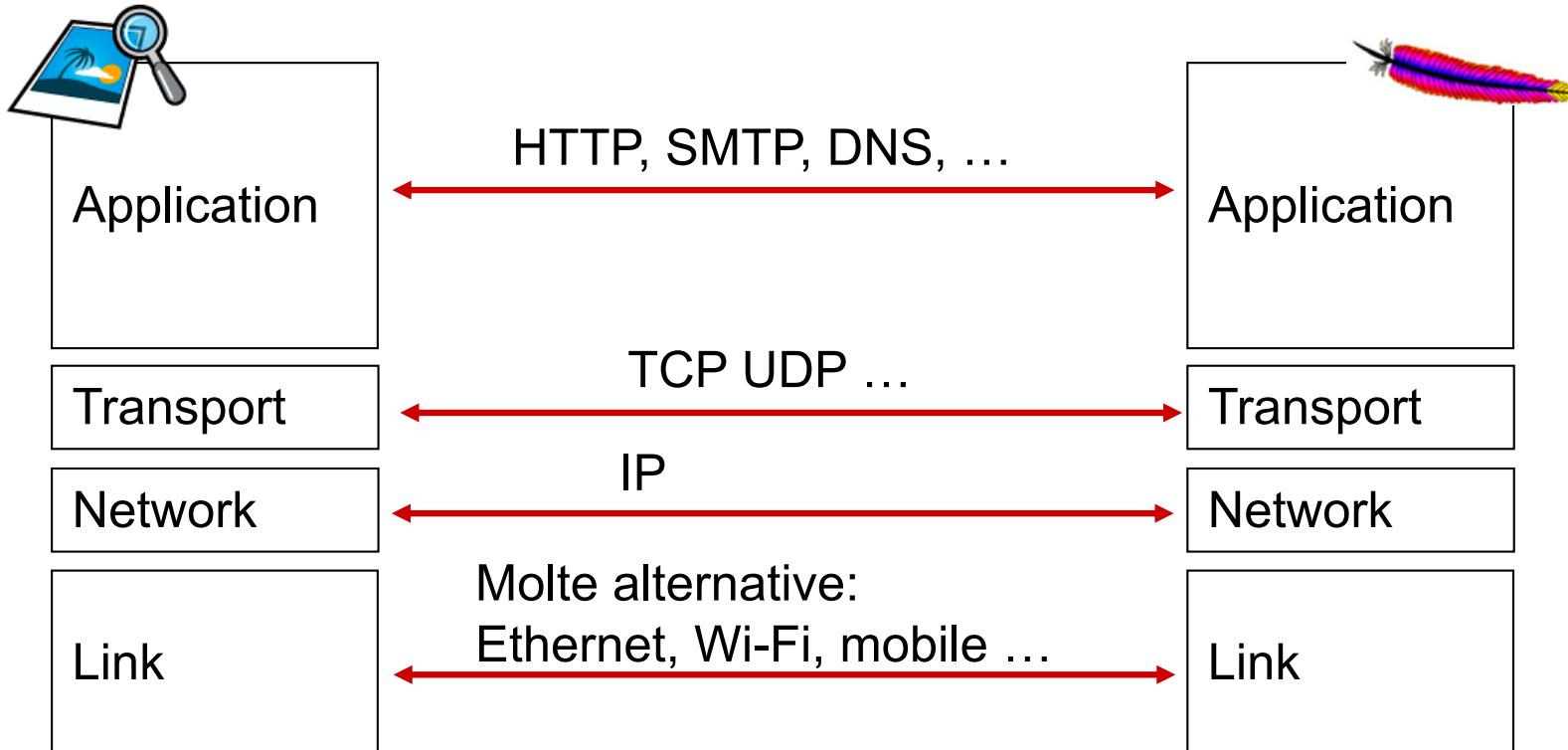
Modello a strati di Internet

- Modello a strati simile a ISO-OSI
- Solo 4 strati, l'applicazione integra presentazione e sessione
- Data link e strato fisico sono visti come un'unica entità



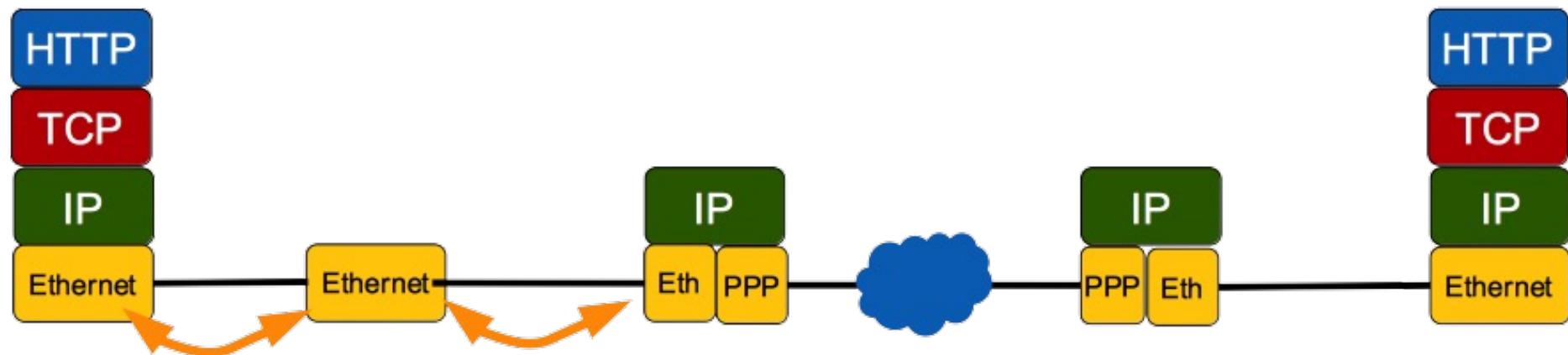
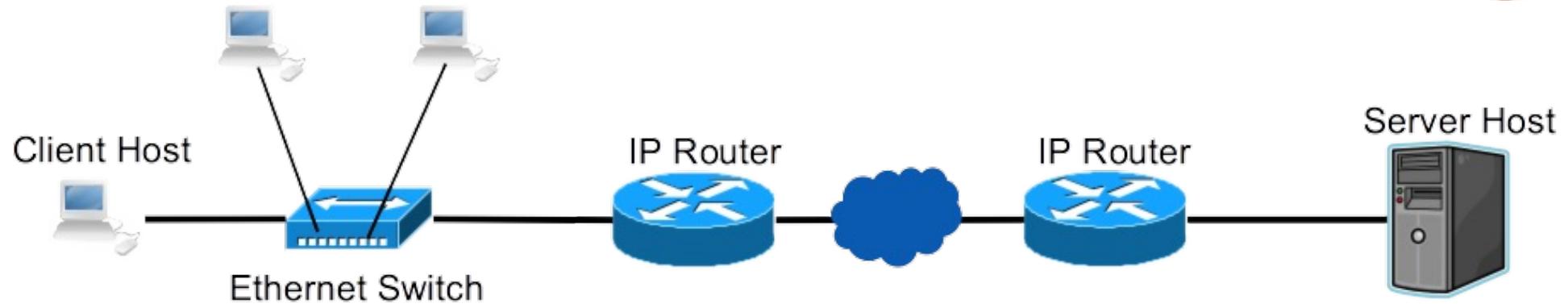


Il modello a strati di Internet





Data Link



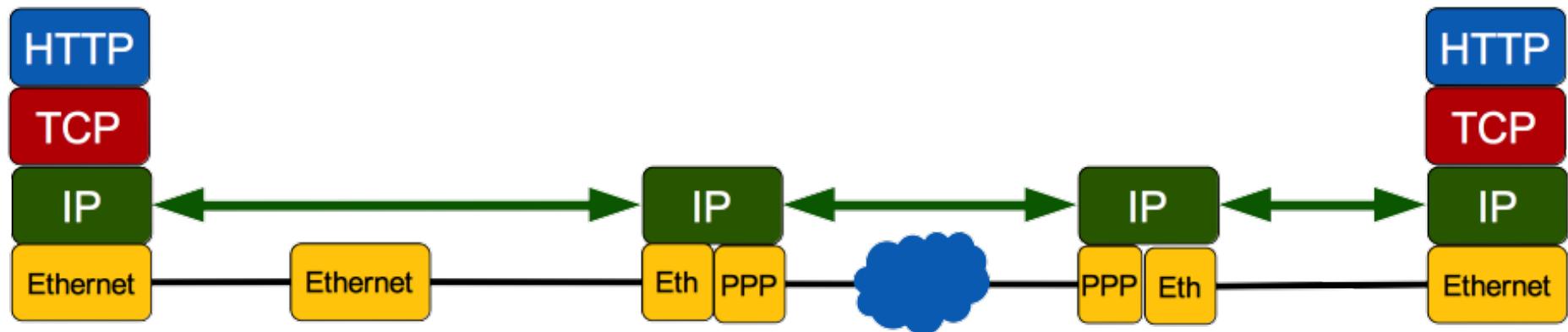
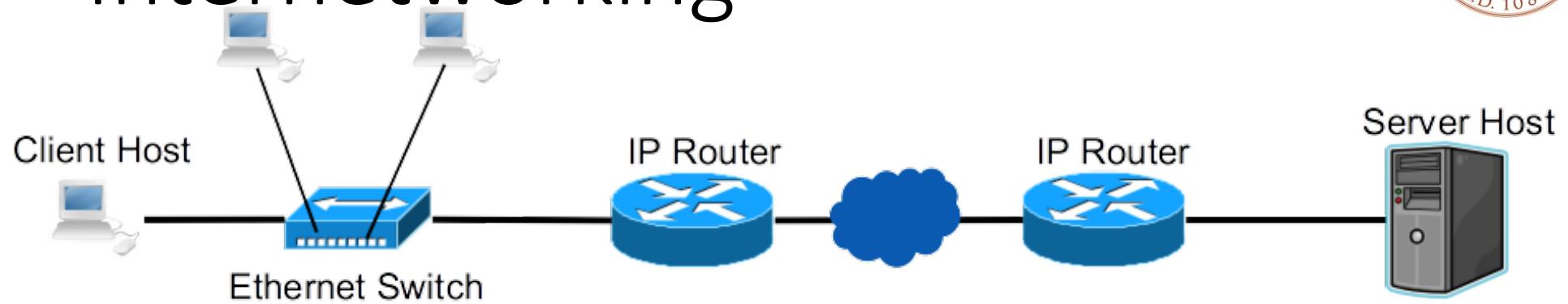
Obiettivo: connettività locale

Problema tipico: canale condiviso, tutti ricevono tutto

Obiettivo: coinvolgere lo strato IP solo se necessario



Internetworking



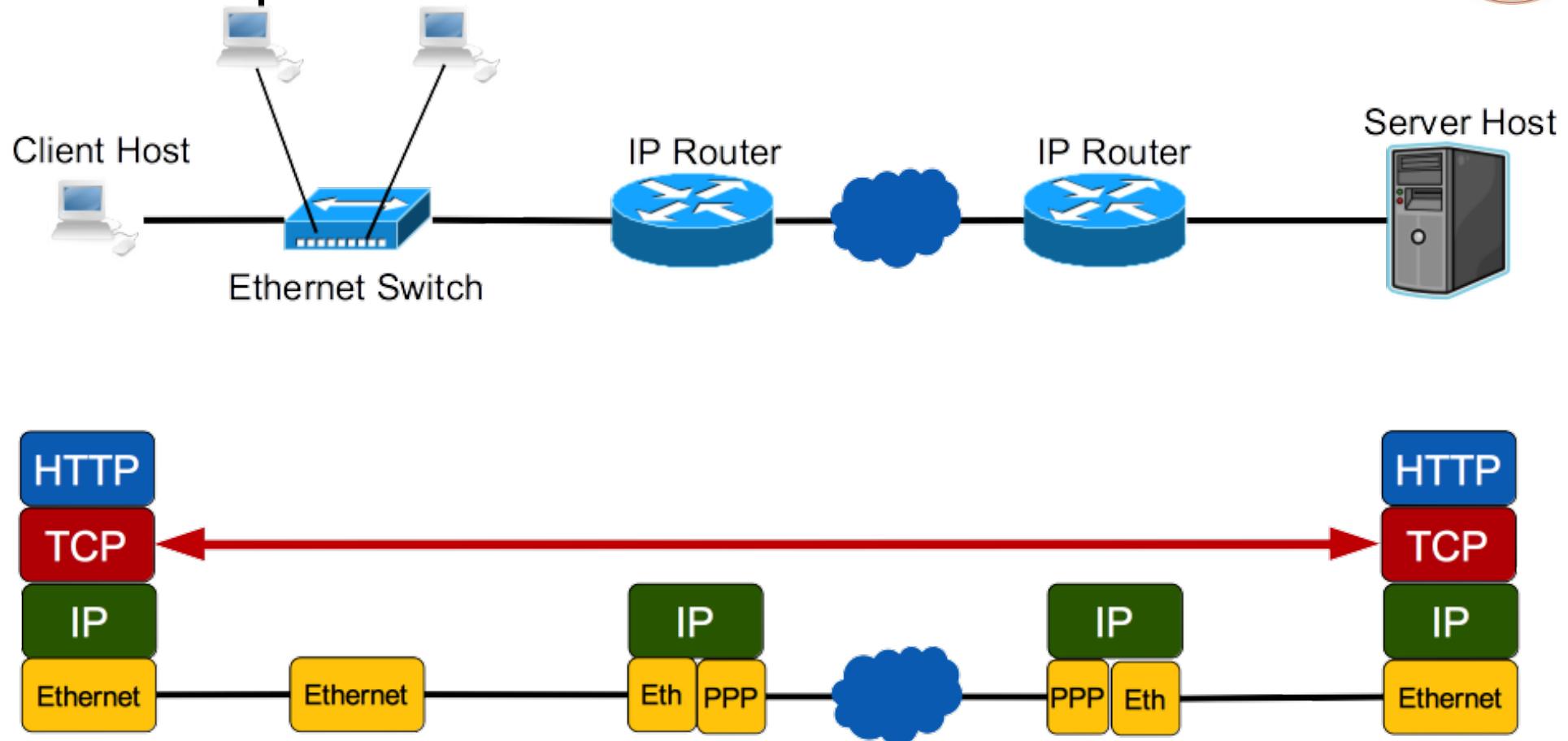
Obiettivo: connettività globale

Creare ponti fra le varie tecnologie locali

Indirizzo per indicare la destinazione finale a prescindere dai passi intermedi



Trasporto end-to-end



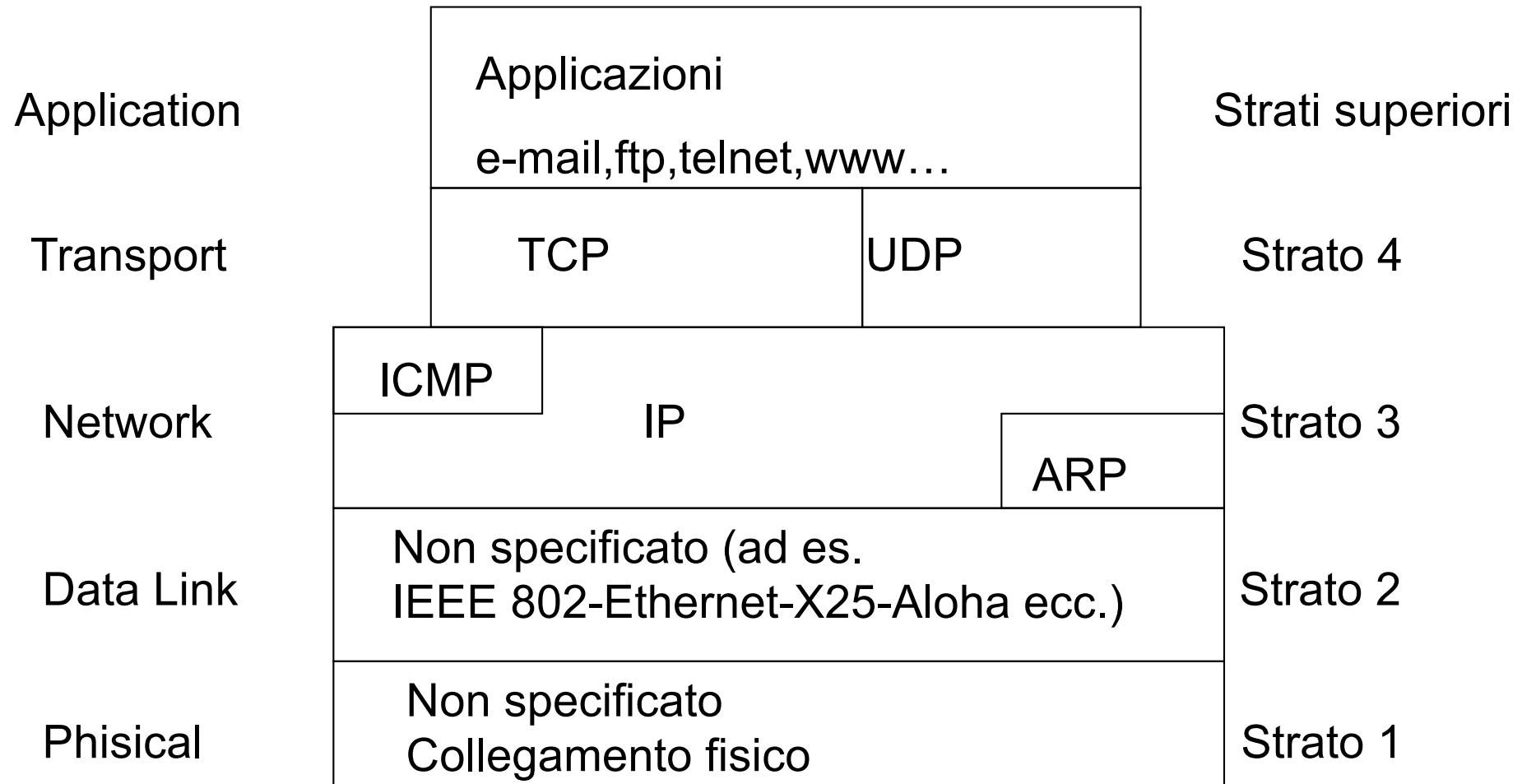
Obiettivo: garantire il dialogo affidabile fra le applicazioni finali

Protocollo end-to-end

Indirizzamento per identificare la singola applicazione una volta
raggiunto il calcolatore

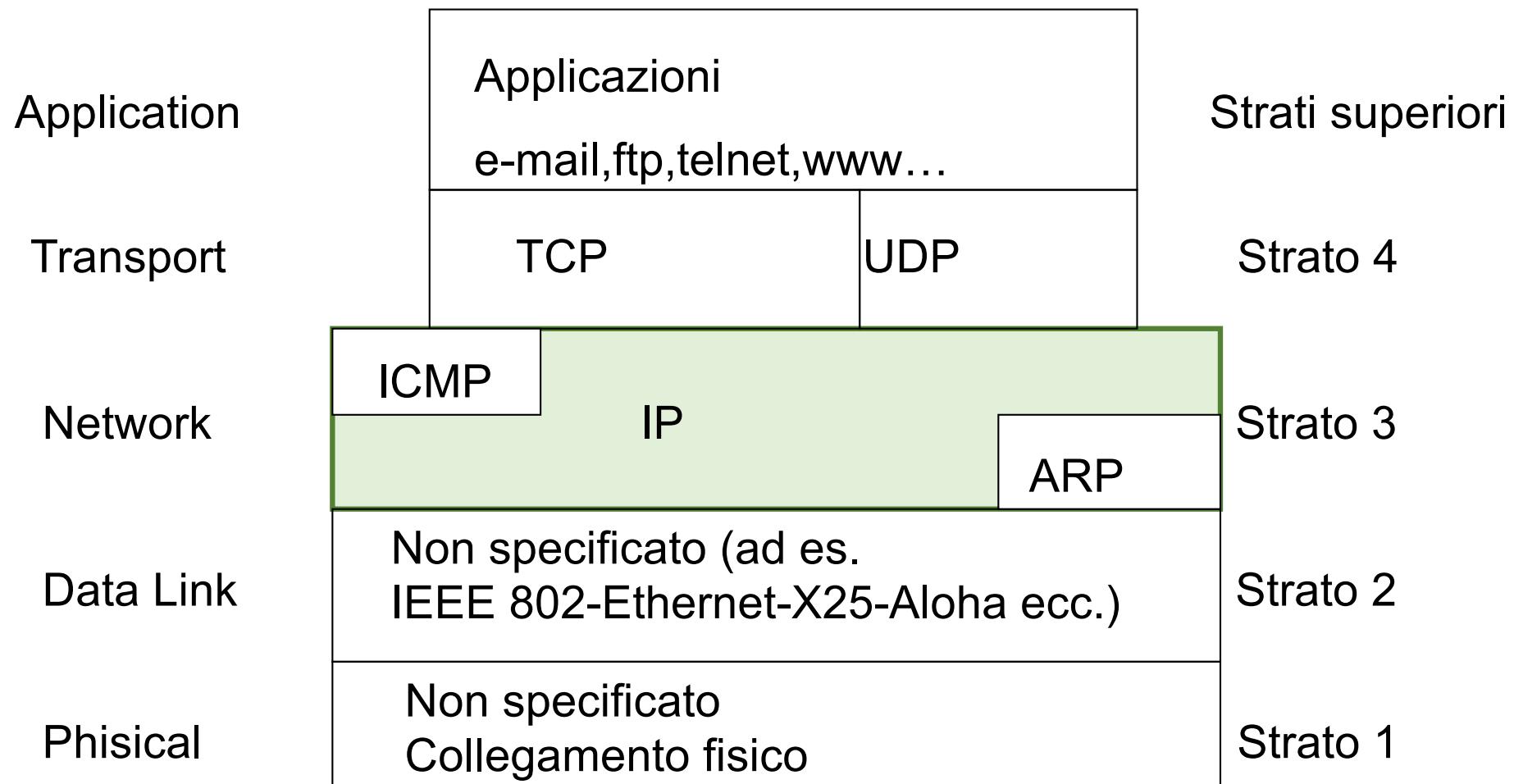


Architettura





Architettura





Internet Protocol (IP) - RFC 791

- Progettato per funzionare a **commutazione di pacchetto** in modalità **connectionless**
- Si prende carico della trasmissione di pacchetti detti **datagrammi** da sorgente a destinazione, attraverso reti eterogenee
- Identifica **host** e **router** tramite indirizzi di **lunghezza fissa**, raggruppandoli in **reti IP**
- **Frammenta** e **riassembla** i datagrammi quando necessario
- Offre un servizio di tipo **best effort**, cioè non sono previsti meccanismi per
 - aumentare l'affidabilità del collegamento end-to-end,
 - eseguire il controllo di flusso e della sequenza (no ARQ).



Struttura degli indirizzi IP

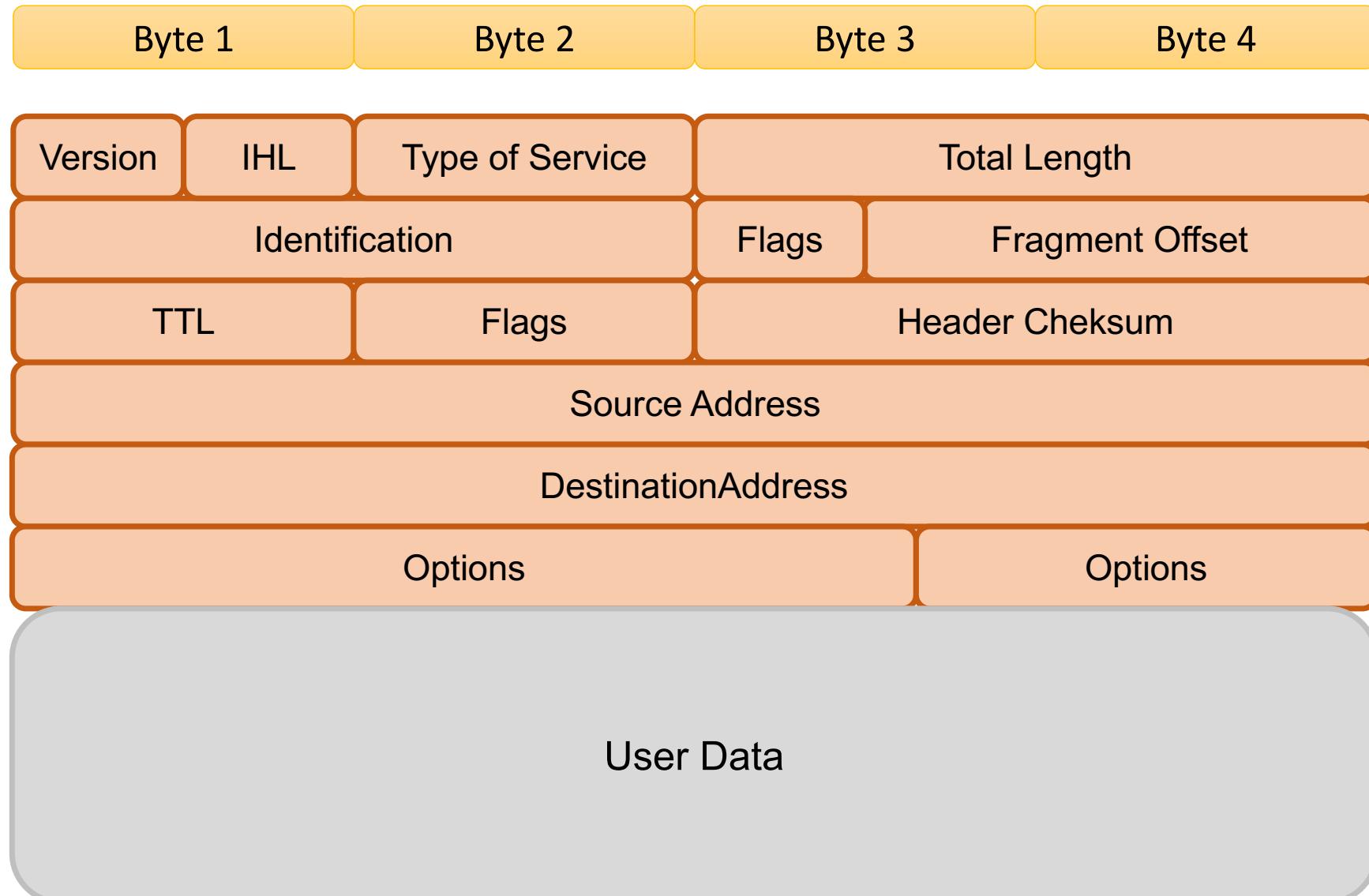
- Indirizzi di lunghezza fissa pari a **32 bit**
- Scritti convenzionalmente come sequenza di 4 numeri decimali, con valori da **0** a **255**, separati da punto (rappresentazione **dotted decimal**)

10001001.11001100.11010100.00000001
137.204.212.1

- Numero teorico max. di indirizzi
 $2^{32} = 4.294.967.296$
 - In realtà si riesce a sfruttare un numero molto inferiore
- Assegnati dalla **IANA** (Internet Assigned Numbers Authority)



Formato del pacchetto





Significato delle PCI (1)

- **Version** : indica il formato dell' intestazione, attualmente la versione in uso è la 4
- **IHL** : lunghezza dell' intestazione, espressa in parole di 32 bit; lunghezza minima = 5
- **Type of service** : indicazione sul tipo di servizio richiesto, usato anche come sorta di priorità
- **Total length** : lunghezza totale del datagramma, misurata in bytes; lunghezza massima = 65535 bytes, ma non è detto che tutte le implementazioni siano in grado di gestire questa dimensione



Significato delle PCI (2)

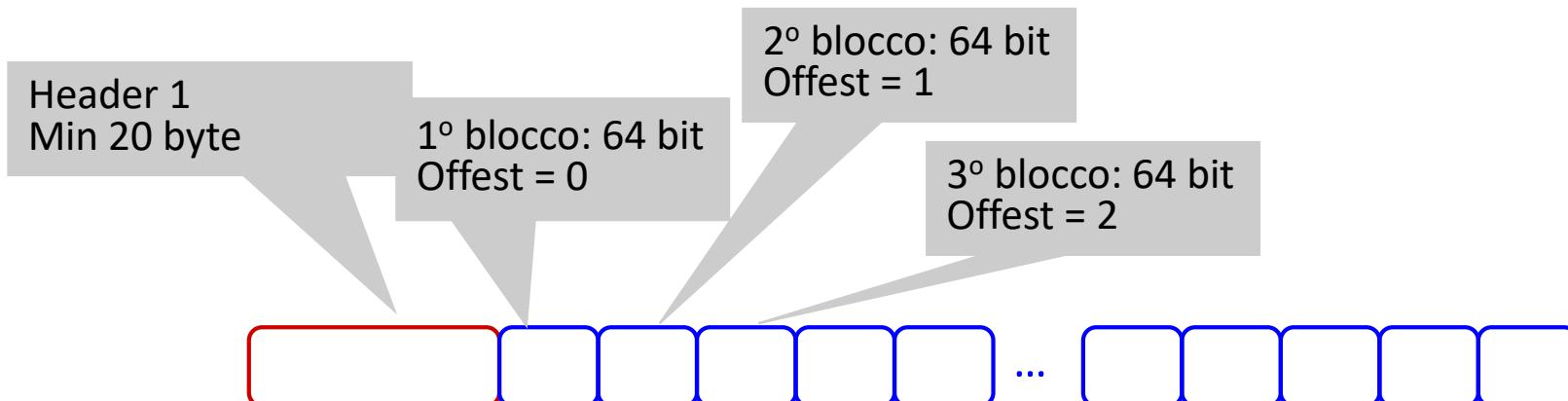
- **Identification** : valore intero che identifica univocamente il datagramma
 - Indica a quale datagramma appartenga un frammento (fragment)
- **Flag** :

bit 0	sempre a 0
bit 1	don't fragment (DF) DF = 0 si può frammentare DF = 1 non si può frammentare
bit 2	more fragments (MF) MF = 0 ultimo frammento MF = 1 frammento intermedio
- **Fragment offset**: indica quale è la posizione di questo frammento nel datagramma, come distanza in unità di 64 bit dall'inizio



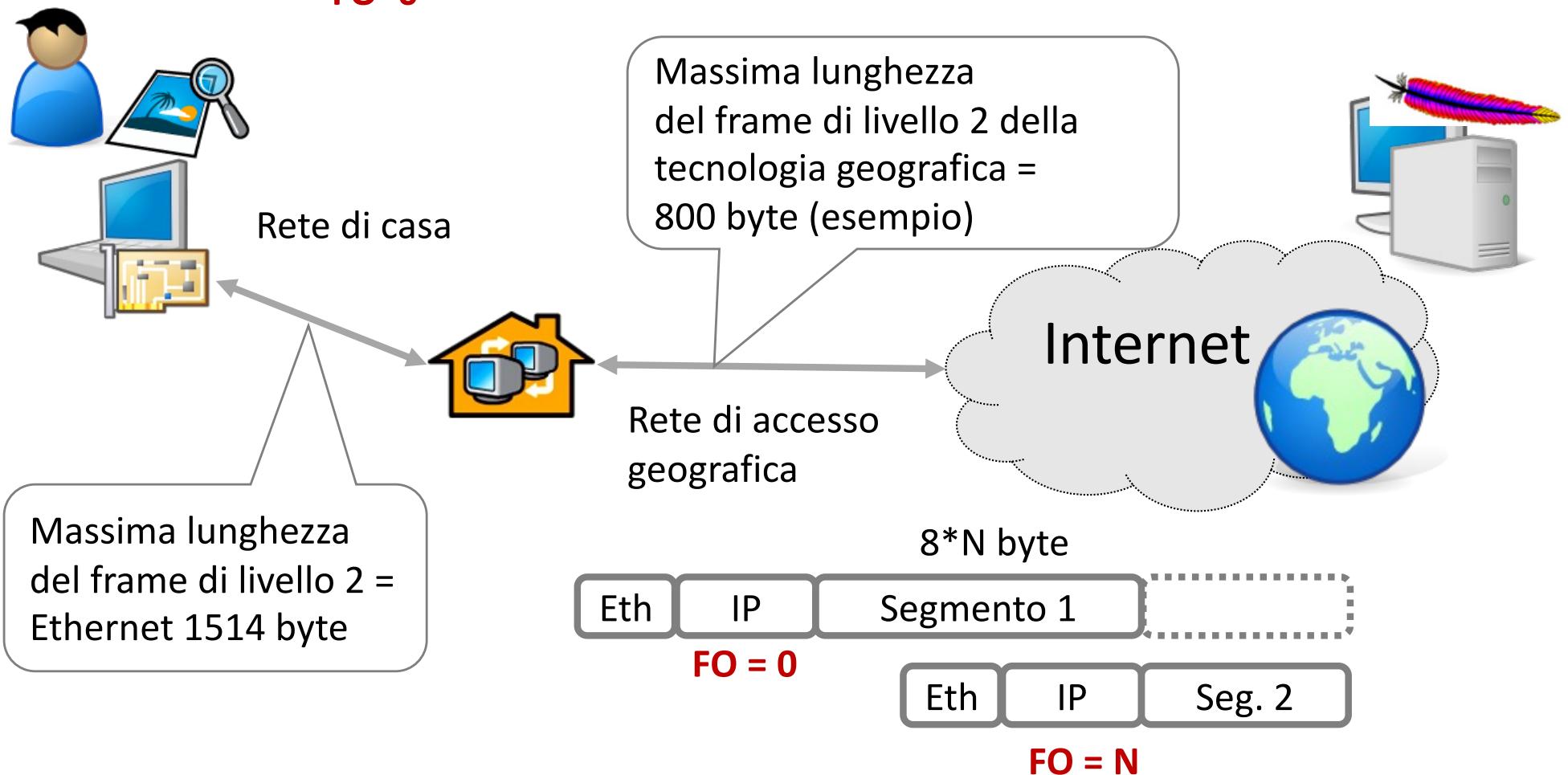
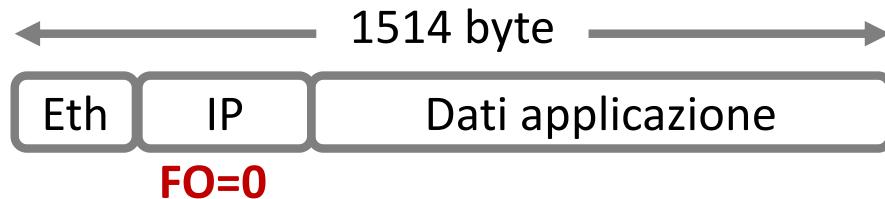
Fragment offset

- Il datagramma IP viene virtualmente suddiviso in sotto-blocchi di 8 byte (64 bit)
- Per chi trasmette (non necessariamente la sorgente dei dati ma anche un nodo intermedio)
 - Il primo blocco del datagramma è il numero 0
 - I blocchi successivi sono logicamente numerati sequenzialmente
- Il numero logico del primo blocco viene scritto nel **Fragment Offset** del datagramma





La segmentazione





Chi frammenta

- La dimensione massima del frame di livello 2 dipende dalle tecnologie
 - Tale dimensione viene tipicamente chiamata Maximum Transfer Unit (MTU) tipicamente misurata in byte
- In teoria qualunque nodo di rete connesso a tecnologie diverse di livello 2, può dover affrontare questo problema
- Pertanto **qualunque nodo di rete** dotato di protocollo IP deve potere e **può frammentare** un datagramma
- I nodi intermedi **non riassemmblano**, ma lo fa solamente il **terminale ricevente**

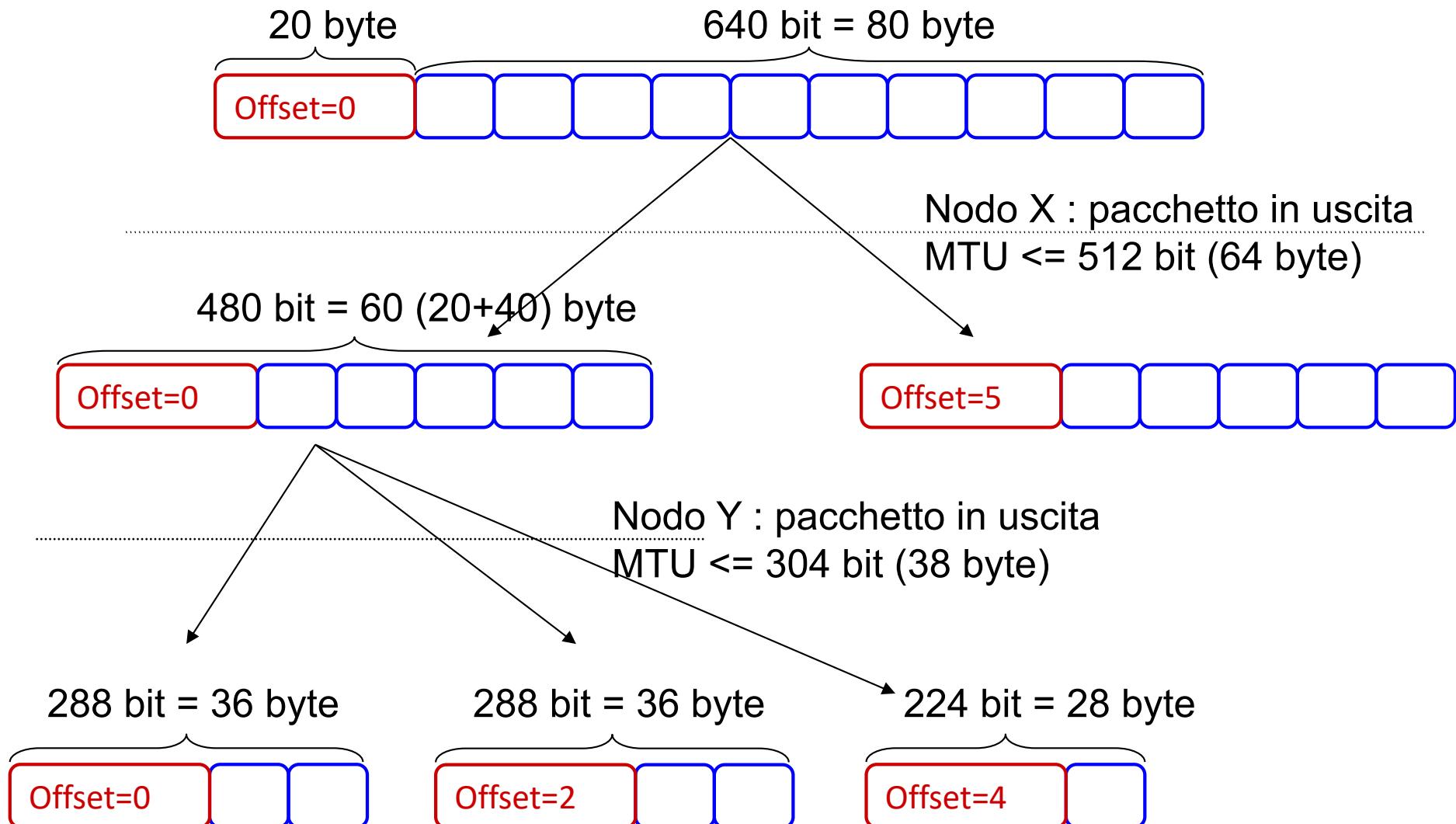


Frammentazioni multiple

- Un datagramma può essere frammentato a più riprese in nodi successivi
- La numerazione tramite “**offset**” è stata concepita per poter rinumerare facilmente **frammenti di un frammento**
- Un datagramma può essere duplicato dalla rete e le copie possono seguire percorsi diversi con frammentazioni diverse
- Possono essere ricevuti frammenti che si sovrappongono parzialmente

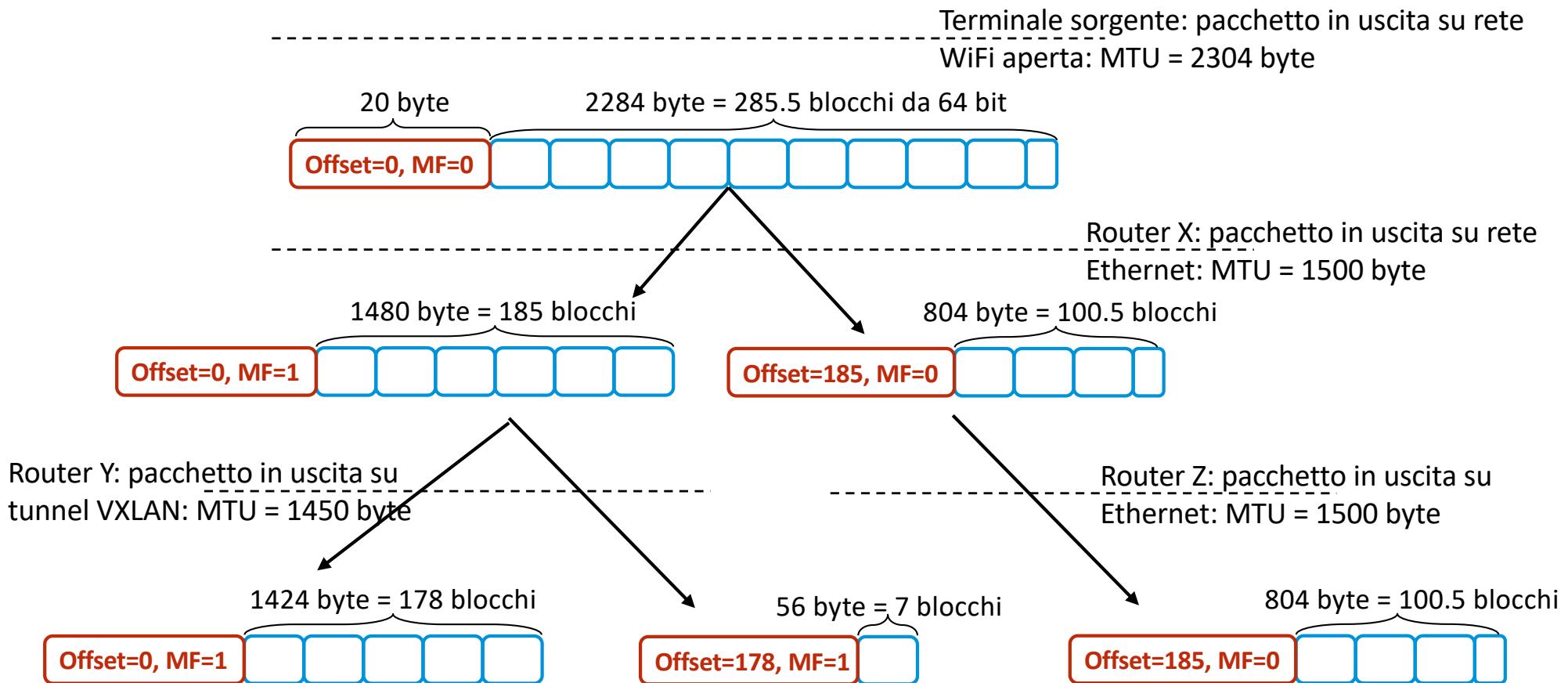


Esempio di calcolo dell'offset



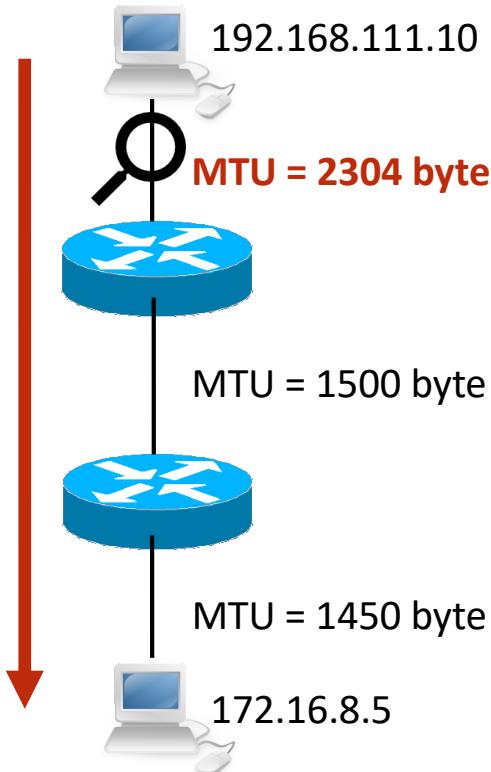


Un esempio più realistico





Primo collegamento: cattura

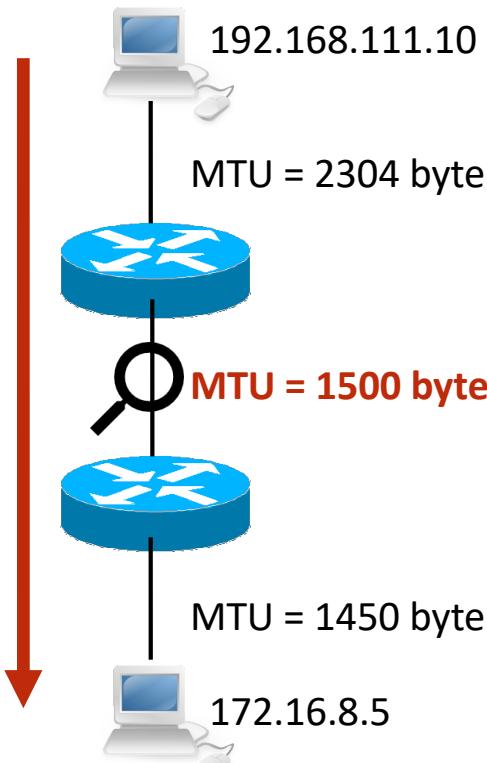


No.	Time	Source	Destination	Protocol	Info
1	0.0000...	192.168.111.10	172.16.8.5	IPv4	ICMP (1)
2	0.0012...	172.16.8.5	192.168.111.10	IPv4	ICMP (1)
3	0.0012...	172.16.8.5	192.168.111.10	IPv4	Fragmented IP protocol


```
Frame 1: 2318 bytes on wire (18544 bits), 2318 bytes captured (18544 b...
Ethernet II, Src: 26:4b:3d:c7:6a:0b (26:4b:3d:c7:6a:0b), Dst: 6e:3e:c2:...
Internet Protocol Version 4, Src: 192.168.111.10, Dst: 172.16.8.5
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 2304
    Identification: 0x2ae7 (10983)
    000. .... = Flags: 0x0
        0... .... = Reserved bit: Not set
        .0... .... = Don't fragment: Not set
        ..0. .... = More fragments: Not set
        ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: ICMP (1)
    Header Checksum: 0x634e [correct]
        [Header checksum status: Good]
        [Calculated Checksum: 0x634e]
    Source Address: 192.168.111.10
    Destination Address: 172.16.8.5
    Data (2284 bytes)
```



Secondo collegamento: cattura



No.	Time	Source	Destination	Protocol
1	0.0000...	192.168.111.10	172.16.8.5	IPv4
2	0.0000...	192.168.111.10	172.16.8.5	IPv4
3	0.0011...	172.16.8.5	192.168.111...	IPv4
4	0.0012...	172.16.8.5	192.168.111...	TPv4

Frame 1: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface
Ethernet II, Src: FujitsuT_5c:3d:54 (90:1b:0e:5c:3d:54), Dst: Cisco WS-C3750-24P (00:0c:29:00:00:04)
Internet Protocol Version 4, Src: 192.168.111.10, Dst: 172.16.8.5
 0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
 > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECN)
 Total Length: 1500
 Identification: 0x43ec (17388)
 > 001. = Flags: 0x1, More fragments
 0.... = Reserved bit: Not set
 ..0.... = Don't fragment: Not set
 ...1.... = More fragments: Set
 ...0 0000 0000 0000 = Fragment Offset: 0
 Time to Live: 63
 Protocol: ICMP (1)
 Header Checksum: 0x2e6d [correct]
 [Header checksum status: Good]
 [Calculated Checksum: 0x2e6d]
 Source Address: 192.168.111.10
 Destination Address: 172.16.8.5
 Data (1480 bytes)

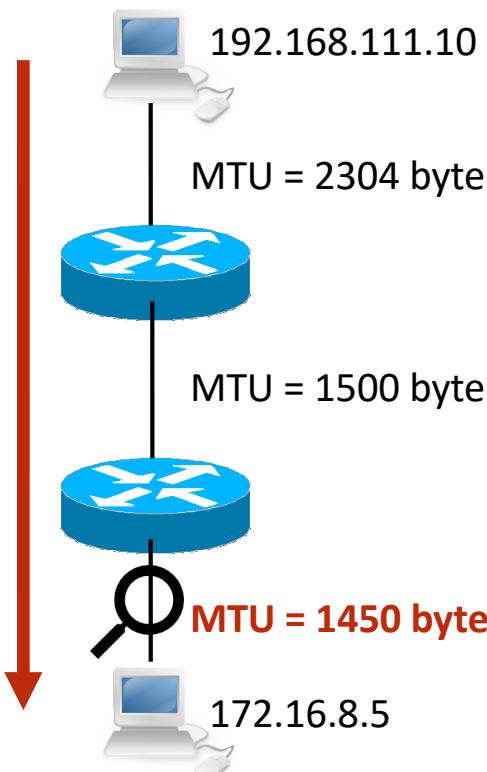
No.	Time	Source	Destination	Protocol
1	0.0000...	192.168.111.10	172.16.8.5	IPv4
2	0.0000...	192.168.111.10	172.16.8.5	IPv4
3	0.0011...	172.16.8.5	192.168.111...	IPv4
4	0.0012...	172.16.8.5	192.168.111...	TPv4

Frame 2: 838 bytes on wire (6704 bits), 838 bytes captured (6704 bits) on interface
Ethernet II, Src: FujitsuT_5c:3d:54 (90:1b:0e:5c:3d:54), Dst: Cisco WS-C3750-24P (00:0c:29:00:00:04)
Internet Protocol Version 4, Src: 192.168.111.10, Dst: 172.16.8.5
 0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
 > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECN)
 Total Length: 824
 Identification: 0x43ec (17388)
 > 000. = Flags: 0x0
 0.... = Reserved bit: Not set
 ..0.... = Don't fragment: Not set
 ...0.... = More fragments: Not set
 ...0 0000 1011 1001 = Fragment Offset: 1480
 Time to Live: 63
 Protocol: ICMP (1)
 Header Checksum: 0x5058 [correct]
 [Header checksum status: Good]
 [Calculated Checksum: 0x5058]
 Source Address: 192.168.111.10
 Destination Address: 172.16.8.5
 Data (804 bytes)

$$1500 + 824 = 2324 = 2304 + 20$$



Terzo collegamento: cattura



No.	Time	Source	Destination	Protocol
1	0.0000...	192.168.111.10	172.16.8.5	IP
2	0.0000...	192.168.111.10	172.16.8.5	IP
3	0.0000...	192.168.111.10	172.16.8.5	IP
4	0.0000...	172.16.8.5	192.168.111.10	IP

Frame 1: 1458 bytes on wire (11664 bits), 1458 bytes captured (11664 bits) on interface br0
Ethernet II, Src: Cisco_3e:fb:71 (00:24:97:3e:fb:71), Dst: Cisco_3e:fb:71 (00:24:97:3e:fb:71)
Internet Protocol Version 4, Src: 192.168.111.10, Dst: 172.16.8.5
Version: 4, Header Length: 20 bytes (5), Total Length: 1444
Identification: 0x73d3 (29651)
Flags: 0x01, More fragments: Set
Fragment Offset: 0
Time to Live: 62
Protocol: ICMP (1)
Header Checksum: 0xffbd [correct]
[Header checksum status: Good]
[Calculated Checksum: 0xffbd]
Source Address: 192.168.111.10
Destination Address: 172.16.8.5
Data (1424 bytes)

No.	Time	Source	Destination	Protocol
1	0.0000...	192.168.111.10	172.16.8.5	IP
2	0.0000...	192.168.111.10	172.16.8.5	IP
3	0.0000...	192.168.111.10	172.16.8.5	IP
4	0.0000...	172.16.8.5	192.168.111.10	IP

Frame 2: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface br0
Ethernet II, Src: Cisco_3e:fb:71 (00:24:97:3e:fb:71), Dst: Cisco_3e:fb:71 (00:24:97:3e:fb:71)
Internet Protocol Version 4, Src: 192.168.111.10, Dst: 172.16.8.5
Version: 4, Header Length: 20 bytes (5), Total Length: 76
Identification: 0x73d3 (29651)
Flags: 0x01, More fragments: Set
Fragment Offset: 0
Time to Live: 62
Protocol: ICMP (1)
Header Checksum: 0x2171 [correct]
[Header checksum status: Good]
[Calculated Checksum: 0x2171]
Source Address: 192.168.111.10
Destination Address: 172.16.8.5
Data (804 bytes)

No.	Time	Source	Destination	Protocol
1	0.0000...	192.168.111.10	172.16.8.5	IP
2	0.0000...	192.168.111.10	172.16.8.5	IP
3	0.0000...	192.168.111.10	172.16.8.5	IP
4	0.0000...	172.16.8.5	192.168.111.10	IP

Frame 3: 838 bytes on wire (6704 bits), 838 bytes captured (6704 bits) on interface br0
Ethernet II, Src: Cisco_3e:fb:71 (00:24:97:3e:fb:71), Dst: Cisco_3e:fb:71 (00:24:97:3e:fb:71)
Internet Protocol Version 4, Src: 192.168.111.10, Dst: 172.16.8.5
Version: 4, Header Length: 20 bytes (5), Total Length: 824
Identification: 0x73d3 (29651)
Flags: 0x00
[Reserved bit: Not set]
[Don't fragment: Not set]
[More fragments: Not set]
[Fragment Offset: 1480]
Time to Live: 62
Protocol: ICMP (1)
Header Checksum: 0x2171 [correct]
[Header checksum status: Good]
[Calculated Checksum: 0x2171]
Source Address: 192.168.111.10
Destination Address: 172.16.8.5
Data (804 bytes)

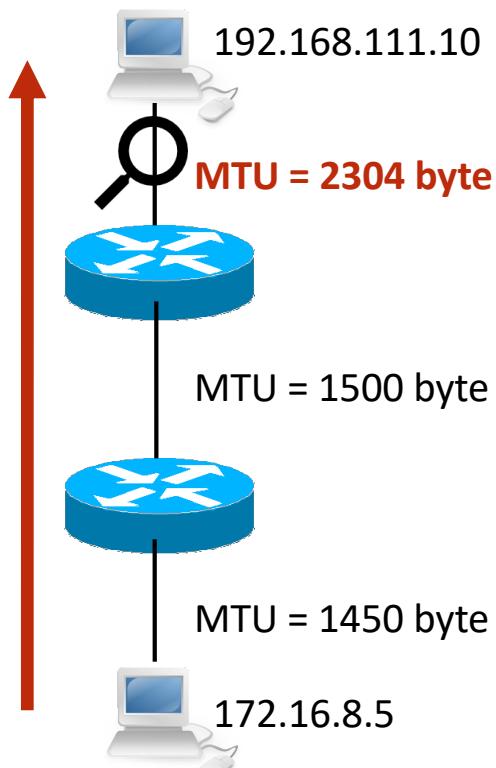
No.	Time	Source	Destination	Protocol
1	0.0000...	192.168.111.10	172.16.8.5	IP
2	0.0000...	192.168.111.10	172.16.8.5	IP
3	0.0000...	192.168.111.10	172.16.8.5	IP
4	0.0000...	172.16.8.5	192.168.111.10	IP

Frame 4: 20 bytes on wire (160 bits), 20 bytes captured (160 bits) on interface br0
Ethernet II, Src: Cisco_3e:fb:71 (00:24:97:3e:fb:71), Dst: Cisco_3e:fb:71 (00:24:97:3e:fb:71)
Internet Protocol Version 4, Src: 192.168.111.10, Dst: 172.16.8.5
Version: 4, Header Length: 20 bytes (5), Total Length: 20
Identification: 0x464 [correct]
[Header checksum status: Good]
[Calculated Checksum: 0x464]
Source Address: 192.168.111.10
Destination Address: 172.16.8.5
Data (56 bytes)

$$1444 + 76 + 824 = 2344 = 2304 + 20 + 20$$



In senso opposto



No.	Time	Source	Destination	Protocol
1	0.0000...	192.168.111.10	172.16.8.5	IP
2	0.0012...	172.16.8.5	192.168.111....	IP
3	0.0012...	172.16.8.5	192.168.111....	IP

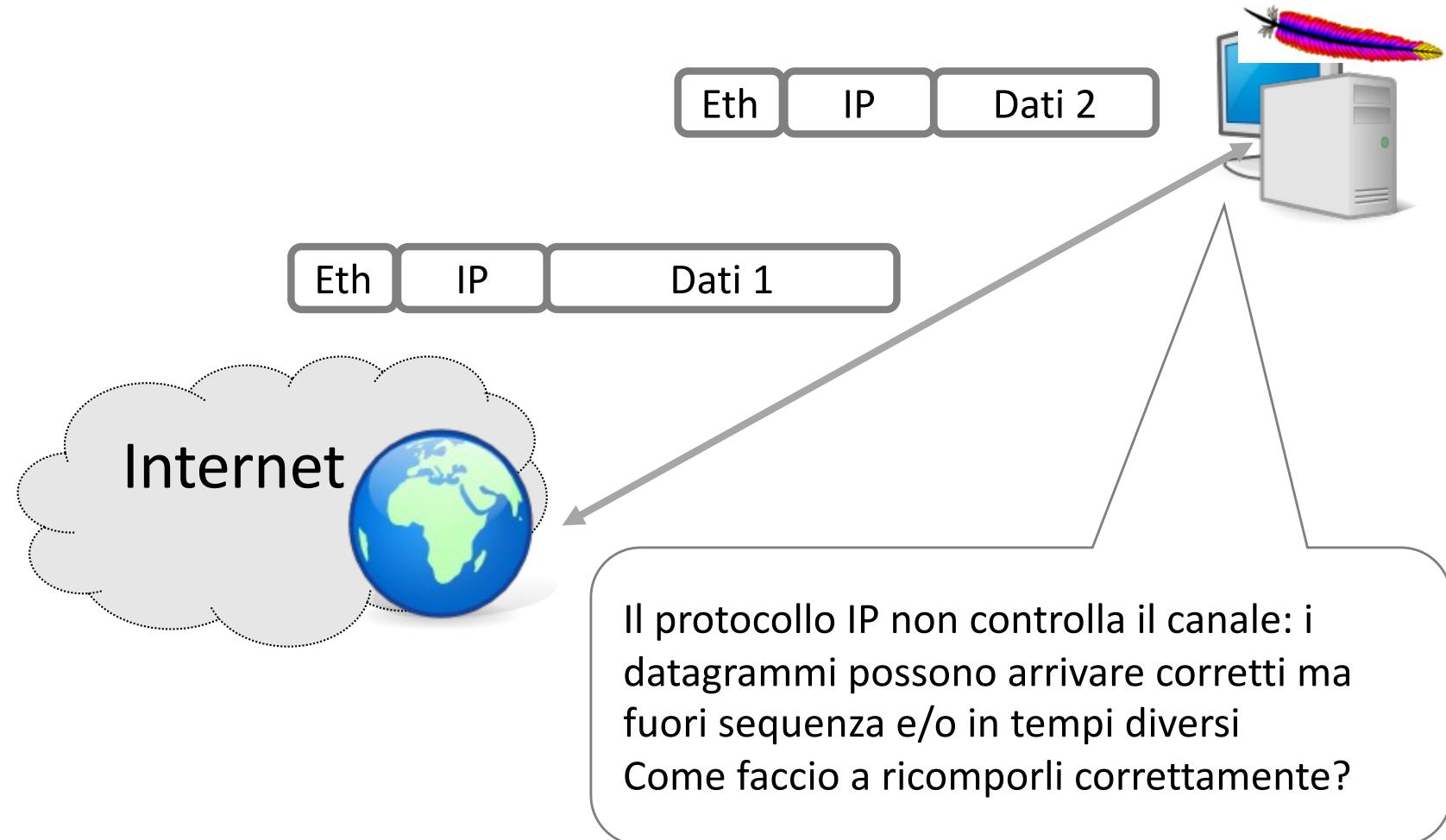
Frame 2: 1458 bytes on wire (11664 bits), 1458 bytes captured (11664 bits) on interface br0
Ethernet II, Src: 6e:3e:c2:b9:35:54 (6e:3e:c2:b9:35:54), Dst: 00:0c:29:1d:01:00 (00:0c:29:1d:01:00)
Internet Protocol Version 4, Src: 172.16.8.5, Dst: 192.168.111.10
 0100 = Version: 4
 0101 = Header Length: 20 bytes (5)
 Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 1444
 Identification: 0x91fd (37373)
 001. = Flags: 0x1, More fragments
 0.... = Reserved bit: Not set
 ..0.... = Don't fragment: Not set
 ...1.... = More fragments: Set
 ...0 0000 0000 0000 = Fragment Offset: 0
 Time to Live: 62
 Protocol: ICMP (1)
 Header Checksum: 0xe193 [correct]
 [Header checksum status: Good]
 [Calculated Checksum: 0xe193]
 Source Address: 172.16.8.5
 Destination Address: 192.168.111.10
 Data (1424 bytes)

No.	Time	Source	Destination	Protocol
1	0.0000...	192.168.111.10	172.16.8.5	IPv4
2	0.0012...	172.16.8.5	192.168.111....	IPv4
3	0.0012...	172.16.8.5	192.168.111....	IPv4

Frame 3: 894 bytes on wire (7152 bits), 894 bytes captured (7152 bits) on interface br0
Ethernet II, Src: 6e:3e:c2:b9:35:54 (6e:3e:c2:b9:35:54), Dst: 00:0c:29:1d:01:00 (00:0c:29:1d:01:00)
Internet Protocol Version 4, Src: 172.16.8.5, Dst: 192.168.111.10
 0100 = Version: 4
 0101 = Header Length: 20 bytes (5)
 Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 880
 Identification: 0x91fd (37373)
 000. = Flags: 0x0
 0.... = Reserved bit: Not set
 ..0.... = Don't fragment: Not set
 ...0.... = More fragments: Not set
 ...0 0000 1011 0010 = Fragment Offset: 1424
 Time to Live: 62
 Protocol: ICMP (1)
 Header Checksum: 0x0316 [correct]
 [Header checksum status: Good]
 [Calculated Checksum: 0x0316]
 Source Address: 172.16.8.5
 Destination Address: 192.168.111.10
 Data (860 bytes)



Il riassemblamento



Si utilizza il Fragment offset per tutti i segmenti intermedi ($MF=1$)
Quando arriva l'ultimo frammento ($MF=0$) si controlla di avere la sequenza di offset completa



Algoritmi per il riassemblamento

- Il problema di come implementare il riassemblamento è affrontato in due RFC
 - RFC 791 è la specifica originale di IP
 - RFC 815 propone un miglioramento successivo
- Da RFC 791
 - Every internet module **must be able to forward a datagram of 68 octets** without further fragmentation. This is because an internet header may be up to 60 octets, and the minimum fragment is 8 octets.



Riassemblamento (RFC 791)

Procedure:

```

(1)  BUFID <- source|destination|protocol|identification;
(2)  IF F0 = 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 F0*8 to
           octet (TL-(IHL*4))+F0*8;
(9)  set RCVBT bits from F0
           to F0+((TL-(IHL*4)+7)/
(10) IF MF = 0 THEN TDL <- TL-(IHL*4)+(F0*8)
(11) IF F0 = 0 THEN put header in header buffer
(12) IF TDL # 0
(13)   AND all RCVBT bits from 0
           to (TDL+7)/8 are
(14)   THEN TL <- TDL+(IHL*4)
(15)   Submit datagram to next step;
(16)   free all reassembly resources
         for this BUFID; DONE.
(17)   TIMER <- MAX(TIMER,TTL);
(18)   give up until next fragment or timer expires;
(19) timer expires: flush all reassembly with this BUFID; DONE.

```

Notation:

F0	- 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

Parametri identificativi del
datagramma
Garantiscono che si considerino
solamente frammenti dello stesso
datagramma originale



Riassemblamento (RFC 791)

Procedure:

```

(1) BUFID <- source|destination|protocol|identification;
(2) IF F0 = 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
    BUFID from octet F0*8 to
                  octet (TL-(IHL*4))+F0*8;
(9)   set RCVBT bits from F0
                  to F0+((TL-(IHL*4)+F0*8)-(TDL+7)/8);
(10)  IF MF = 0 THEN TDL <- TL-(IHL*4)+(F0*8);
(11)  IF F0 = 0 THEN put header in header buffer;
(12)  IF TDL # 0
(13)    AND all RCVBT bits from 0
                  to (TDL+7)/8 and
(14)    THEN TL <- TDL+(IHL*4);
(15)    Submit datagram to next step;
(16)    free all reassembly resources
        for this BUFID; DONE.
(17)    TIMER <- MAX(TIMER,TTL);
(18)    give up until next fragment or timer expires;
(19) timer expires: flush all reassembly with this BUFID; DONE.

```

Notation:

F0	- 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

Se

- Fragment Offset = 0 (comincia dal primo byte del datagramma originale)
- More Fragments = 0 (non ci sono altri frammenti)

Allora tutto il datagramma è stato ricostruito e può essere consegnato



Riassemblamento (RFC 791)

Procedure:

```

(1)  BUFID <- source|destination|protocol|identification;
(2)  IF F0 = 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 F0*8 to
        octet (TL-(IHL*4))+F0*8;
(9)    set RCVBT bits from F0
        to F0+((TL-(IHL*4)+7)/8);
(10)   IF MF = 0 THEN TDL <- TL-(IHL*4)+(F0*8)
(11)   IF F0 = 0 THEN put header in header bu
(12)   IF TDL # 0
(13)     AND all RCVBT bits from 0
        to (TDL+7)/8
(14)     THEN TL <- TDL+(IHL*4)
(15)     Submit datagram to next step;
(16)     free all reassembly resources
        for this BUFID; DONE.
(17)     TIMER <- MAX(TIMER,TTL);
(18)     give up until next fragment or timer
(19) timer expires: flush all reassembly with this

```

Notation:

F0	- 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

Vengono allocate le risorse di memoria per il datagramma (Total Data Length)
Viene fissato un timer (timeout)
Il frammento viene memorizzato al punto giusto della sequenza
F0*8 perché si ragiona a blocchi di 8 byte (64 bit)



In sintesi

- L'algoritmo proposto in RFC 791 mira a contare il numero di byte ricevuti e controllare che questo numero sia uguale alla dimensione originale del datagramma
- L'algoritmo non è particolarmente efficace
- RFC 815 propone un algoritmo molto più semplice ed efficace che non ha problemi con alcuna combinazione di frammenti



RFC 815

- Utilizza il concetto di «buco (hole)»
- `hole.first` offset dell'inizio di un buco (dati mancanti)
- `hole.last` offset della fine di un buco
- Crea una lista ordinata di buchi che inizialmente è composta da un solo buco uguale a tutto il pacchetto
 - Istante iniziale: istante di ricezione di un primo frammento con informazioni identificative non ancora registrate
 - Si crea una lista con un elemento solo che ha
 - `hole.first=0` e `hole.last=infinito`



Algoritmo

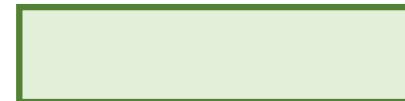
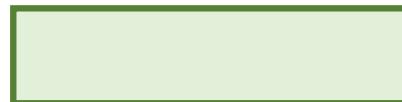
- Ricevuto un frammento si procede a controllare la lista dei buchi
 - Se `fragment.first > hole.last` si passa al buco successivo
 - Se `fragment.last < hole.first` si passa al buco successivo
 - Se non è vera nessuna delle precedenti il segmento arrivato intercetta il buco corrente che va sostituito da un buco nuovo
 - Se `fragment.first > hole.first` il nuovo buco avrà `hole.first=hole.first` e `hole.last=fragment.first`
 - Se `fragment.last < hole.last` il nuovo buco avrà `hole.first=fragment.last` e `hole.last=hole.last`
- L'algoritmo termina quando **non ci sono più buchi**



Graficamente

hole.first

hole.last





Graficamente

hole.first

hole.last

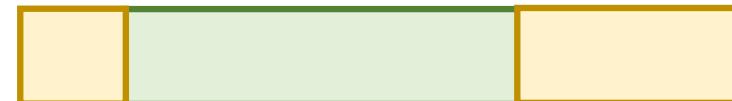
Hole

New Hole

Hole

New Hole

Hole



Hole



Significato delle PCI (3)

- **Time to live (TTL)** : max numero di nodi attraversabili
 - Il nodo sorgente attribuisce un valore maggiore di 0 a TTL (tipicamente TTL = 64, al massimo 255)
 - Ogni nodo che attraversa il datagramma pone TTL = TTL - 1
 - Il primo nodo che vede TTL = 0 distrugge il datagramma
- **Protocol** : indica a quale protocollo di livello superiore appartengono i dati del datagramma
- **Header checksum** : controllo di errore della sola intestazione, viene ricalcolato da ogni nodo attraversato dal datagramma
- **Source and Destination Address** : indirizzi sorgente e destinazione



Significato delle PCI (4)

- **Options** : contiene opzioni relative al trasferimento del datagramma (registrazione del percorso, meccanismi di sicurezza), è perciò di lunghezza variabile
- **Padding** : bit privi di significato aggiunti per fare in modo che l' intestazione sia con certezza multipla di 32 bit



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

L'instradamento IP



Instradamento

- La rete Internet è una rete a commutazione di pacchetto
 - Oggi un sistema molto complesso
- In generale esistono più modi per raggiungere una destinazione da una certa sorgente
- Chi decide quale percorso seguire e come lo fa?
- Si decide pacchetto per pacchetto o per flusso di dati applicativi?
- ...

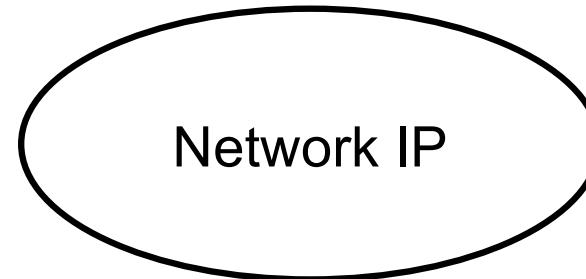
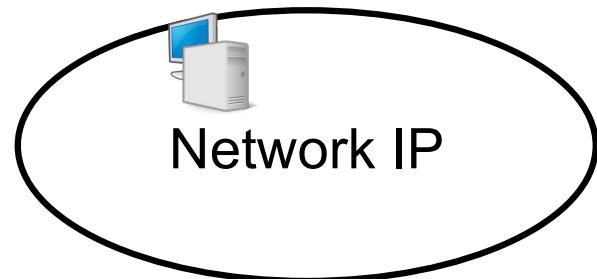


Come funziona Internet

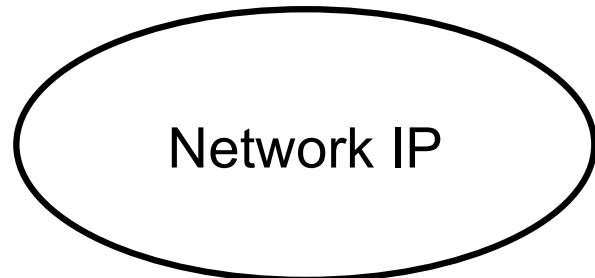
- Internet è una grande “rete di reti”
- La componente elementare è la **network IP**
 - Ogni network IP è una sorta di isola
 - L'isola tipicamente contiene calcolatori che fungono da nodi terminali della rete detti **host**
 - Le isole sono interconnesse da apparati che svolgono la funzione di “ponte”
 - Si tratta di calcolatori specializzati detti **router** o **gateway**



Internet: reti di reti



Tante Network IP isolate





La tecnologia

- Ogni network IP può essere implementata con una **tecnologia specifica**
- Esempio
 - Wi-Fi : Network realizzata con tecnologia wireless in area locale
 - ADSL e xDSL: Network realizzata con tecnologia a media distanza via cavo tramite infrastruttura di uno specifico fornitore di servizio pubblico
 - Ethernet: Network realizzata con tecnologia a breve distanza via cavo privata in area locale
 - GPRS/EDGE/LTE: Network realizzata con tecnologia radio a media distanza tramite infrastruttura di uno specifico fornitore di servizio pubblico



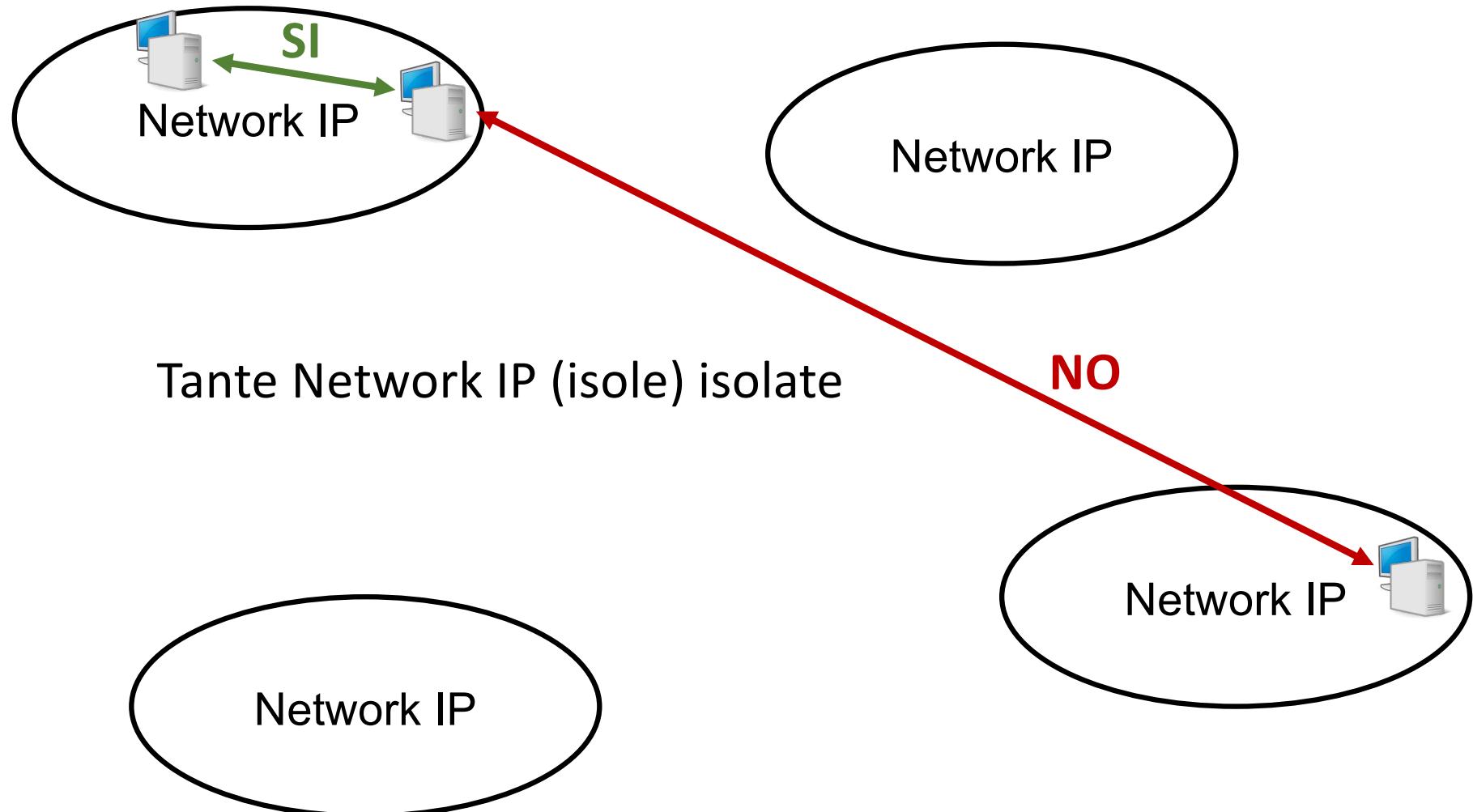
La network IP

- I calcolatori di una network IP sono connessi dalla medesima infrastruttura di rete fisica (livelli 1 e 2)

- Ipotesi fondamentale
 - Tutti gli host appartenenti alla medesima network IP sono in grado di parlare tra loro grazie alla tecnologia con cui essa viene implementata



Internet: reti di reti





Rete logica e rete fisica

- Nella terminologia di Internet si definisce
 - **Rete logica**: la network IP a cui un Host appartiene logicamente
 - **Rete fisica**: la rete (tipicamente LAN) a cui un Host è effettivamente connesso
- La rete fisica normalmente ha capacità di instradamento e può avere indirizzi locali (es. indirizzi MAC)
- L'architettura a strati nasconde gli indirizzi fisici e consente alle applicazioni di lavorare solo con indirizzi IP



Interconnettere le isole

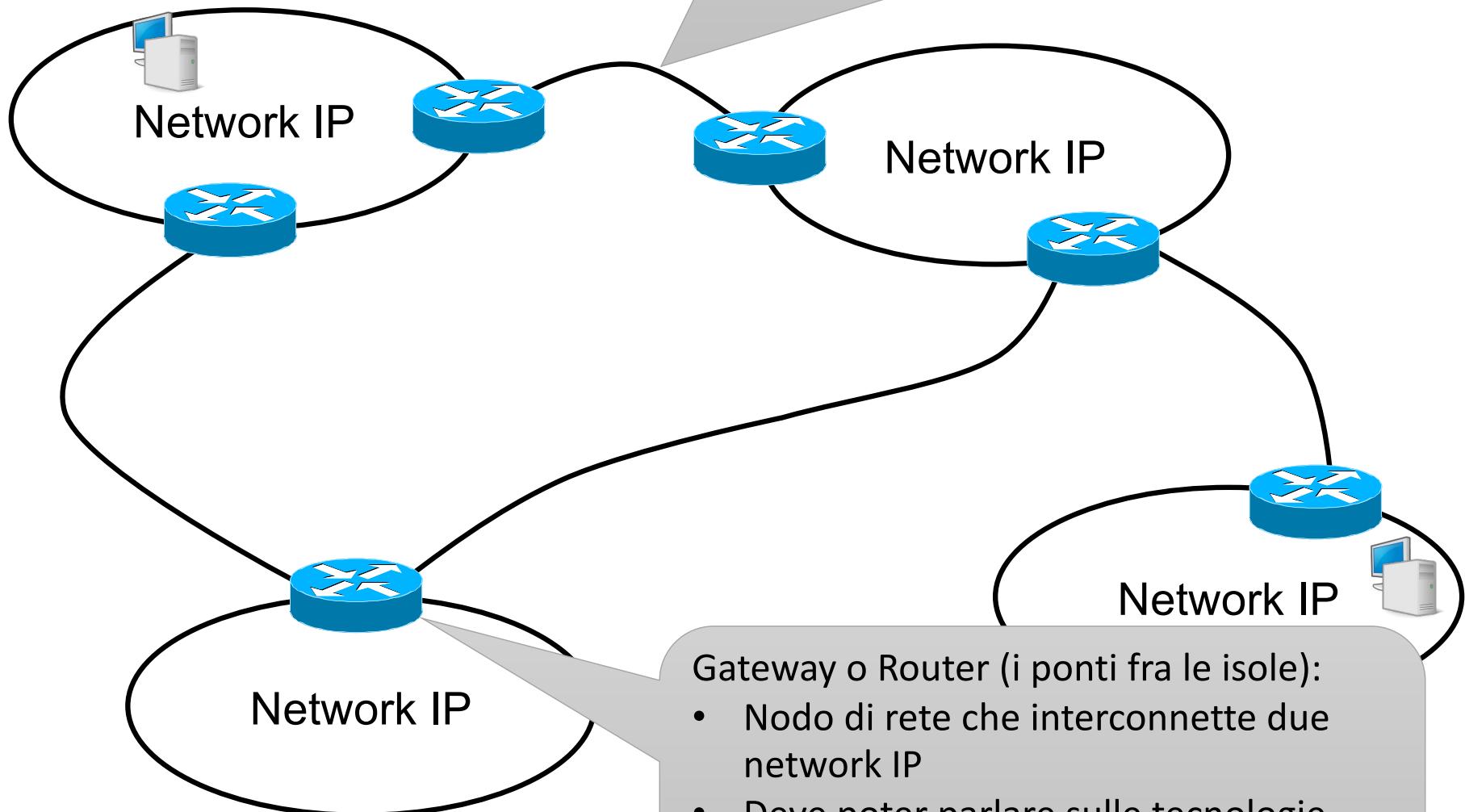
- Per far parlare tra loro le isole (network IP) è necessario che
 - Vi siano dei collegamenti fra le isole stesse, spesso realizzati con tecnologie diverse da quelle dell'isola
 - Vi siano degli apparati che permettono di usare questi collegamenti nel modo opportuno
 - Sia possibile scegliere il giusto collegamento verso l'isola che si vuole raggiungere

I router



Collegamento fra router:

- Può essere una tecnologia simile a quella delle network oppure molto diversa

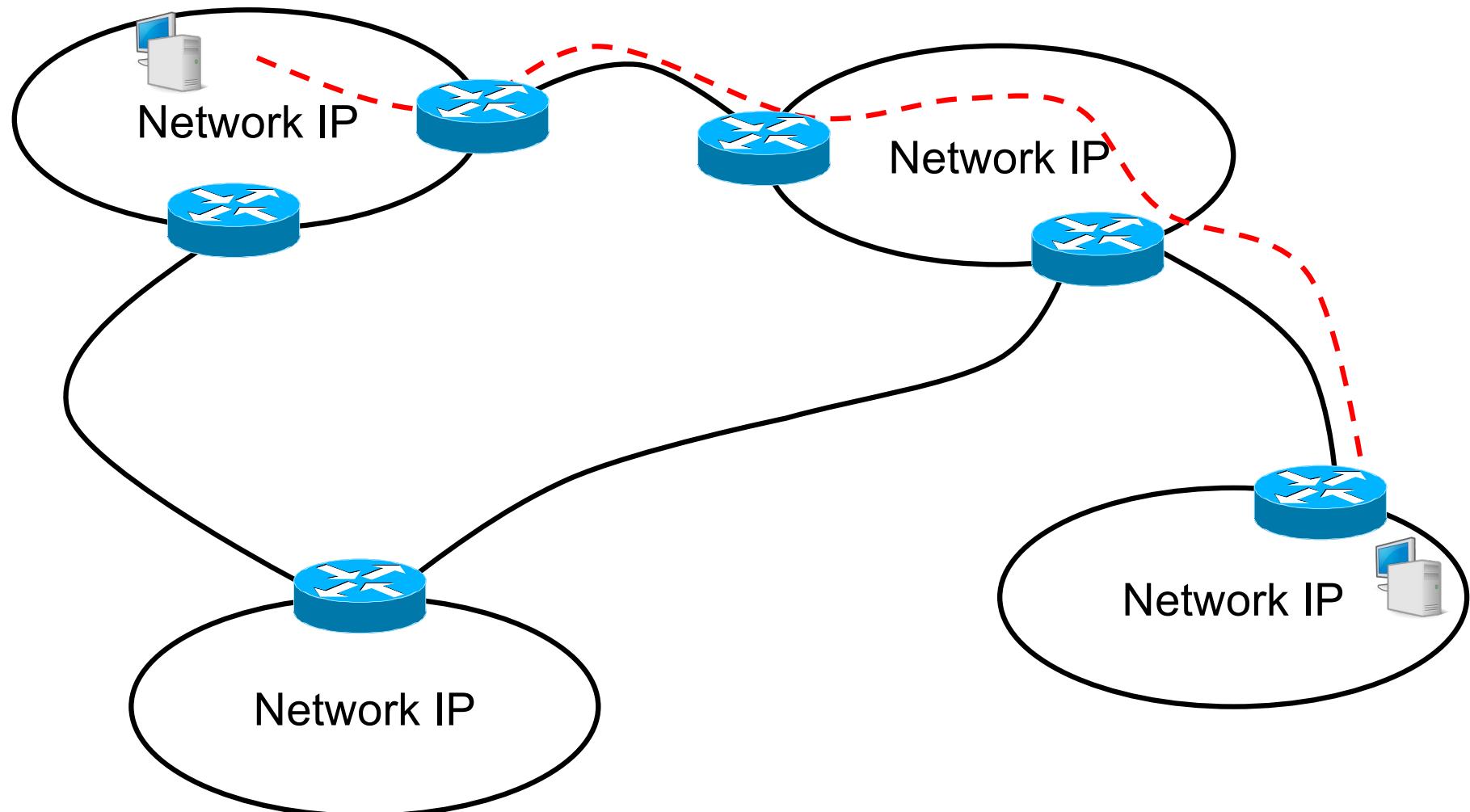


Gateway o Router (i ponti fra le isole):

- Nodo di rete che interconnecte due network IP
- Deve poter parlare sulle tecnologie specifiche delle due Network
- Ha funzioni dal livello 1 al livello 3 OSI



Il percorso end-to-end





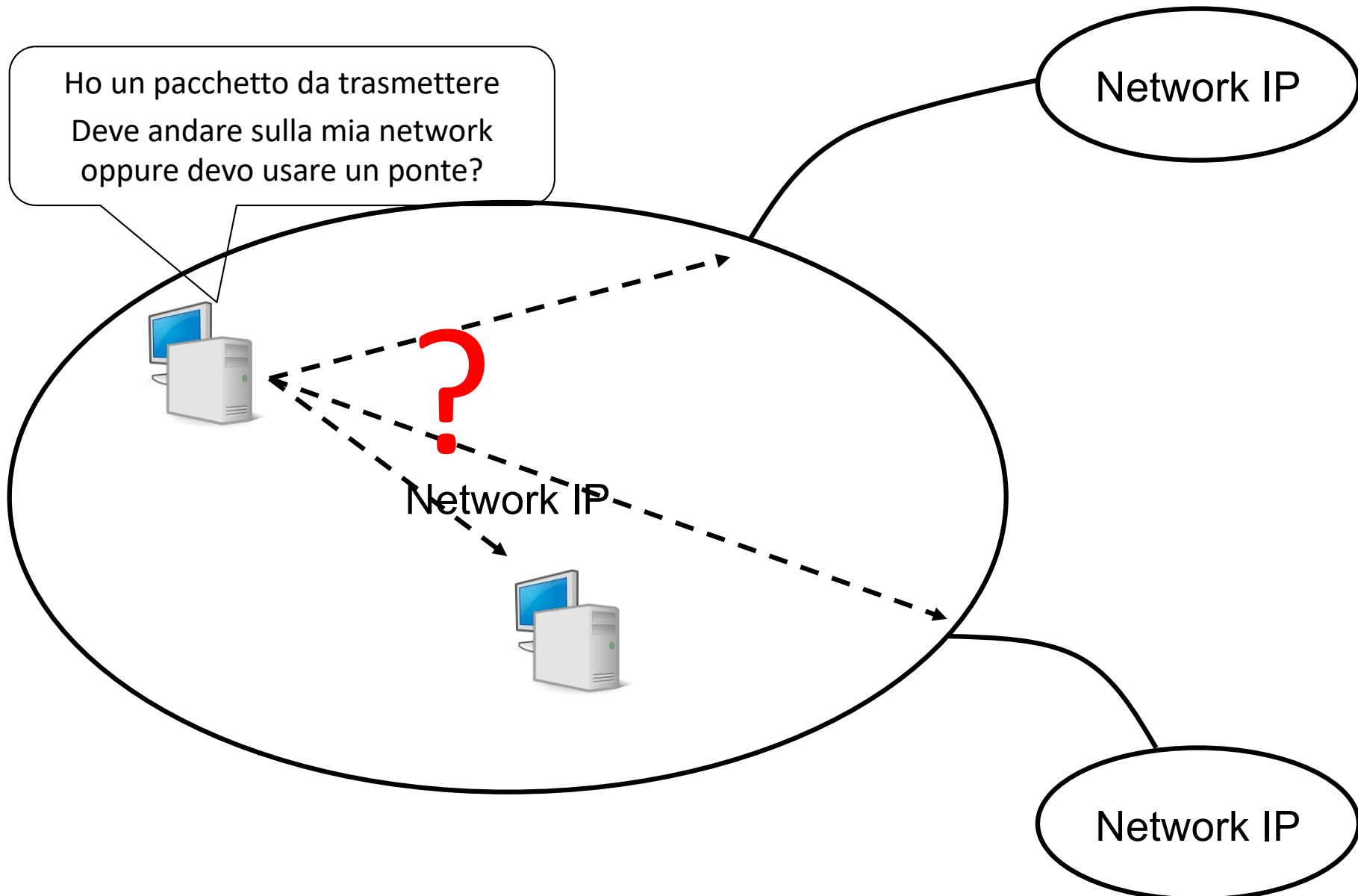
Cosa fa IP

- La tecnologia IP è agnosta rispetto alla tecnologia con cui sono realizzate le network
 - Il protocollo IP è concepito per lavorare indifferentemente su tecnologie diverse

- L'obiettivo di IP è quello di rendere possibile il dialogo fra network a prescindere dalla loro implementazione e localizzazione



La domanda cruciale



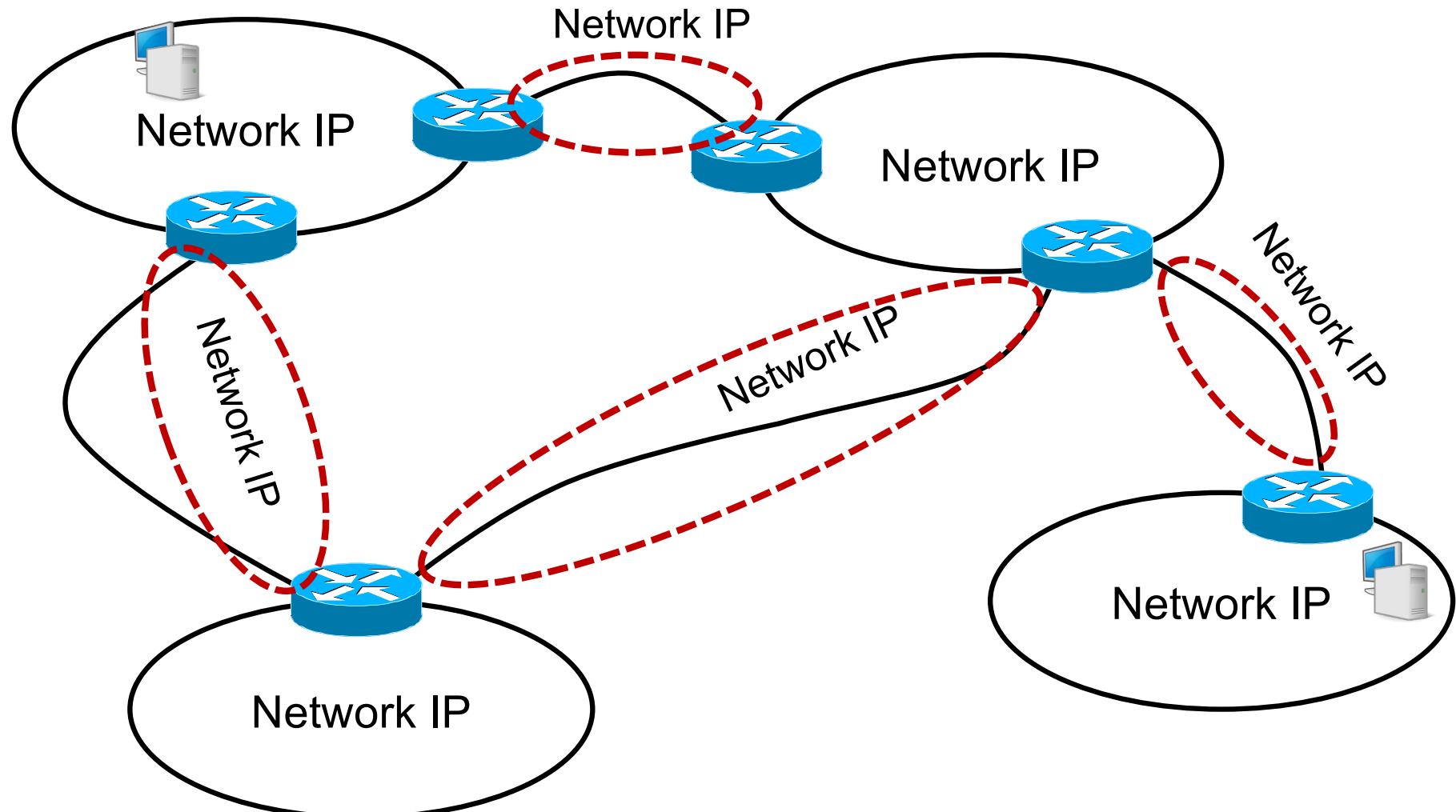


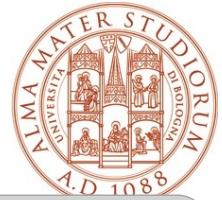
La risposta

- Ogni nodo di Internet ha una base dati di destinazioni possibili
 - Quando deve inviare un datagramma
 - Parte dall'indirizzo IP di destinazione
 - Legge la base dati
 - Decide quale azione intraprendere
- La tecnologia della propria network può essere utilizzata:
 - Per raggiungere la destinazione finale
 - Per raggiungere il primo ponte da attraversare

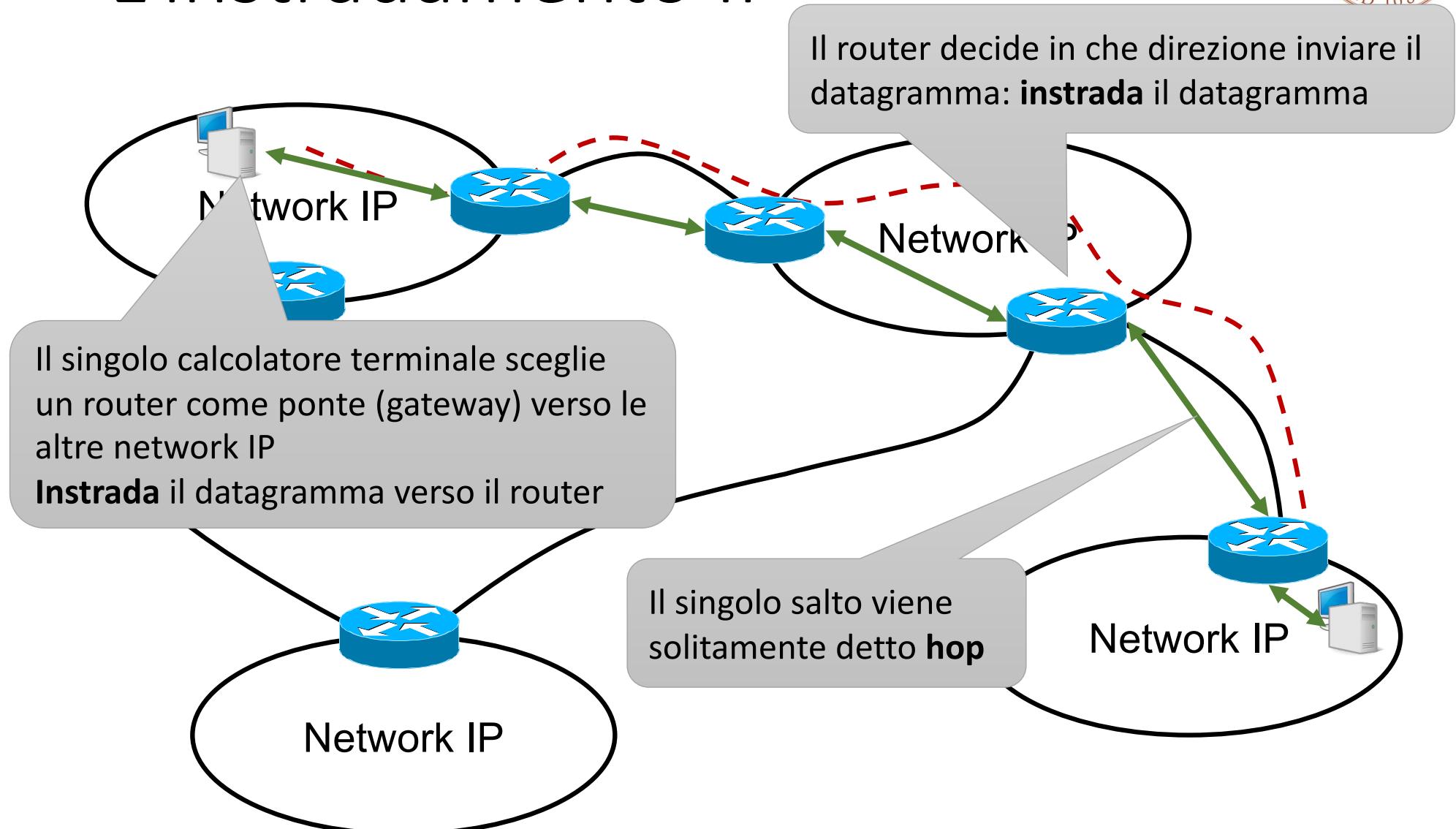


Le network fra i router





L'instradamento IP





Semantica dell'indirizzo IP

- L' indirizzo IP è logicamente suddiviso in due parti:
 - **Network (Net) ID**
 - Prefisso che identifica la **Network IP** a cui appartiene l' indirizzo
 - Tutti gli indirizzi di una medesima **Network IP** hanno il medesimo *Network ID*
 - **Host ID**
 - Identifica l' host (l' interfaccia) vero e proprio di una certa Network
- Per Net e Host ID vengono utilizzati bit contigui
 - Net ID occupa la parte *sinistra* dell' indirizzo
 - Host ID occupa la parte *destra* dell' indirizzo



Reti IP private (RFC 1918)

- Alcuni gruppi di indirizzi sono riservati a reti IP private
 - Essi non sono raggiungibili dalla rete pubblica
 - I router di Internet non instradano datagrammi destinati a tali indirizzi
 - Possono essere riutilizzati in reti isolate
-
- da **10.0.0.0** a **10.255.255.255**
 - da **172.16.0.0** a **172.31.255.255**
 - da **192.168.0.0** a **192.168.255.255**



Come si distingue net-ID da host-ID?

- Si usa la netmask
 - Al numero IP viene associata una **maschera** di 32 bit

137.204.191.25

10001001.11001100.10111111.00011001
11111111.11111111.11111111.11000000

Net-ID	Host-ID
--------	---------

- I bit a 1 della netmask identificano i bit dell' indirizzo IP che fanno parte del net-ID
- La netmask si può rappresentare
 - In notazione dotted-decimal
 - 11111111.11111111.11111111.11000000 = 255.255.255.192
 - In notazione esadecimale
 - 11111111.11111111.11111111.11000000 = ff.ff.ff.c0
 - Utilizzando la notazione abbreviata
 - 11111111.11111111.11111111.11000000 = /26



Identifichiamo la network

- 137.204.191.25 con netmask 255.255.255.192
- In binario
 - 137 -> 10001001
 - 204 -> 11001100
 - 191 -> 10111111
 - 25 -> 00011011
- Net-ID = 10001001 11001100 10111111 00
- Host-ID = 011011
- È scomodo usare sequenze di bit di lunghezza variabile qualora voglia per comodità scriverle in forma decimale



La network

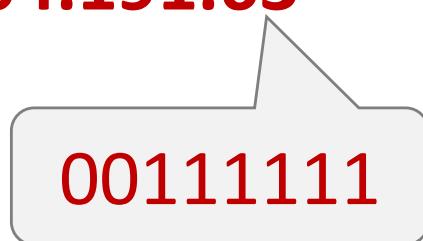
- Si mantiene il riferimento ai 32 bit
- Normalmente si indica l'intera network utilizzando il Net-ID opportuno e ponendo a 0 l'Host-ID
- Quindi nell'esempio l'identificativo della network è:
10001001 11001100 10111111 00000000
137.204.191.0
- Però così facendo si perde l'informazione sulla lunghezza del Net-ID per cui si deve scrivere
137.204.191.0/26
- Se 137.204.191.0 viene usato come «nome» della network *non si può usare per gli host*



Il broadcast

- In alcuni casi può essere utile avere modo di comunicare in contemporanea con tutti i calcolatori della propria network
 - Questo significa inviare un pacchetto IP con un indirizzo di destinazione dedicato a questo scopo (che non può essere anche indirizzo di un host specifico)
- Si definisce indirizzo di broadcast l'indirizzo che ha l'host-ID composto da soli 1
- Nell'esempio

137.204.191.63





In conclusione

- La netmask definisce il Net-ID che avrà lunghezza N
- Rimangono per l'Host-ID $H=32-N$ bit
- Quindi sono disponibili $I = 2^H$ indirizzi IP
- Di questi due non possono essere usati poichè servono per una diversa semantica
 - Host-ID di tutti 0 viene utilizzato per indicare la network
 - Host-ID di tutti 1 viene utilizzato come indirizzo broadcast
- Quindi gli indirizzi effettivamente disponibili sono
- $I' = I - 2 = 2^H - 2$



Esercizio

- Identificare l'intervallo dei numeri IP disponibili per gli host delle seguenti network
 - 192.168.10.0/22
 - Da 192.168.10.1 a 192.168.13.254
 - Numero totale $2^{10} - 2 = 1022$
 - 10.0.0.128/25
 - Da 10.0.0.129 a 10.0.0.254
 - Numero totale $2^7 - 2 = 126$

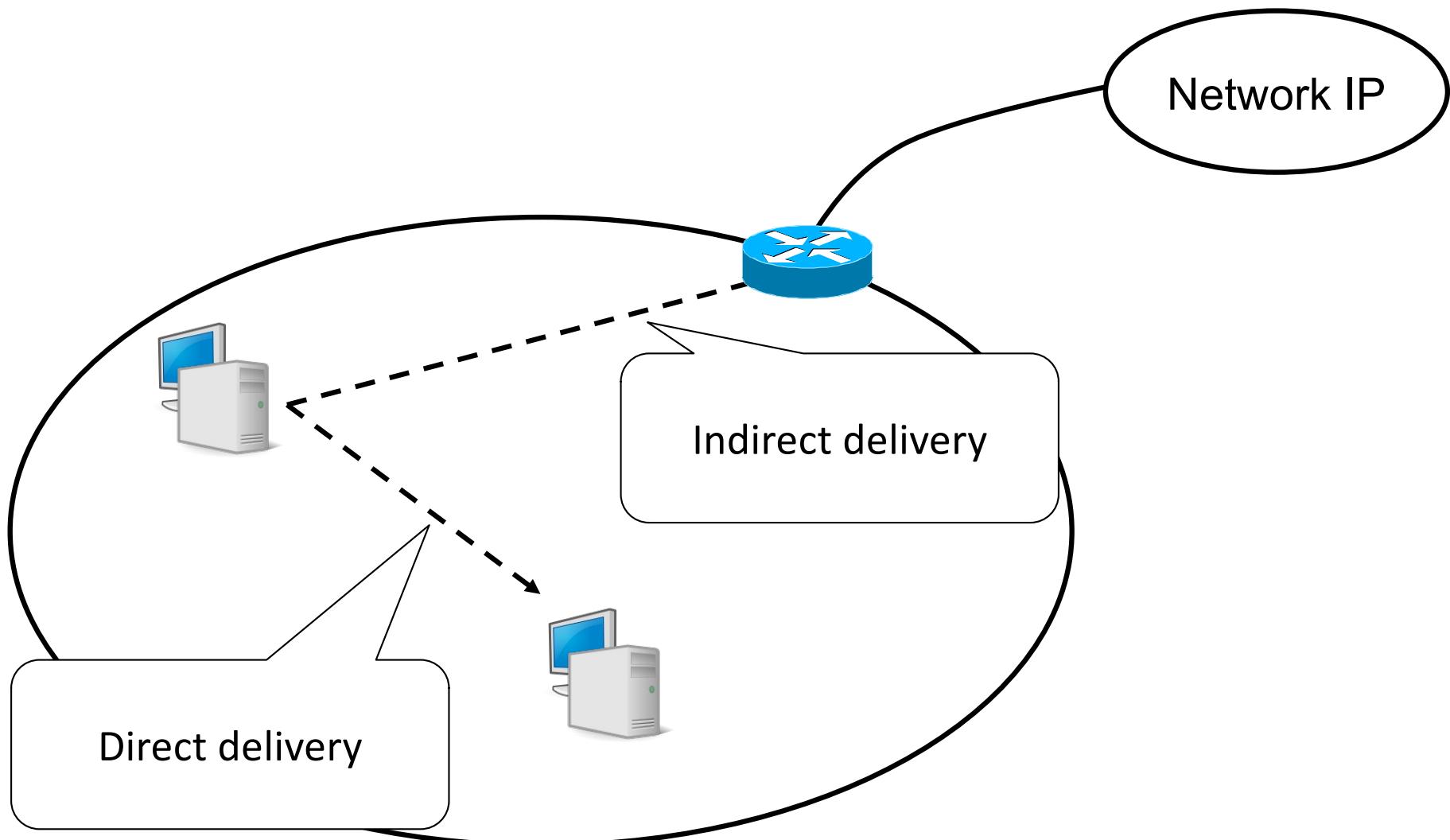


Esempio: Università di Bologna

- **Net ID = 137.204**
 - La network corrispondente ha indirizzo **137.204.0.0**
 - Tutti i numeri IP dell' Università di Bologna hanno il medesimo prefisso
- **Host ID**
 - Qualunque combinazione dei rimanenti 16 bit
 - Escluso 137.204.0.0 e 137.204.255.255
 - Server web UniBO
 - 137.204.24.35
 - Server web del DEIS
 - 137.204.24.40
 - Server web DEISNet
 - 137.204.57.85



Domanda e risposta





Instradamento diretto e indiretto

- **Direct delivery :**

- IP sorgente e IP destinatario sono sulla stessa network
- L'host sorgente spedisce il datagramma direttamente al destinatario

- **Indirect delivery :**

- IP sorgente e IP destinatario non sono sulla stessa network
- L'host sorgente invia il datagramma ad un router intermedio

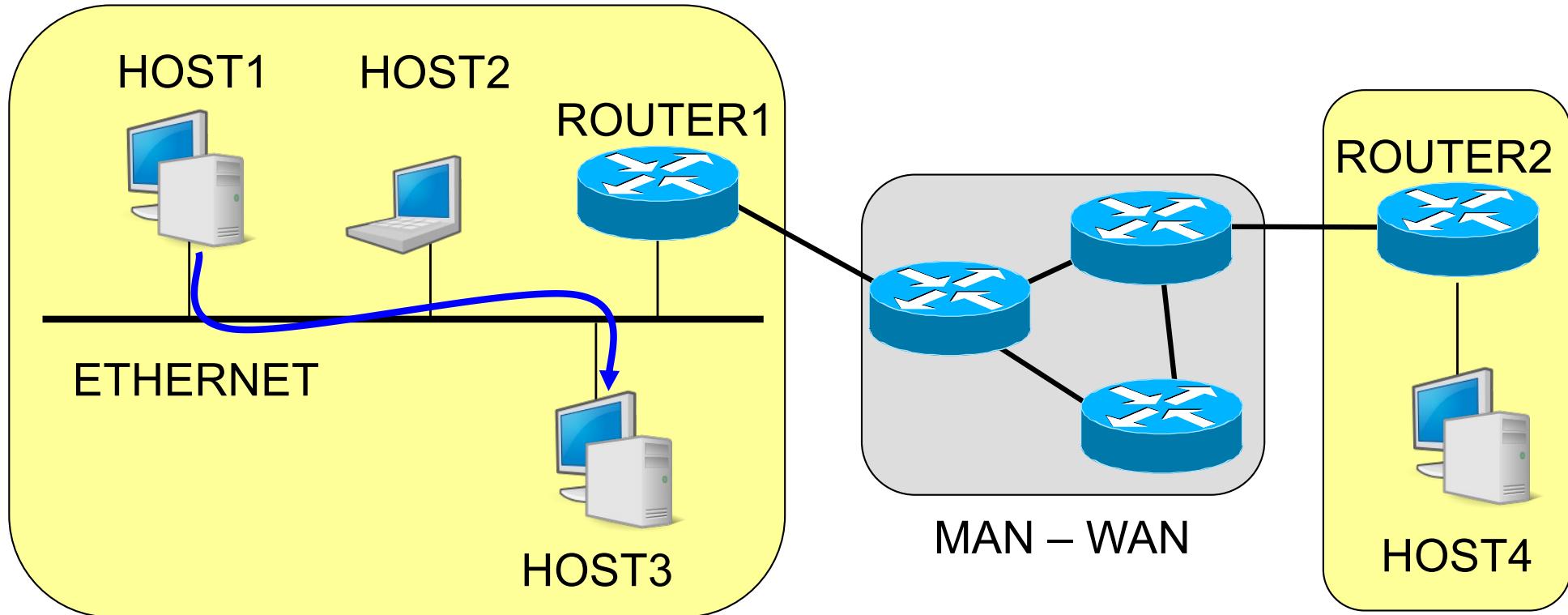
- **Routing :** scelta del percorso su cui inviare i dati

- i router formano struttura interconnessa e cooperante:

- i datagrammi passano dall'uno all'altro finché raggiungono quello che può consegnarli direttamente al destinatario



Direct Delivery



L2 ADDRESS: HOST3

IP ADDRESS: HOST3

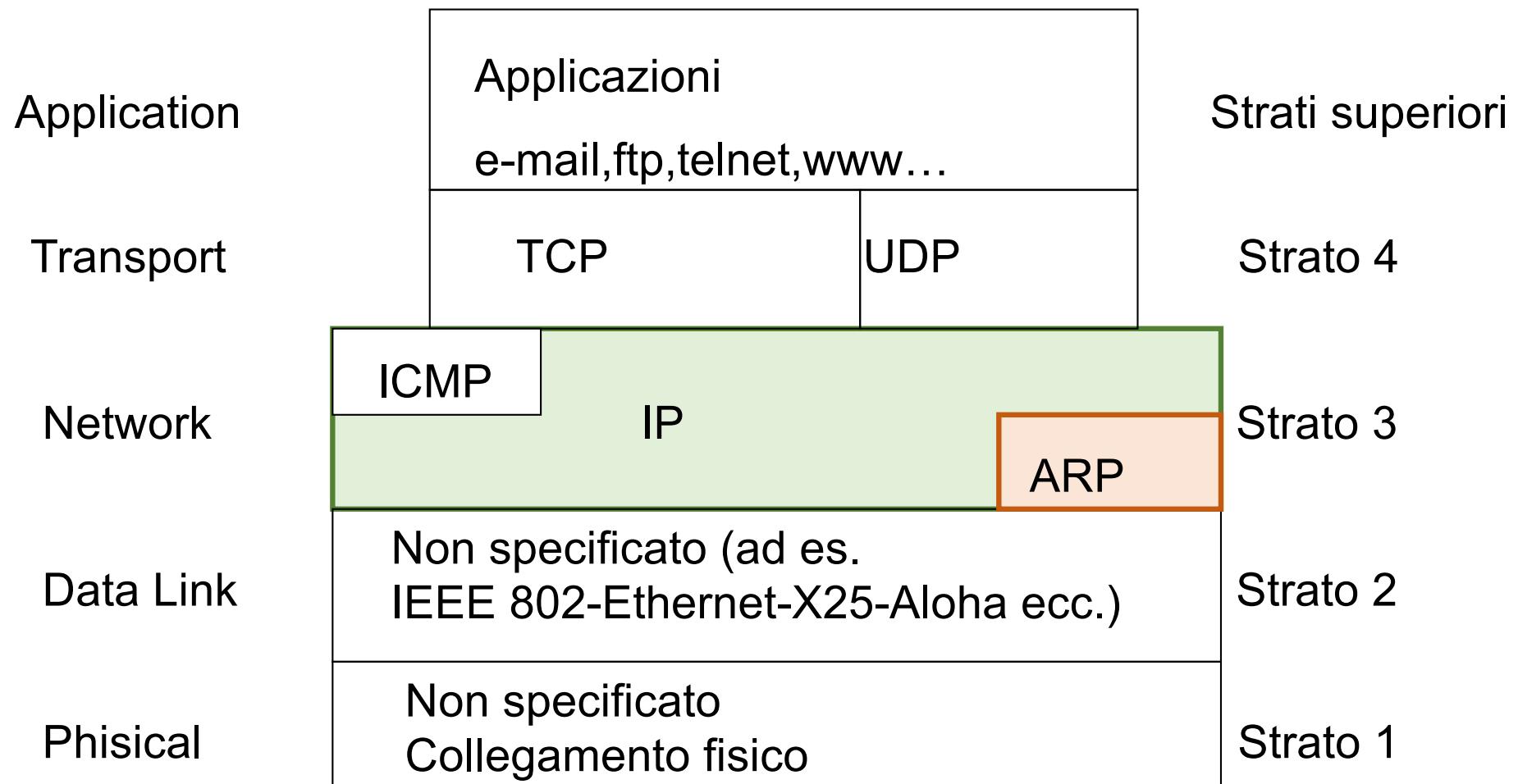
DATI

Relazione Indirizzi Fisici – Indirizzi IP

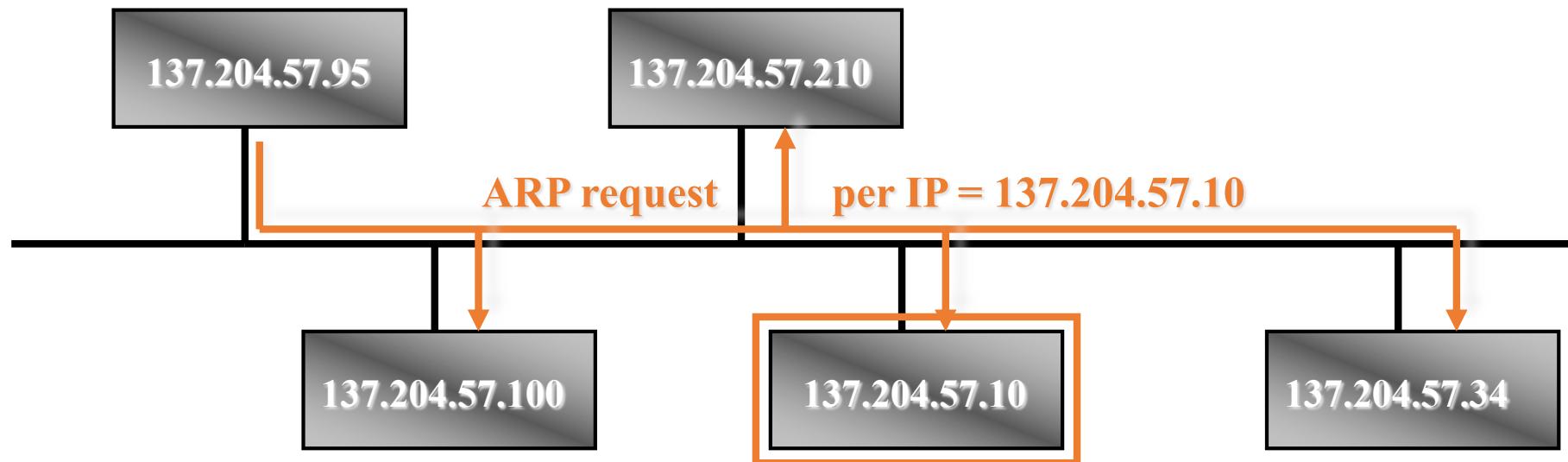
- Software di basso livello nasconde gli indirizzi fisici e consente ai livelli superiori di lavorare solo con indirizzi IP
- Gli host comunicano attraverso una **rete fisica** (ad es. LAN) quindi devono conoscere reciprocamente gli indirizzi fisici
- L'host A vuole mandare datagrammi a B, che si trova sulla stessa rete fisica e di cui conosce solo l'indirizzo IP
- Come si ricava l'indirizzo fisico di B dato il suo indirizzo IP?



Architettura

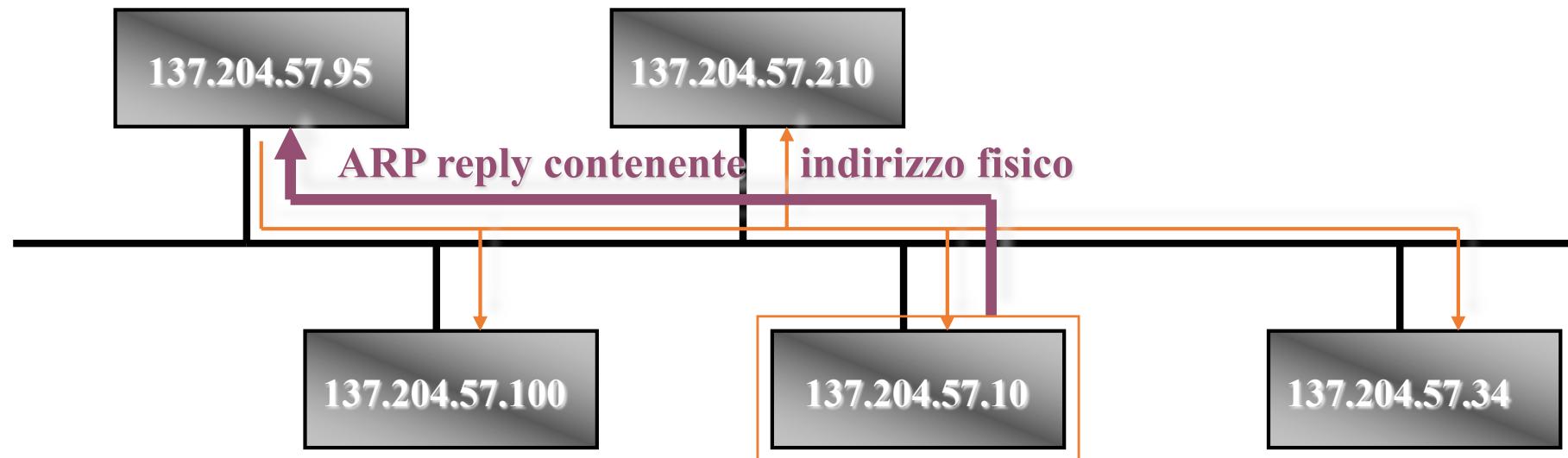


Address Resolution Protocol – ARP (RFC 826)



- Il nodo sorgente invia una trama broadcast (**ARP request**) contenente l' indirizzo IP del nodo destinazione
- Tutte le stazioni della rete locale leggono la trama broadcast

Address Resolution Protocol - ARP (3)



- Il destinatario risponde al mittente, inviando un messaggio (**ARP reply**) che contiene il proprio indirizzo fisico
- Con questo messaggio host sorgente è in grado di associare l' appropriato indirizzo fisico all' IP destinazione
- Ogni host mantiene una tabella (**cache ARP**) con le corrispondenze fra indirizzi logici e fisici

Comando ARP

arp -a

visualizza il contenuto della cache ARP con le diverse corrispondenze tra indirizzi IP e MAC

Comando ARP – Esempio

```
Command Prompt

C:\>arp -a
Interface: 137.204.57.174 on Interface 0x10000003
  Internet Address      Physical Address      Type
  137.204.57.1           08-00-20-9c-9c-93    dynamic
  137.204.57.88          00-60-b0-78-e8-fd    dynamic
  137.204.57.180         00-10-4b-db-0a-3a    dynamic
  137.204.57.181         00-30-c1-d5-ee-9b    dynamic
  137.204.57.254         00-50-54-d9-ba-00    dynamic

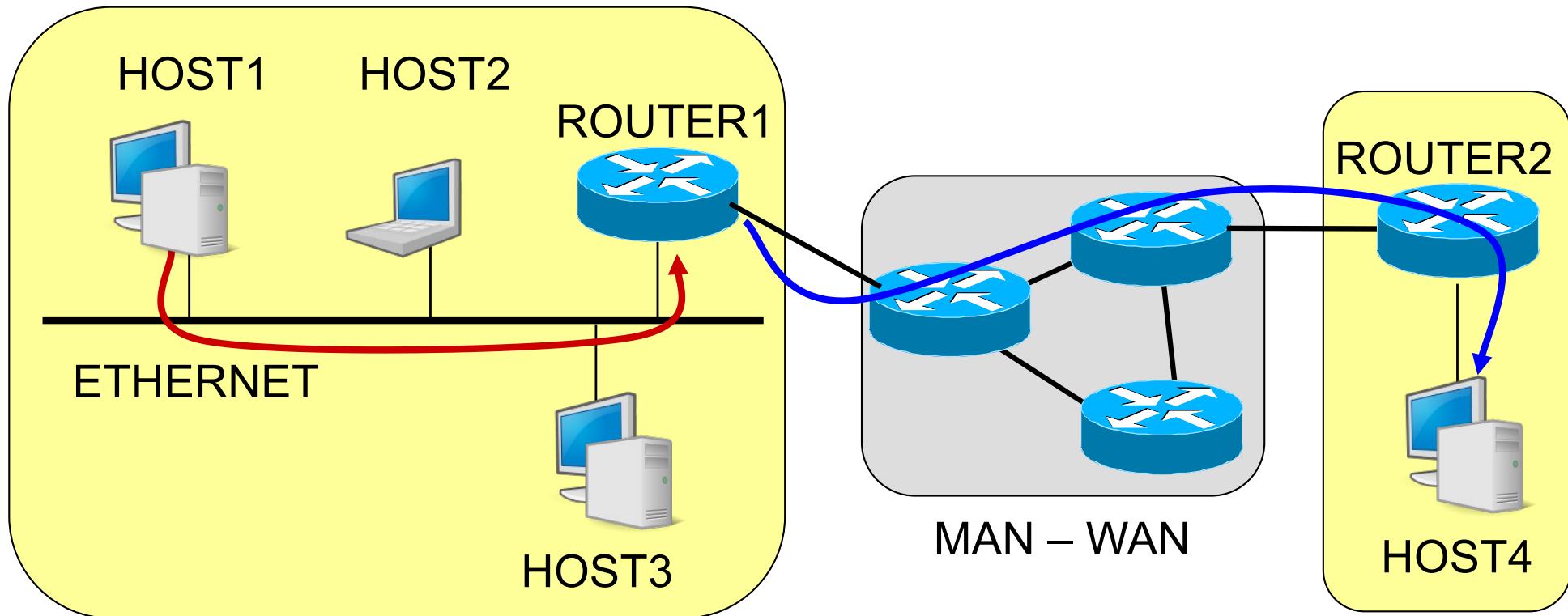
C:\>ping -n 1 137.204.57.177
Pinging 137.204.57.177 with 32 bytes of data:
Reply from 137.204.57.177: bytes=32 time<10ms TTL=128
Ping statistics for 137.204.57.177:
  Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>arp -a
Interface: 137.204.57.174 on Interface 0x10000003
  Internet Address      Physical Address      Type
  137.204.57.1           08-00-20-9c-9c-93    dynamic
  137.204.57.177          00-b0-d0-ec-46-62    dynamic
  137.204.57.180         00-10-4b-db-0a-3a    dynamic
  137.204.57.181         00-30-c1-d5-ee-9b    dynamic
  137.204.57.254         00-50-54-d9-ba-00    dynamic

C:\>_
```



Indirect Delivery



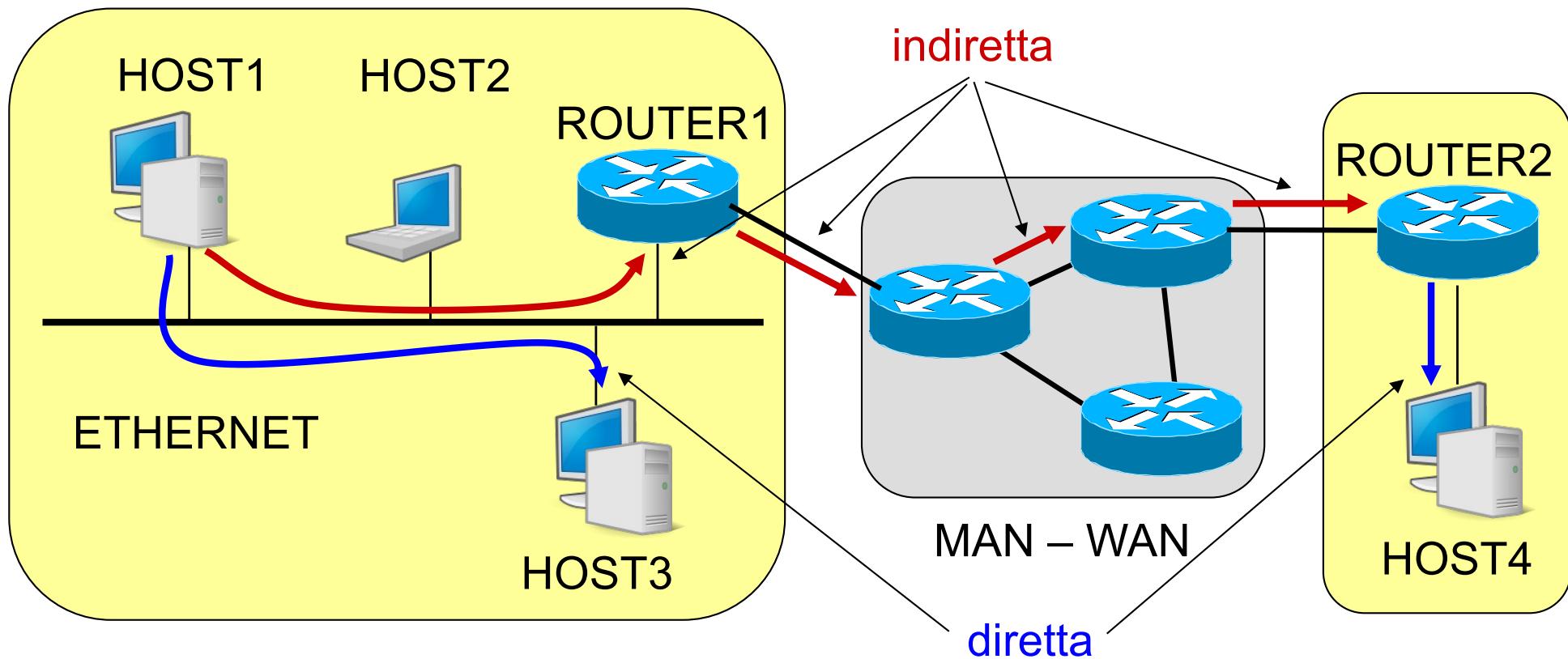
L2 ADDRESS: ROUTER1

IP ADDRESS: HOST4

DATI

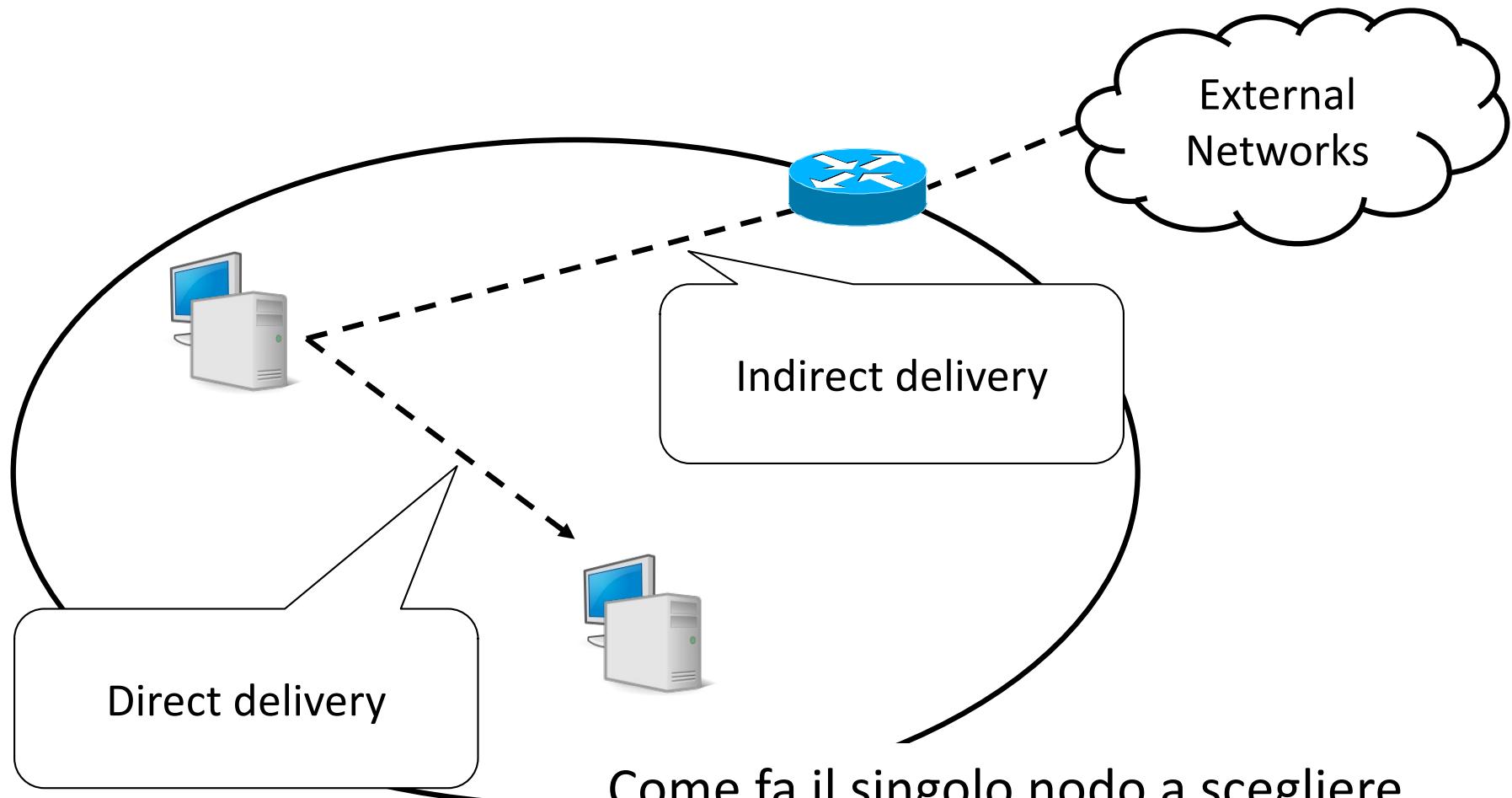
Da mittente a destinatario

- C' è sempre una consegna diretta
- Può non esserci alcuna consegna indiretta
- Possono esserci una o più consegne indirette





Come scegliere?



Come fa il singolo nodo a scegliere

- fra instradamento diretto e indiretto?
- il gateway giusto qualora ve ne siano molteplici?



La tabella di instradamento IP

- Base dati in forma di tabella
 - Righe (dette anche route, rotte, entry, record)
 - Insieme di informazioni relative alla singola informazione di instradamento
 - Colonne (dette campi)
 - Informazioni del medesimo tipo relative a diverse opzioni di instradamento
- Formato della tabella
 - Dipende dal sistema operativo e dall'implementazione
 - Le informazioni sono le medesime
 - Il modo di presentarle ed elaborarle può essere diverso



Route

- Tipici campi della singola rotta sono:
 - **Destinazione (D)**: numero IP valido
 - Può essere un indirizzo di network o di host
 - **Netmask (N)**: maschera di rete valida
 - Identifica il Net-ID
 - **Gateway (G)**: numero IP a cui consegnare il datagramma
 - Indica il tipo di consegna da effettuare
 - **Interfaccia di rete (IF)**: interfaccia di rete utilizzare (loopback compreso) per la consegna del datagramma
 - Seleziona il dispositivo hardware da utilizzare per l'invio del datagramma
 - **Metrica (M)**: specifica il “costo” di quel particolare route
 - Possono esistere più route verso una medesima destinazione



La tabella

Destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.10.1	ppp0	1
137.204.64.0	255.255.255.0	137.204.64.254	en0	1
137.204.65.0	255.255.255.0	137.204.65.254	en1	1
137.204.66.0	255.255.255.0	137.204.66.254	en2	1
137.204.67.0	255.255.255.0	137.204.67.254	en3	1
192.168.10.0	255.255.255.252	192.168.10.2	ppp0	1



Uso della tabella di instradamento

- Il singolo nodo riceve un datagramma:
 - Estrae dall' intestazione IP_D = indirizzo IP di destinazione
 - Seleziona il route per tale IP_D, confrontandolo con i campi D presenti nella tabella
 - Processo di “**table lookup**”
 - Se il route esiste
 - Esegue l'azione di instradamento suggerita dai campi G e IF
 - Se il route non esiste genera un messaggio di errore
 - Tipicamente notificato all' indirizzo sorgente (ICMP - **Destination Unreachable**)



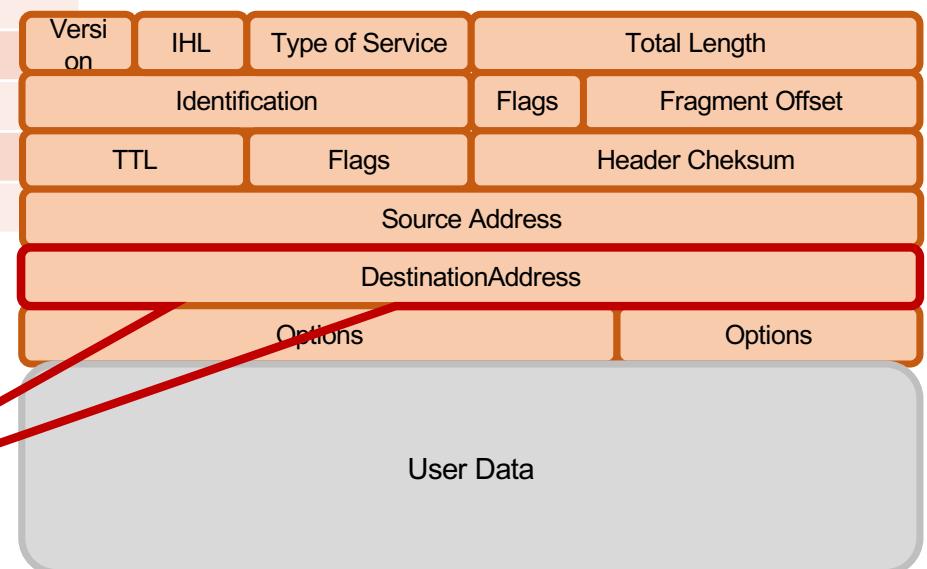
Table lookup

- La ricerca nella tabella avviene confrontando
 - Indirizzo IP di destinazione **IP_D** del datagramma
 - Destinazione (D) di ciascun route
 - Utilizzando la **netmask (N)** del route
- La procedura viene detta di “longest prefix match”
 - **IP_D AND N = R**
 - Indirizzo di destinazione del datagramma e netmask di ciascuna riga
 - **R = D ?**
 - SI : la route viene selezionata e il processo termina
 - NO : si passa al route successivo
- In quale ordine leggere i route
 - dalla riga che presenta una netmask con un numero maggiore di bit a uno



II lookup

Destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.10.1	ppp0	1
137.204.64.0	255.255.255.0	137.204.64.254	en0	1
137.204.65.0	255.255.255.0	137.204.65.254	en1	1
137.204.66.0	255.255.255.0	137.204.66.254	en2	1
137.204.67.0	255.255.255.0	137.204.67.254	en3	1
192.168.10.0	255.255.255.252	192.168.10.2	ppp0	1



Destination Address
Netmask
AND

Result
==
Destination

YES/NO



Esempio di lookup – 1

	Destinazione	Netmask	Etc.
1	0.0.0.0	0.0.0.0	...
2	192.168.2.0	255.255.255.0	...
3	192.168.2.18	255.255.255.255	...

- Datagramma con IP dest. = 192.168.2.18
 - Confronto prima con riga 3, poi con riga 2 e poi riga 1

192.168.002.018 bitwise AND
255.255.255.255
192.168.002.018 \equiv 192.168.002.018

- La riga 3 è quella giusta (host specific)



Esempio di lookup – 2

	Destinazione	Netmask	Etc.
1	0.0.0.0	0.0.0.0	...
2	192.168.2.0	255.255.255.0	...
3	192.168.2.18	255.255.255.255	...

- Datagramma con IP dest. = 192.168.2.22

192.168.002.022

255.255.255.255

192.168.002.022 != 192.168.002.018

192.168.002.022

255.255.255.000

192.168.002.000 == 192.168.002.000

- La riga 2 è quella giusta (network specific)



Esempio di lookup – 3

	Destinazione	Netmask	Etc.
1	0.0.0.0	0.0.0.0	...
2	192.168.2.0	255.255.255.0	...
3	192.168.2.18	255.255.255.255	...

- Datagramma con IP dest. = 80.48.15.170

080.048.015.170
255.255.255.255
080.048.015.170 != 192.168.002.018

080.048.015.170
255.255.255.000
080.048.015.000 != 192.168.002.000

080.048.015.170
000.000.000.000
000.000.000.000 == 000.000.000.000

- La riga 1 è quella giusta (default gateway)



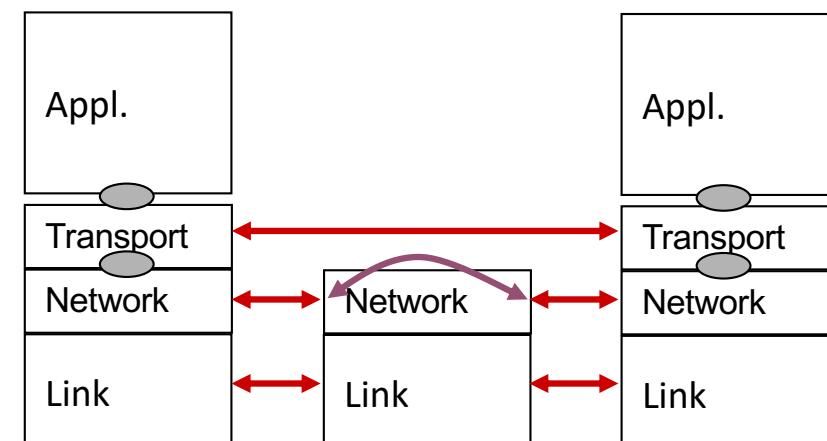
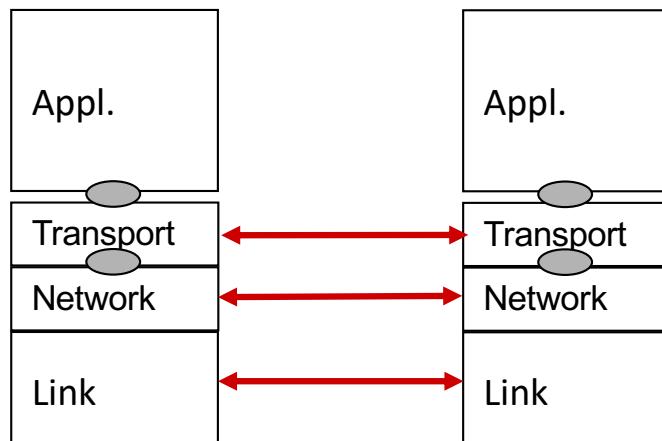
Gateway

- Nella tabella di instradamento compaiono
 - Gateway
 - Interfaccia
- Perché due informazioni distinte?
- Chi è il gateway?



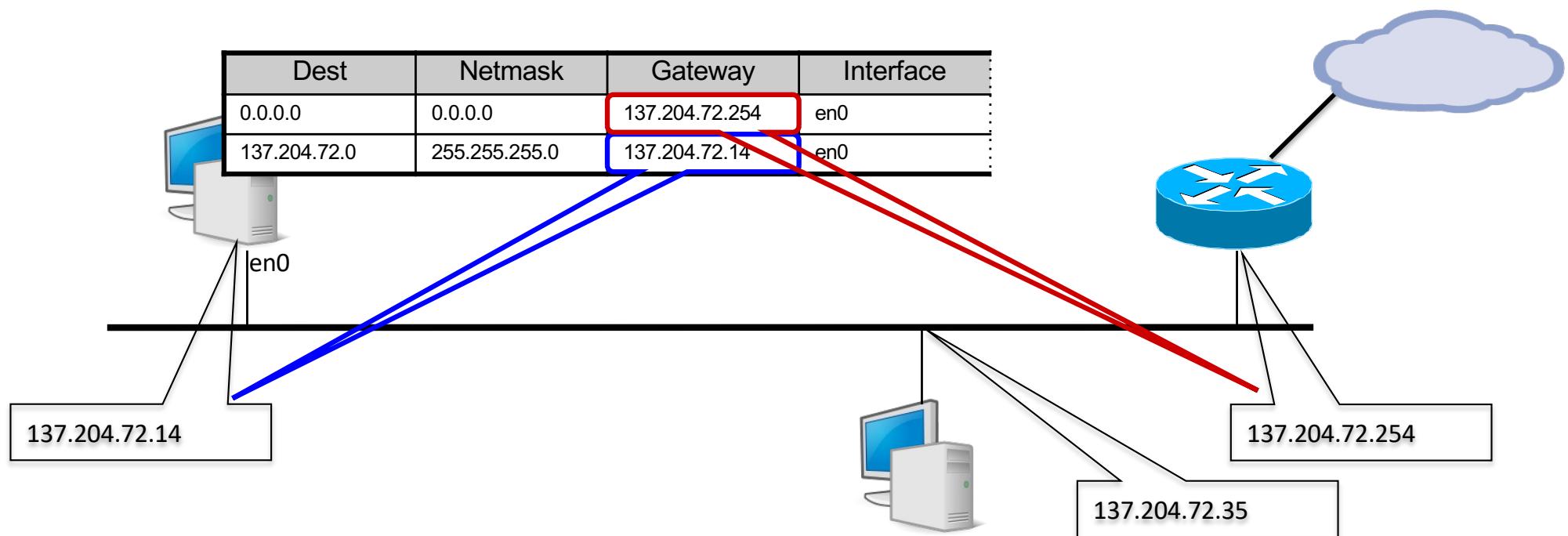
Il ruolo del Gateway

- Il table look-up sceglie la D i-esima = D_i
- La funzione di instradamento invia il datagramma a IF_i
- Con l' obiettivo di consegnarlo al **gateway** G_i
- Perché non è sufficiente IF_i ?
- L'instradamento IP è basato sull'appartenenza alla network
 - Host della medesima network possono comunicare direttamente
 - Host di network diverse comunicano tramite gateway
- **Gateway** = responsabile della consegna del datagramma



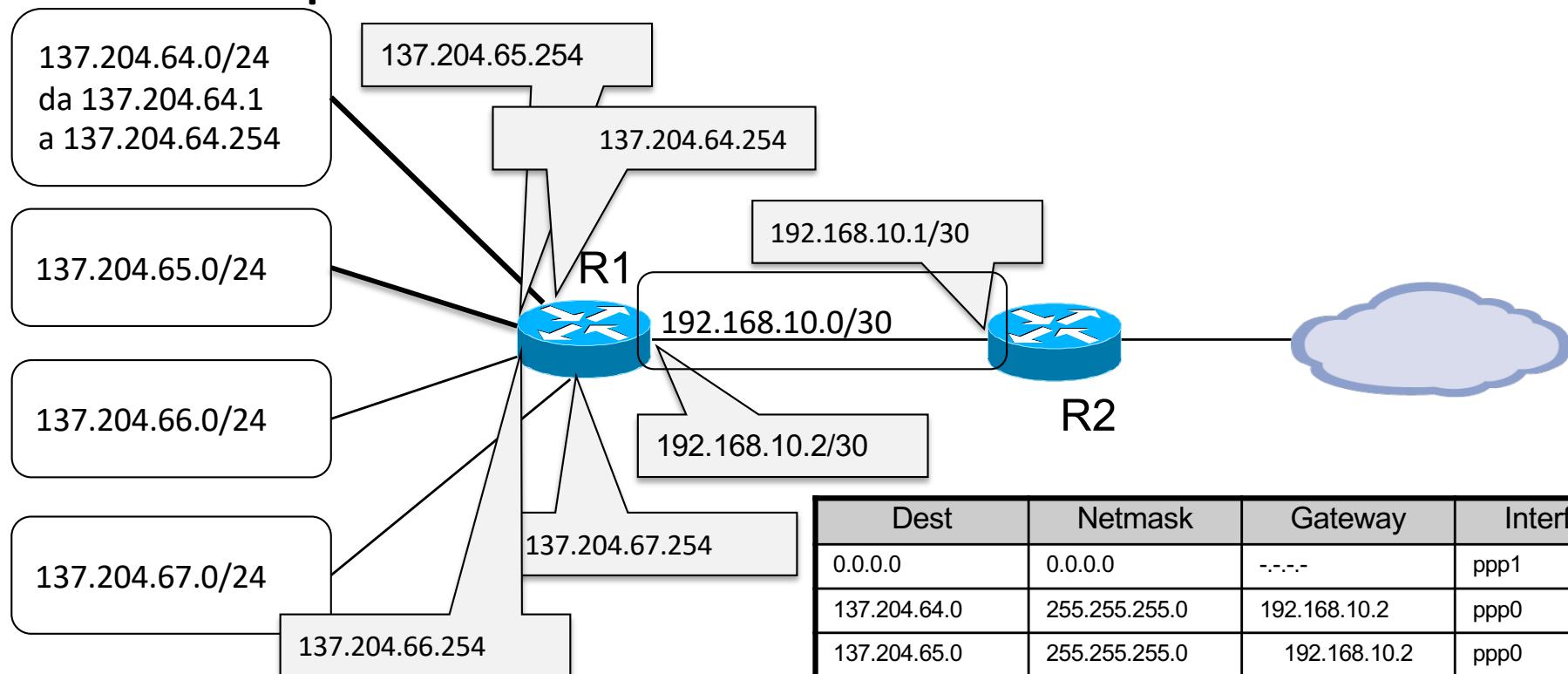
Uso del Gateway

- Il campo gateway della tabella di routing serve per specificare il tipo di instradamento
 - Instradamento diretto: la sintassi dipende dall'implementazione
 - In Windows: instradamento diretto se gateway = IP locale
 - In Linux/Unix: instradamento diretto se gateway = 0.0.0.0
 - Instradamento indiretto
 - Gateway = numero IP del router da contattare

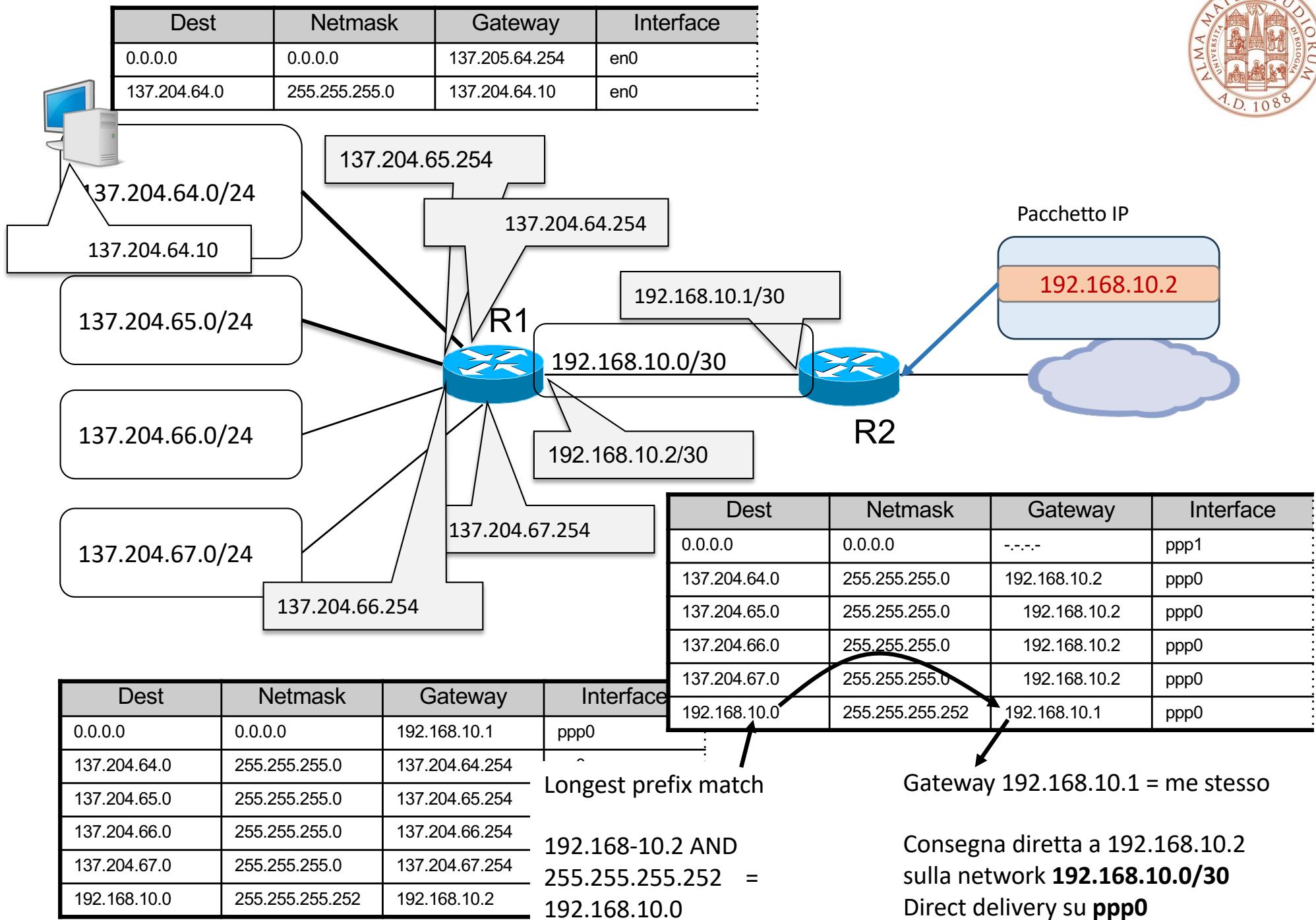


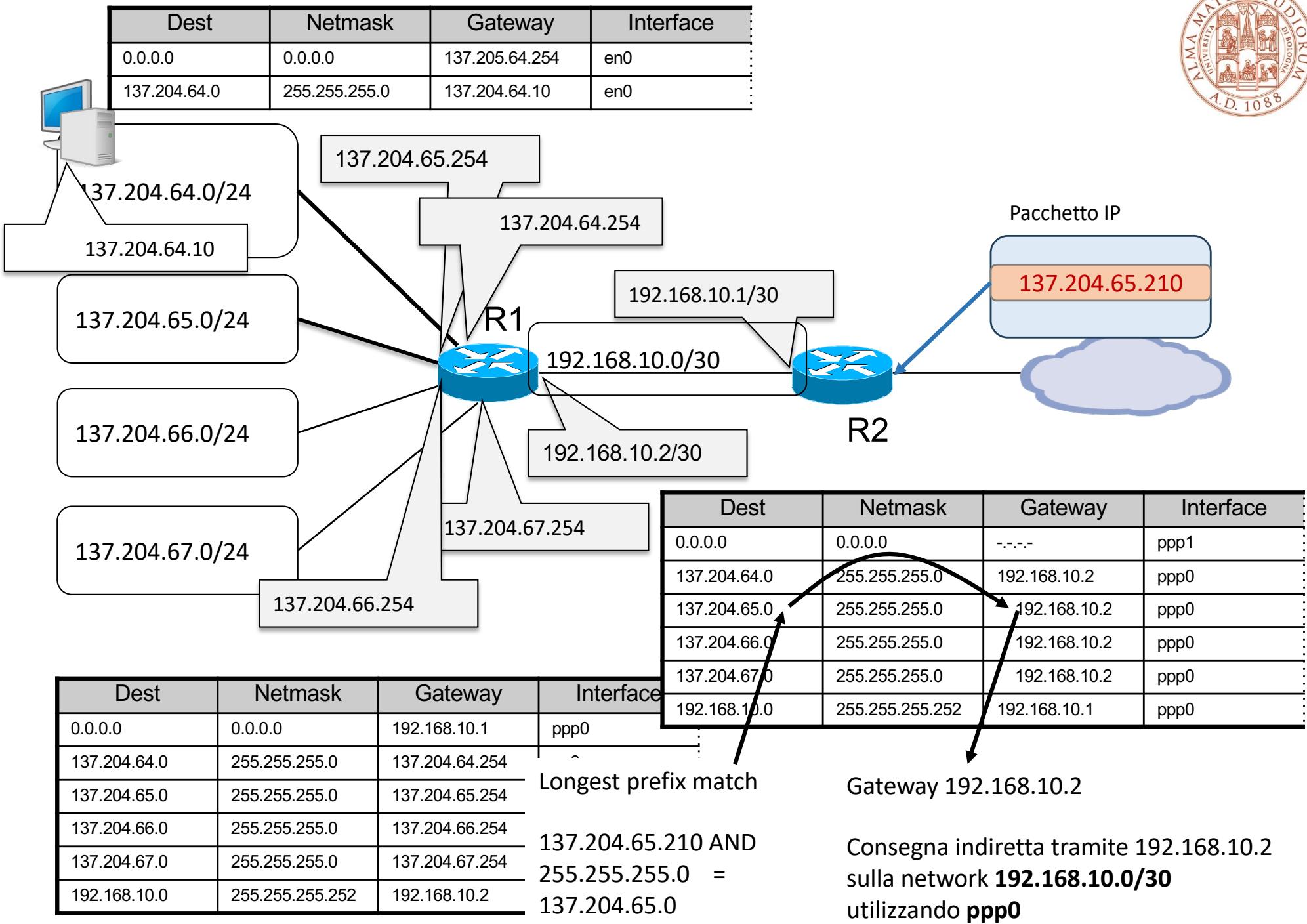


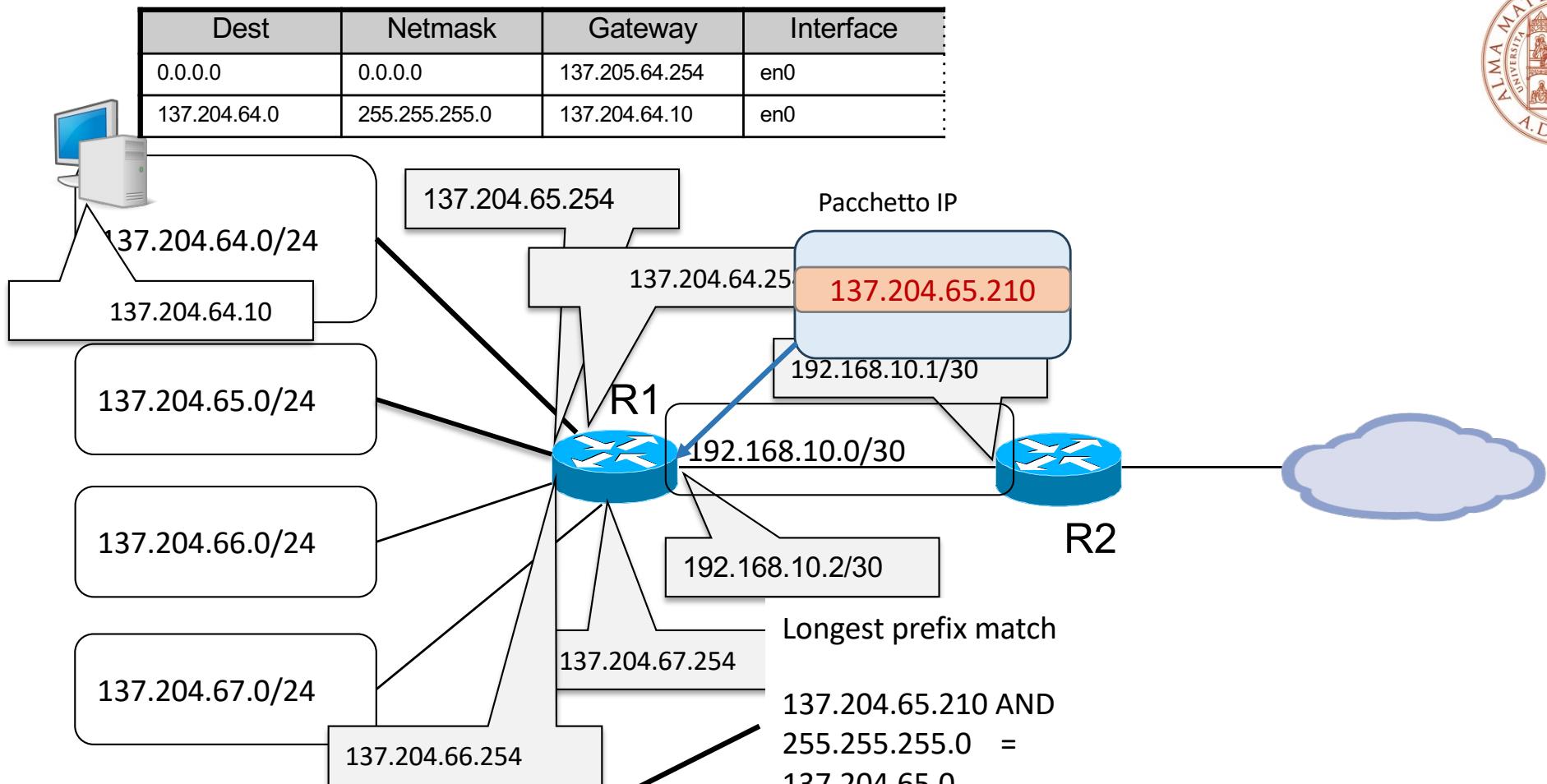
Esempio



Dest	Netmask	Gateway	Interface
0.0.0.0	0.0.0.0	192.168.10.1	ppp0
137.204.64.0	255.255.255.0	137.204.64.254	en0
137.204.65.0	255.255.255.0	137.204.65.254	en1
137.204.66.0	255.255.255.0	137.204.66.254	en2
137.204.67.0	255.255.255.0	137.204.67.254	en3
192.168.10.0	255.255.255.252	192.168.10.2	ppp0





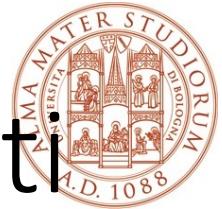


Router R1 Routing Table:

Dest	Netmask	Gateway	Interface
0.0.0.0	0.0.0.0	192.168.10.1	ppp0
137.204.64.0	255.255.255.0	137.204.64.254	en0
137.204.65.0	255.255.255.0	137.204.65.254	en1
137.204.66.0	255.255.255.0	137.204.66.254	en2
137.204.67.0	255.255.255.0	137.204.67.254	en3
192.168.10.0	255.255.255.252	192.168.10.2	ppp0

Gateway 137.204.65.254 me stesso

Consegna indiretta tramite
sulla network **137.204.65.0/24**
utilizzando **en1**



Analizziamo gli indirizzi delle 4 reti

- 137.204.64.0 il terzo byte è 01000000
- 137.204.65.0 il terzo byte è 01000001
- 137.204.66.0 il terzo byte è 01000010
- 137.204.67.0 il terzo byte è 01000011
 - I primi 2 byte ed i primi 6 bit del terzo byte sono comuni a tutte e quattro le network. Se usiamo NETMASK=255.255.252.0

10001001.11001100.01000000.**xxxxxxxx**
11111111.11111111.11111100.00000000
10001001.11001100.01000000.00000000
137 204 **64**

10001001.11001100.01000010.**xxxxxxxx**
11111111.11111111.11111100.00000000
10001001.11001100.01000000.00000000
137 204 **66**

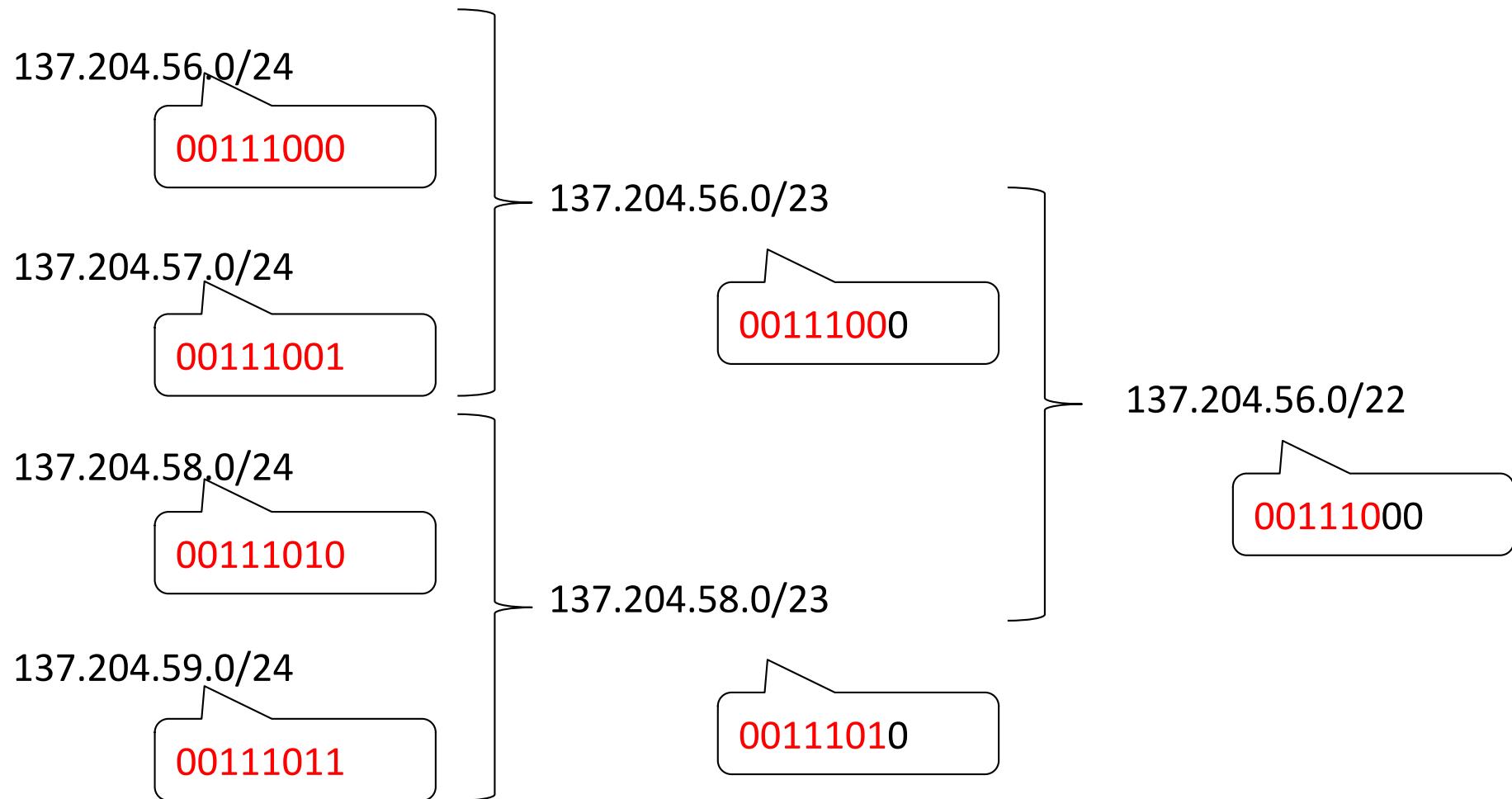
10001001.11001100.01000001.**xxxxxxxx**
11111111.11111111.11111100.00000000
10001001.11001100.01000000.00000000
137 204 **65**

10001001.11001100.01000011.**xxxxxxxx**
11111111.11111111.11111100.00000000
10001001.11001100.01000000.00000000
137 204 **67**

- Otteniamo il medesimo risultato in tutti e quattro i casi:
 - Il prefisso di rete è sempre 137.204.**64**.0



Un altro esempio



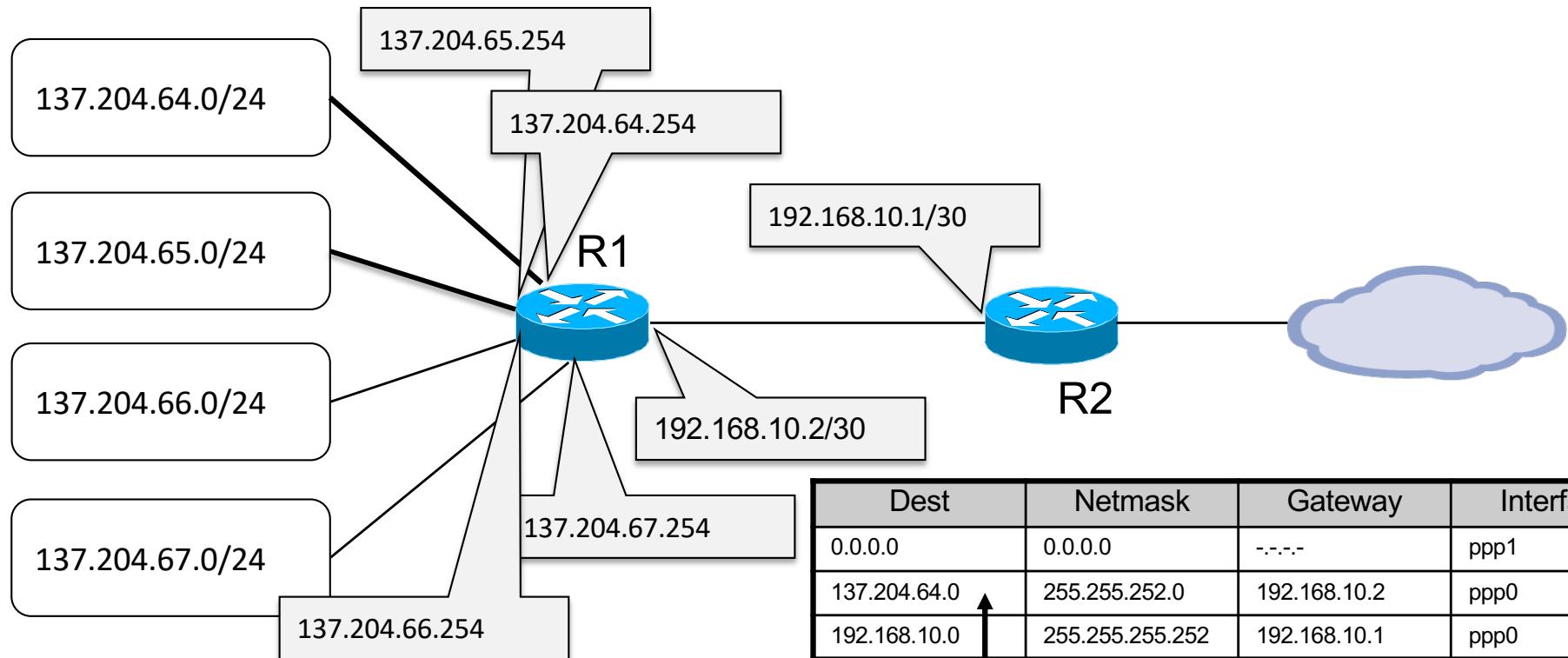


Semplificazione delle tabelle

- È necessario che R2 conosca il dettaglio di come le reti sono connesse a R1?
 - R2 invia comunque i datagrammi tramite R1
 - È sufficiente un' informazione più “riassuntiva”
- I route verso le 4 network possono essere aggregate in una sola
- R2 vede le 4 reti come una sola
 - Il gateway verso quelle destinazioni è R1



Aggregazione



Dest	Netmask	Gateway	Interface
0.0.0.0	0.0.0.0	192.168.10.1	ppp0
137.204.64.0	255.255.255.0	137.204.64.254	en0
137.204.65.0	255.255.255.0	137.204.65.254	en1
137.204.66.0	255.255.255.0	137.204.66.254	en2
137.204.67.0	255.255.255.0	137.204.67.254	en3
192.168.10.0	255.255.255.252	192.168.10.2	ppp0

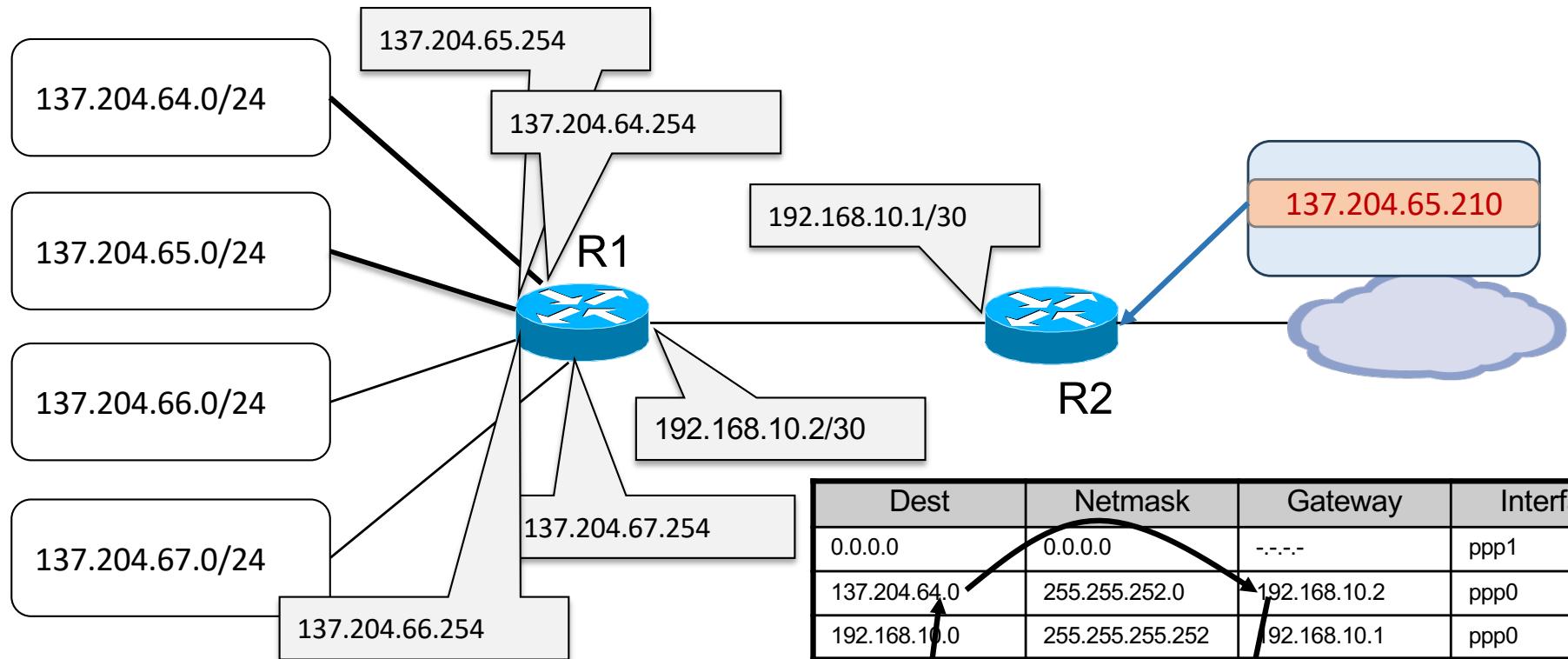
Le network

137.204.64.0/24
 137.204.65.0/24
 137.204.66.0/24
 137.204.66.0/24

Vengono aggregate in un'unica destinazione
137.204.64.0/22



Aggregazione



Dest	Netmask	Gateway	Interface
0.0.0.0	0.0.0.0	192.168.10.1	ppp0
137.204.64.0	255.255.255.0	137.204.64.254	
137.204.65.0	255.255.255.0	137.204.65.254	
137.204.66.0	255.255.255.0	137.204.66.254	
137.204.67.0	255.255.255.0	137.204.67.254	
192.168.10.0	255.255.255.252	192.168.10.2	

Longest prefix match

137.204.65.210 AND
255.255.252.0 =
137.204.64.0

Gateway 192.168.10.2

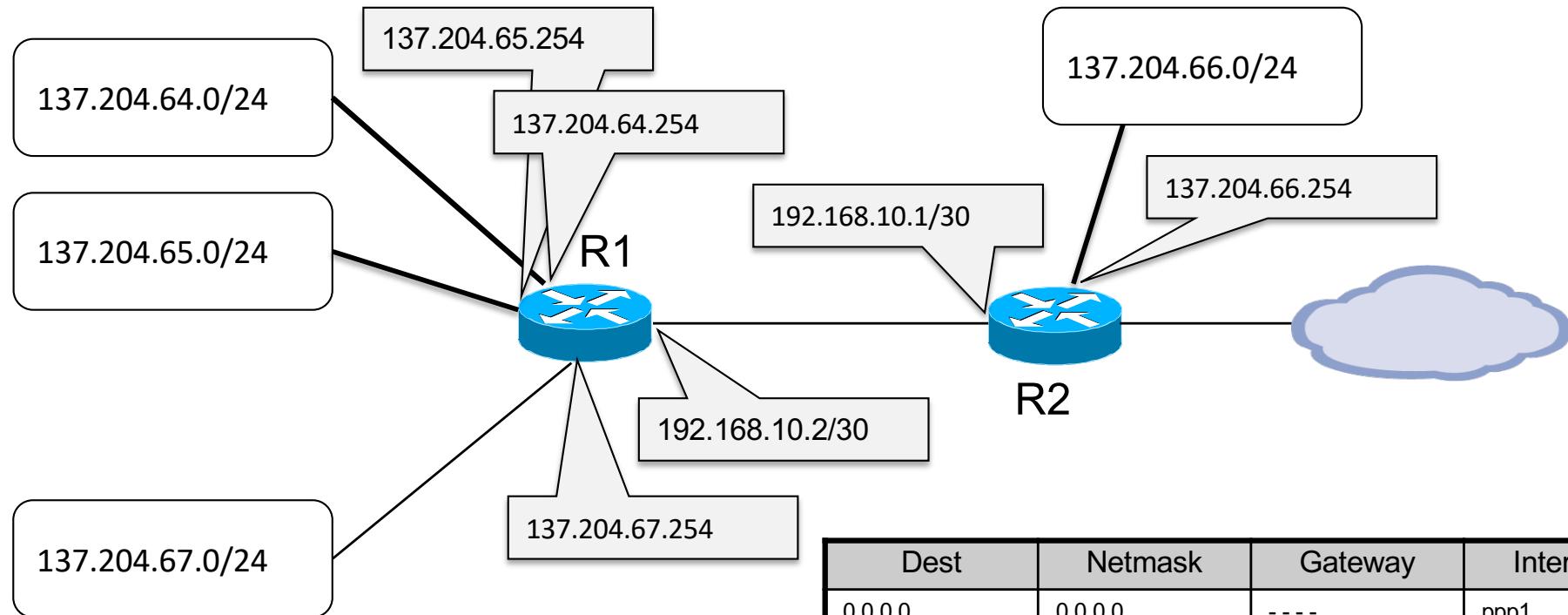
Consegna indiretta tramite 192.168.10.2
sulla network **192.168.10.0/30**
utilizzando **ppp0**



Perché ordinare i route?

- Dare priorità alle route più specifiche
- L'ordinamento in funzione della Netmask decrescente garantisce di considerare in ordine
 - singoli host
 - reti piccole
 - reti grandi
- È possibile implementare eccezioni a regole generali che possono convivere nella medesima tabella

Eccezioni



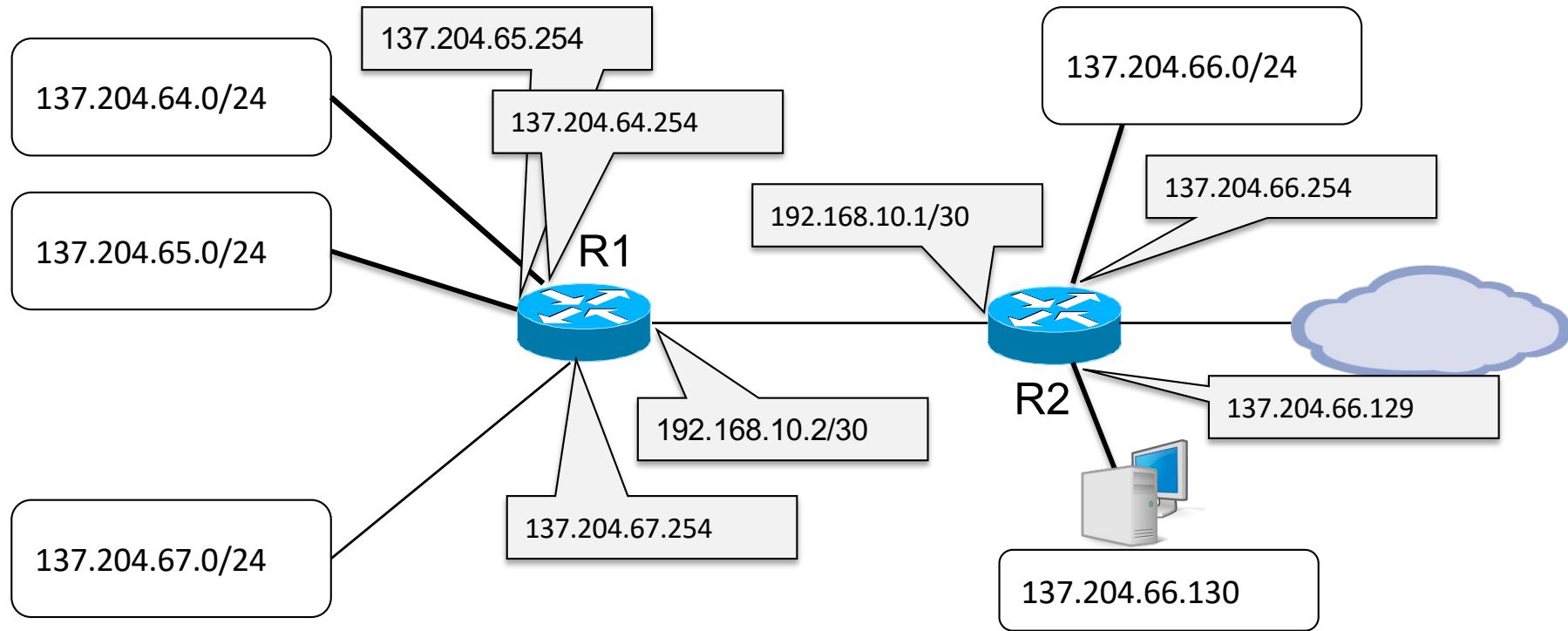
Dest	Netmask	Gateway	Interface
0.0.0.0	0.0.0.0	192.168.10.1	ppp0
137.204.64.0	255.255.255.0	137.204.64.254	en0
137.204.65.0	255.255.255.0	137.204.65.254	en1
137.204.67.0	255.255.255.0	137.204.67.254	en3
192.168.10.0	255.255.255.252	192.168.10.2	ppp0

Dest	Netmask	Gateway	Interface
0.0.0.0	0.0.0.0	-.-.-	ppp1
137.204.64.0	255.255.252.0	192.168.10.2	ppp0
137.204.66.0	255.255.255.0	137.204.66.254	en0
192.168.10.0	255.255.255.252	192.168.10.1	Ppp0

La rotta per 137.204.66.0/24 viene cancellata e non è necessario modificarla perché adesso viene assorbita dalla rotta di default

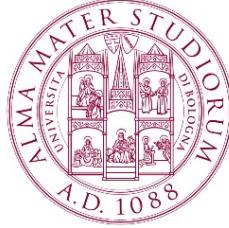


Eccezioni



Dest	Netmask	Gateway	Interface
0.0.0.0	0.0.0.0	192.168.10.1	ppp0
137.204.64.0	255.255.255.0	137.204.64.254	en0
137.204.65.0	255.255.255.0	137.204.65.254	en1
137.204.67.0	255.255.255.0	137.204.67.254	en3
192.168.10.0	255.255.255.252	192.168.10.2	ppp0

Dest	Netmask	Gateway	Interface
0.0.0.0	0.0.0.0	-.-.-	ppp1
137.204.64.0	255.255.252.0	192.168.10.2	ppp0
137.204.66.0	255.255.255.0	137.204.66.254	en0
192.168.10.0	255.255.255.252	192.168.10.1	Ppp0
137.204.66.128	255.255.255.252	137.204.66.129	en1



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Classless VS Classfull la logica degli indirizzi IP



IP e netmask

- Il numero IP ha valore assoluto in rete
 - Un numero IP pubblico deve essere unico su Internet
 - I numeri IP sorgente e destinazione caratterizzano il datagramma in quanto parte della sua intestazione
- La netmask è relativa al singolo nodo
 - Non viene trasportata nell'intestazione del datagramma
 - È parte della tabella di routing dei singoli nodi
 - Ai medesimi indirizzi possono corrispondere netmask diverse in nodi diversi (route aggregation)
- È sempre stato così?
 - NO: inizialmente la suddivisione net-ID e host-ID era assoluta



Classe delle reti

- Durante la fase iniziale di Internet furono definite diverse “**classi**” di network differenziate per **dimensione**
 - La parte iniziale del Net-ID differenzia le classi
 - 0 classe A
 - 10 classe B
 - 110 classe C
 - La definizione delle classi è standard e quindi nota a tutti
 - I router riconoscono la classe di una rete dai primi bit dell’ indirizzo
 - Ricavano di conseguenza il Net-ID



Classi di indirizzi

Network ID

Host ID



Classe A



Classe B



Classe C



Classe D (multicast)



Classe E (sperimentale)

32 bit

Network ID :

identifica una rete IP

Host ID :

identifica i singoli calcolatori della rete



Intervalli di indirizzi

- Classe A: da 0.0.0.0 a 127.255.255.255
- Classe B: da 128.0.0.0 a 191.255.255.255
- Classe C: da 192.0.0.0 a 223.255.255.255
- Classe D: da 224.0.0.0 a 239.255.255.255
- Classe E: da 240.0.0.0 a 255.255.255.255
- Indirizzi riservati (RFC 1700)
 - 0.0.0.0 indica l' host corrente senza specificarne l' indirizzo
 - Host-ID **tutto a 0** viene usato per **indicare la rete**
 - Host-ID **tutto a 1** è l' indirizzo di **broadcast** per quella rete
 - 0.x.y.z indica un certo Host-ID sulla rete corrente senza specificare il Net-ID
 - 255.255.255.255 è l' indirizzo di broadcast su Internet
 - 127.x.y.z è il **loopback**, che redirige i datagrammi agli strati



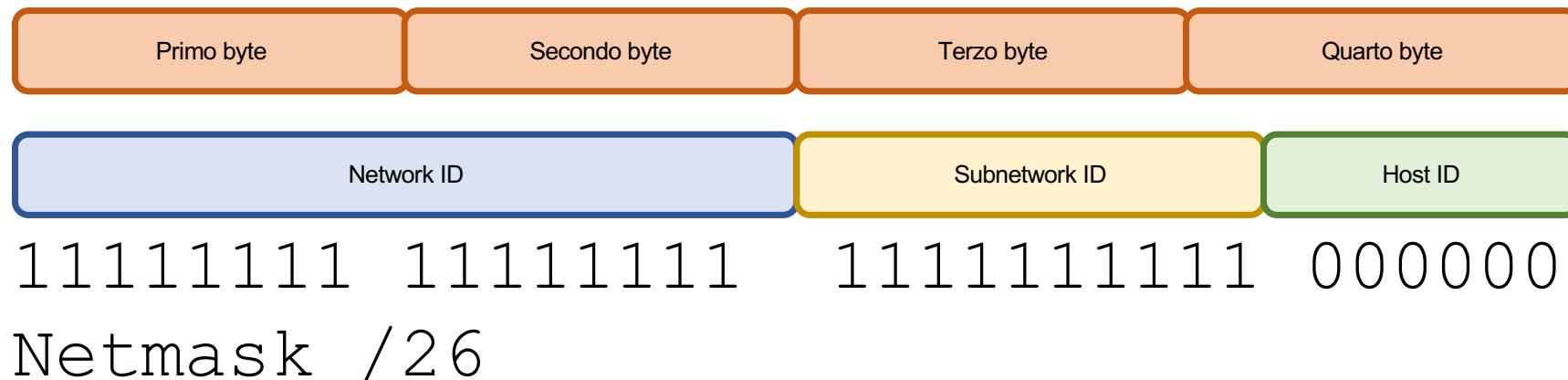
Le sottoreti

- A un'amministrazione è assegnata una network
 - L' amministrazione potrebbe essere suddivisa in sotto-amministrazioni *logicamente separate*
 - Converrebbe “*frammentare*” la network in “*sub-network*” da assegnare alle sotto-amministrazioni
- Si decide localmente una sotto-ripartizione Net/Host ID **indipendente dalle classi**
- Si frammenta l' Host-ID in due parti:
 - la prima identifica la sottorete (**subnet-ID**)
 - la seconda identifica i singoli host della sottorete
- La ripartizione deve essere *locale* e *reversibile*
 - Tutta Internet vede comunque una certa network come un' entità unitaria



Subnetting

- La suddivisione è locale alla singola interfaccia
 - Deve essere configurabile localmente
- Si personalizza la **Netmask**





Esempio: Università di Bologna

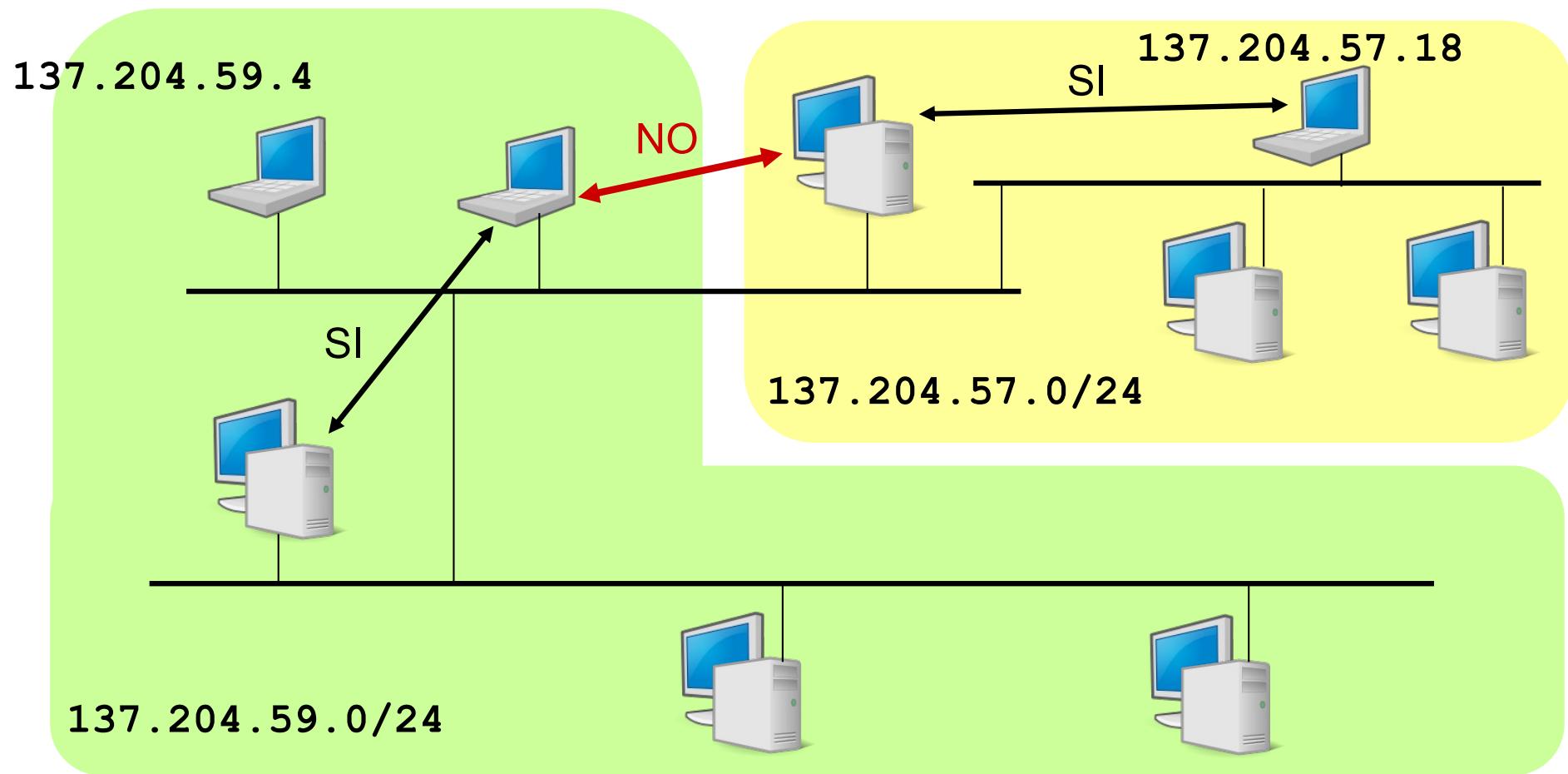
- Una network di classe B (137.204.0.0)
 - Numerose entità distinte nella stessa amministrazione
 - Facoltà, Dipartimenti, Centri di ricerca ecc.
 - Si suddivide la rete (network) in sottoreti (subnetwork)
- Il primo byte del Host-ID viene utilizzato come indirizzo di sottorete
 - Dalla network di classe B si ricavano 254 network della dimensione di una classe C

Netmask = 255.255.255.0



Subnetting

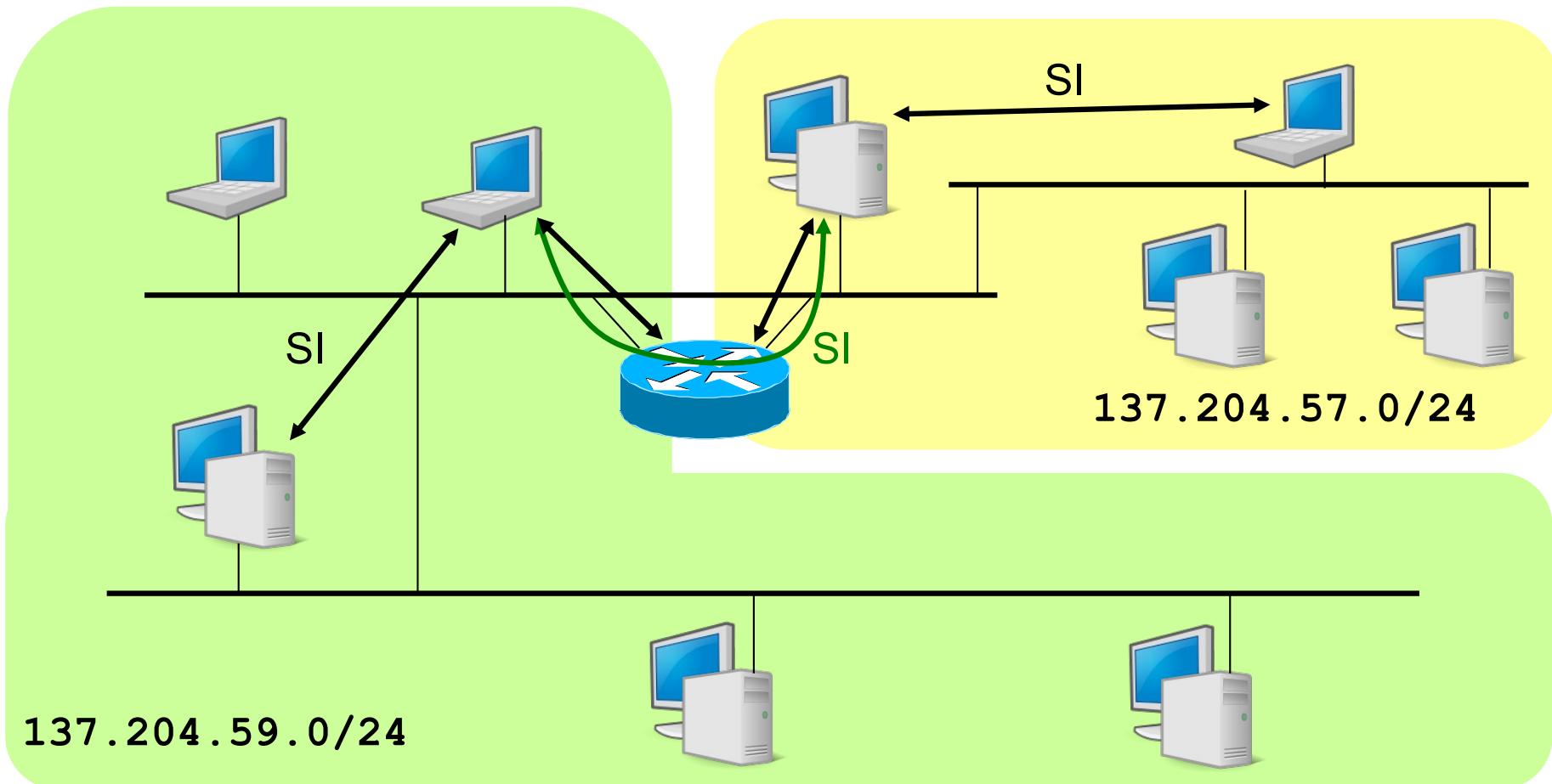
- Subnet diverse sono di fatto Network diverse e quindi non comunicano
- È necessario un gateway





Subnetting

- Il Gateway permette instradamento indiretto fra le Subnetwork





CIDR

- Con la grande diffusione di Internet la rigida suddivisione nelle 3 classi rendono l' instradamento poco flessibile e scalabile
- **CIDR** (RFC 1519) Classless InterDomain Routing
 - Si decide di rompere la logica delle classi nei router
 - La dimensione del Net-ID può essere qualunque
 - Le tabelle di routing devono **comprendere anche le Netmask**
 - Generalizzazione del subnetting/supernetting
 - reti IP definite da **Net-ID/Netmask**



Obiettivi del CIDR

- Allocazione di reti IP di dimensioni variabili
 - utilizzo più efficiente dello spazio degli indirizzi
- Accorpamento delle informazioni di routing
 - più reti contigue rappresentate da un' unica riga nelle tabelle di routing
- Miglioramento di due situazioni critiche
 - Limitatezza di reti di classe A e B
 - Crescita esplosiva delle dimensioni delle tabelle di routing



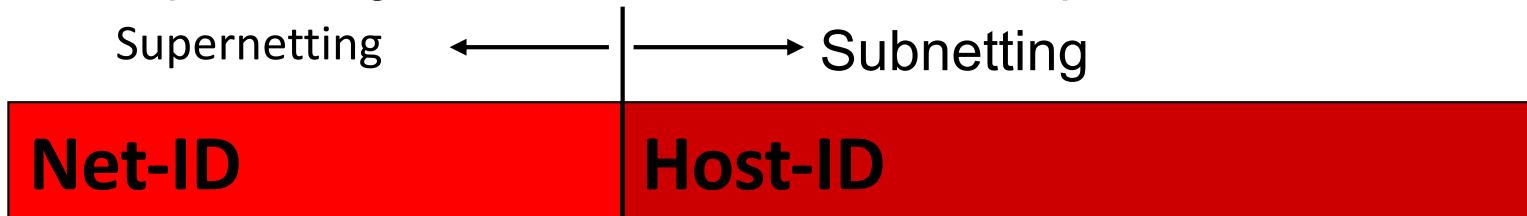
Supernetting

- Raggruppare più reti con indirizzi consecutivi
 - Indicarle nelle tabelle di routing con una sola entry accompagnata dalla opportuna Netmask
- Es. Un ente ha bisogno di circa 2000 indirizzi IP
 - una rete di classe B è troppo grande (64K indirizzi)
 - meglio 8 reti di classe C ($8 \times 256 = 2048$ indirizzi) dalla 194.24.0.0 alla 194.24.7.0
- **Supernetting:** si accorpano le 8 reti contigue in un'unica super-rete:
 - Identificativo: 194.24.0.0/21
 - Supernet mask: 255.255.248.0
 - Indirizzi: 194.24.0.1 – 194.24.7.254
 - Broadcast: 194.24.7.255



Supernetting

- Subnetting e Supernetting sono operazioni duali
 - Subnetting → **n** bit del Host-ID diventano parte del Net-ID
 - Supernetting → **n** bit del Net-ID diventano parte dell' Host-ID



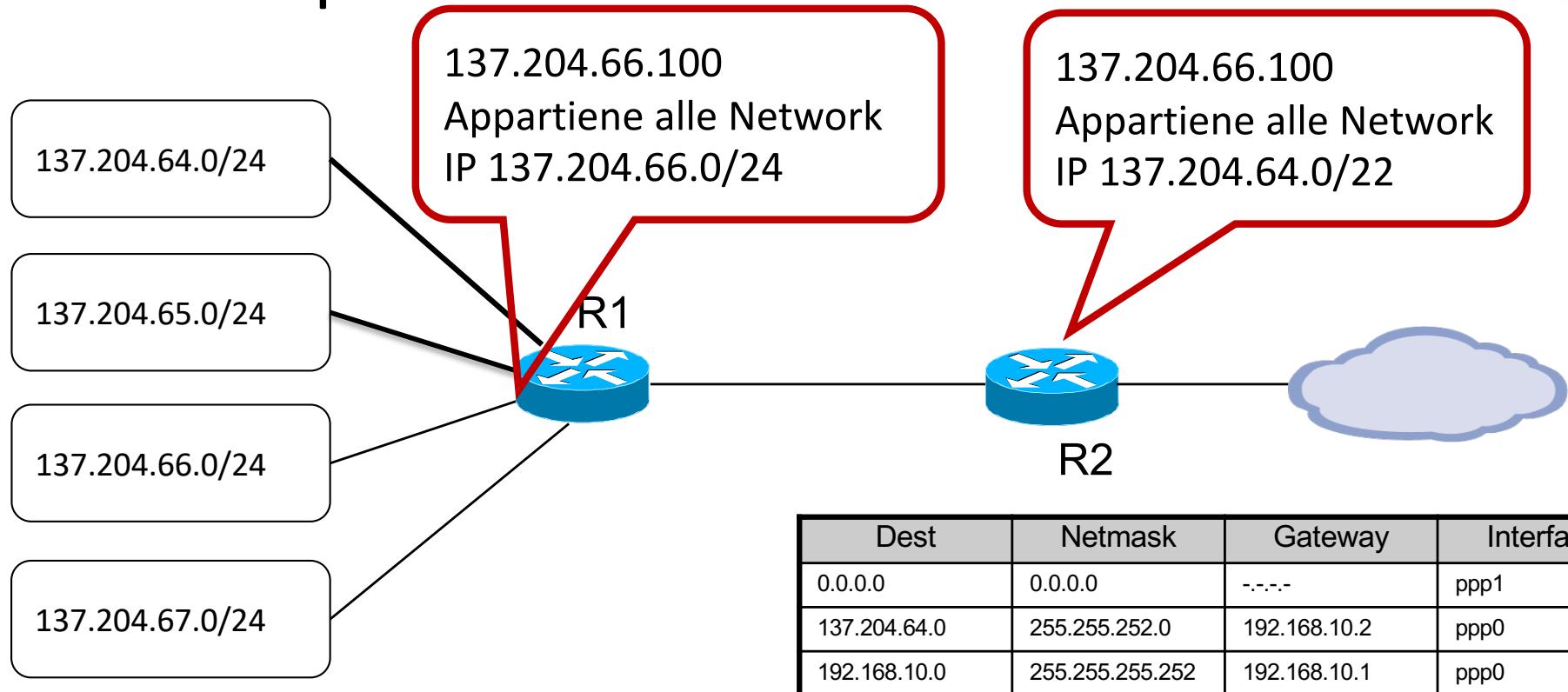
- Accorpamento di **N** reti IP (**N = 2ⁿ**)
 - **contigue**:
 - $194.24.0.0/24 + 194.24.1.0/24 = 194.24.0.0/23$
 - $194.24.0.0/24 + 194.24.2.0/24$ = non contigue
 - **allineate** secondo i multipli di 2^n
 - $194.24.0.0/24 + .1.0/24 + .2.0/24 + .3.0/24 = 194.24.0.0/22$
 - $194.24.2.0/24 + .3.0/24 + .4.0/24 + .5.0/24$ = non allineate



Oggi

- La distinzione fra Net-ID e Host-ID è locale funzione della Netmask
- Lo stesso indirizzo può essere interpretato in modo diverso in punti diversi della rete
- Tutte le tabelle di instradamento devo contenere la colonna delle Netmask

Esempio



Dest	Netmask	Gateway	Interface
0.0.0.0	0.0.0.0	192.168.10.1	ppp0
137.204.64.0	255.255.255.0	137.204.64.254	en0
137.204.65.0	255.255.255.0	137.204.65.254	en1
137.204.66.0	255.255.255.0	137.204.66.254	en2
137.204.67.0	255.255.255.0	137.204.67.254	en3
192.168.10.0	255.255.255.252	192.168.10.2	ppp0



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Pianificare la numerazione di reti IP

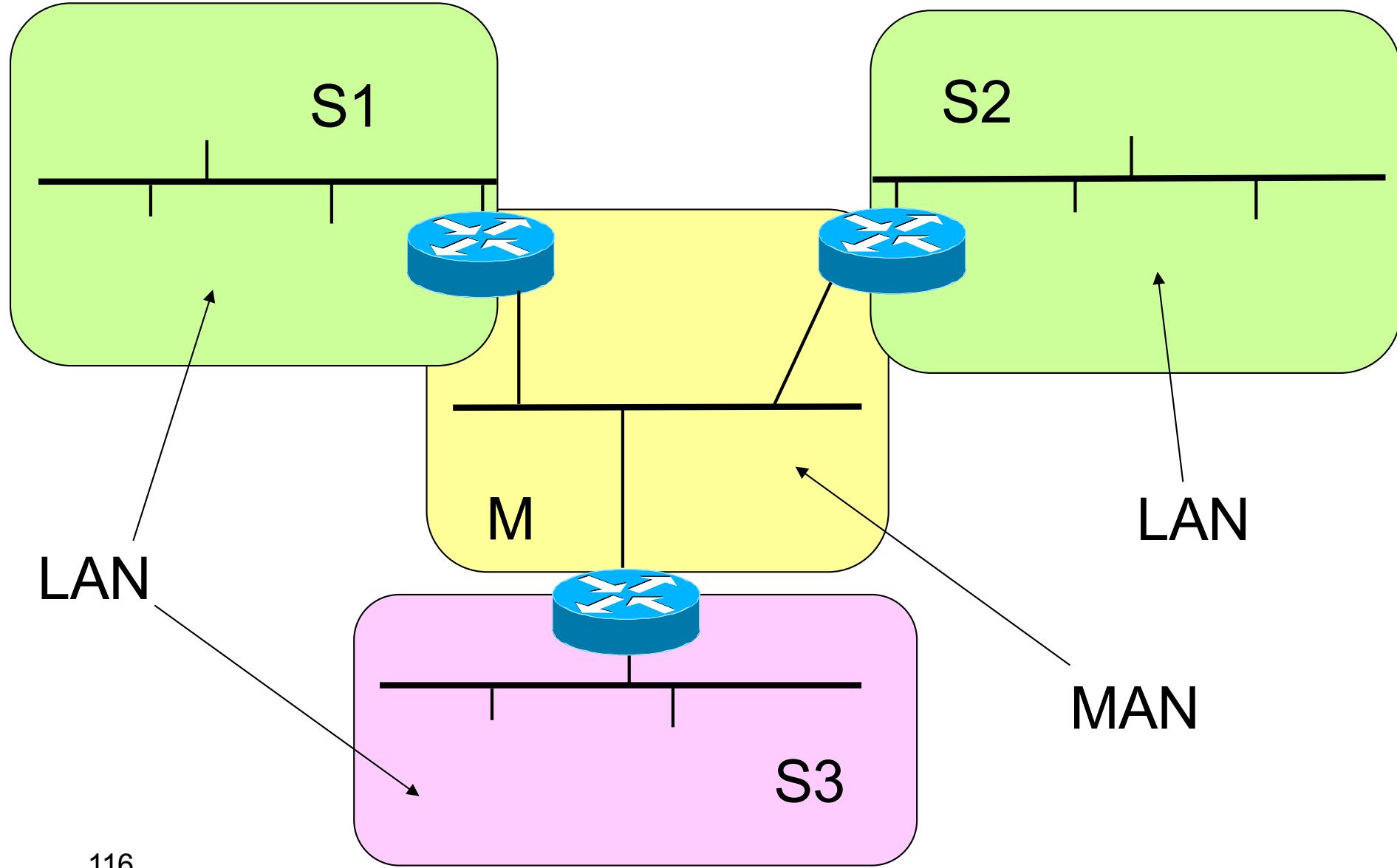


Esempio

- Un'azienda possiede tre siti distribuiti su una grande area urbana: S1, S2, S3.
- Ciascun sito aziendale è dotato di infrastrutture informatiche comprendenti, tra l'altro, una LAN ed un router di uscita verso il mondo esterno. Tutti i siti devono essere interconnessi tra loro con una rete a maglia completa.
- I siti sono così divisi:
 - S1, S2: 50 host
 - S3: 20 host
- Si richiede di progettare una rete di classe C a cui viene assegnato l'indirizzo 196.200.96.0/24 comprensiva della numerazione dei router, definendo le relative netmask



Architettura



La scelta della netmask

Ultimo byte netmask	# host	# subnets
00000000	254	1
10000000	126	2
11000000	62	4
11100000	30	8
11110000	14	16
11111000	6	32
11111100	2	64

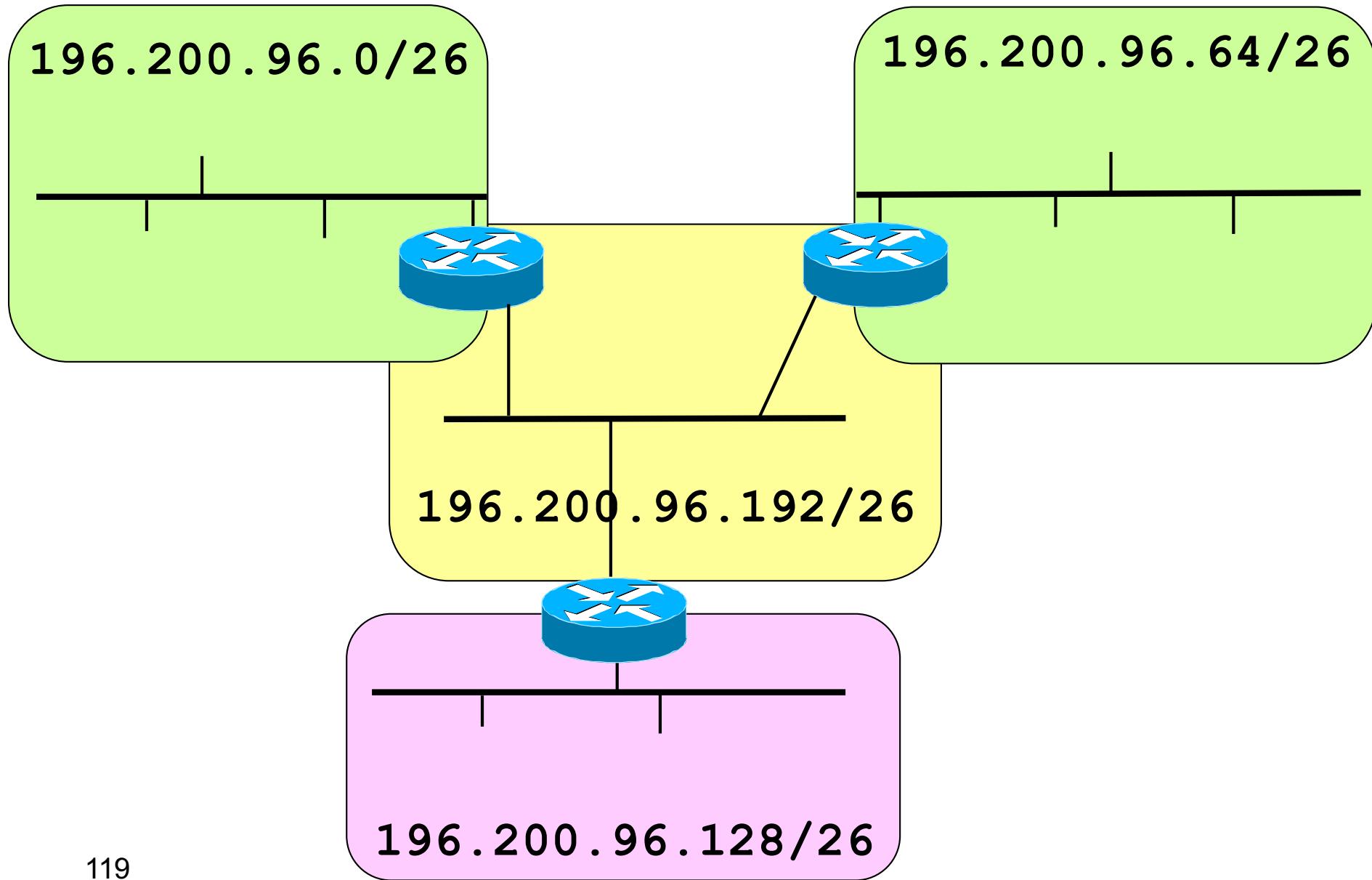


Soluzione 1

- Subnets: **196.200.96.0/26** (S1)
 196.200.96.64/26 (S2)
 196.200.96.128/26 (S3)
 196.200.96.192/26 (M)
- Netmask: **255.255.255.192**
- Broadcast: **196.200.96.63** (S1)
 196.200.96.127 (S2)
 196.200.96.191 (S3)
 196.200.96.255 (M)



Soluzione 1





Soluzione 1

- Routers LAN: 196.200.96.62 (S1)
196.200.96.126 (S2)
196.200.96.190 (S3)
 - Routers MAN: qualunque indirizzo tra:
196.200.96.193 e .254 (M)
 - IP Hosts: qualunque indirizzo tra:
196.200.96.1 e .61 (S1)
196.200.96.65 e .125 (S2)
196.200.96.129 e .189 (S3)

Scelta di netmask diverse

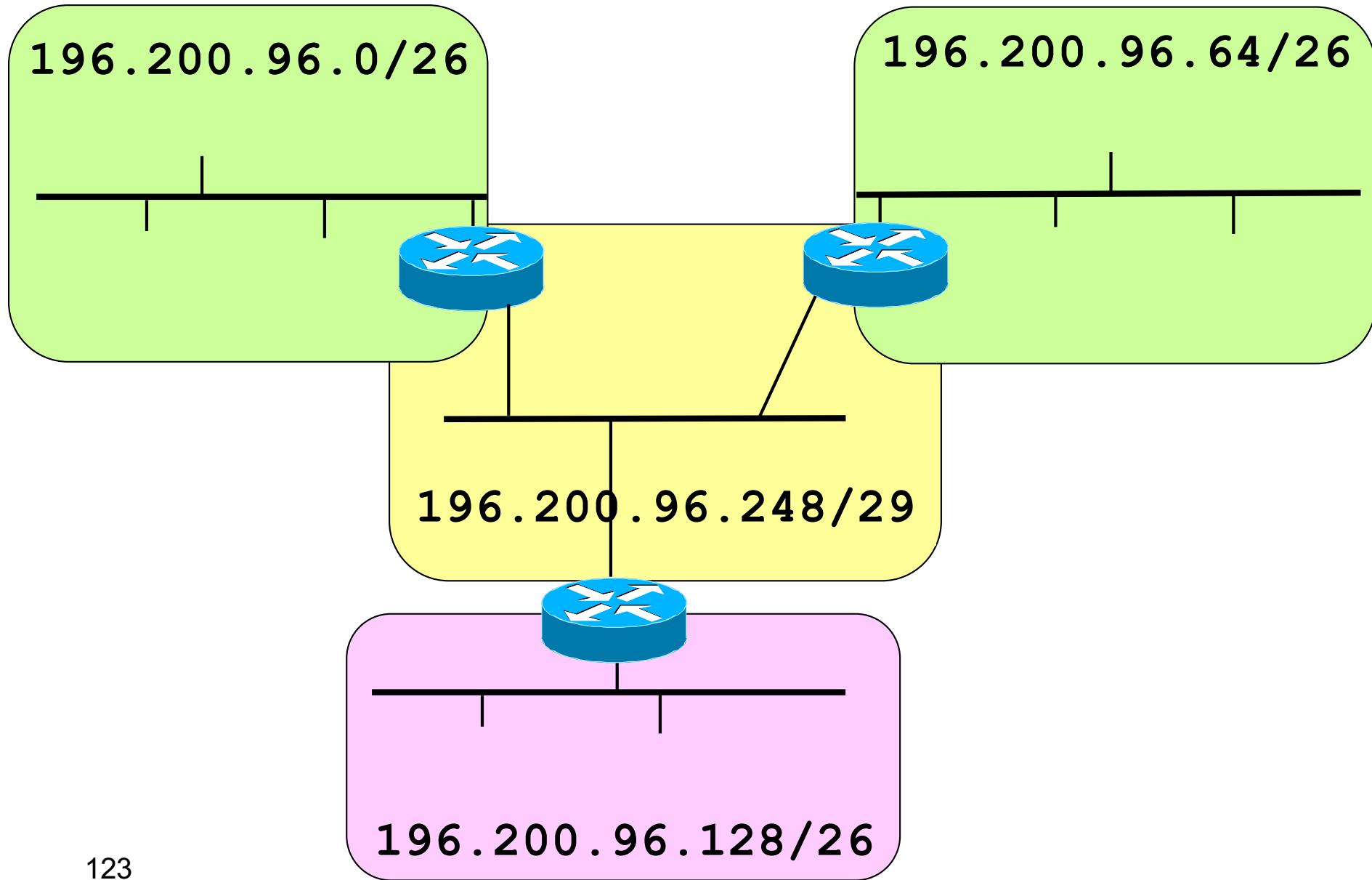
Ultimo byte netmask	# host	# subnets
00000000	254	1
10000000	126	2
11000000	62	4
11100000	30	8
11110000	14	16
11111000	6	32
11111100	2	64

Soluzione 2

Subnet	# host	Indirizzi	Broadcast
196.200.96.0/26	62	1 – 62	63
196.200.96.64/26	62	65 – 126	127
196.200.96.128/27	30	129 – 158	159
196.200.96.160/27	30	161 – 190	191
196.200.96.192/27	30	193 – 222	223
196.200.96.224/29	6	225 – 230	231
196.200.96.232/29	6	233 – 238	239
196.200.96.240/29	6	241 – 246	247
196.200.96.248/29	6	249 – 254	255



Soluzione 1



Scelta di netmask diverse

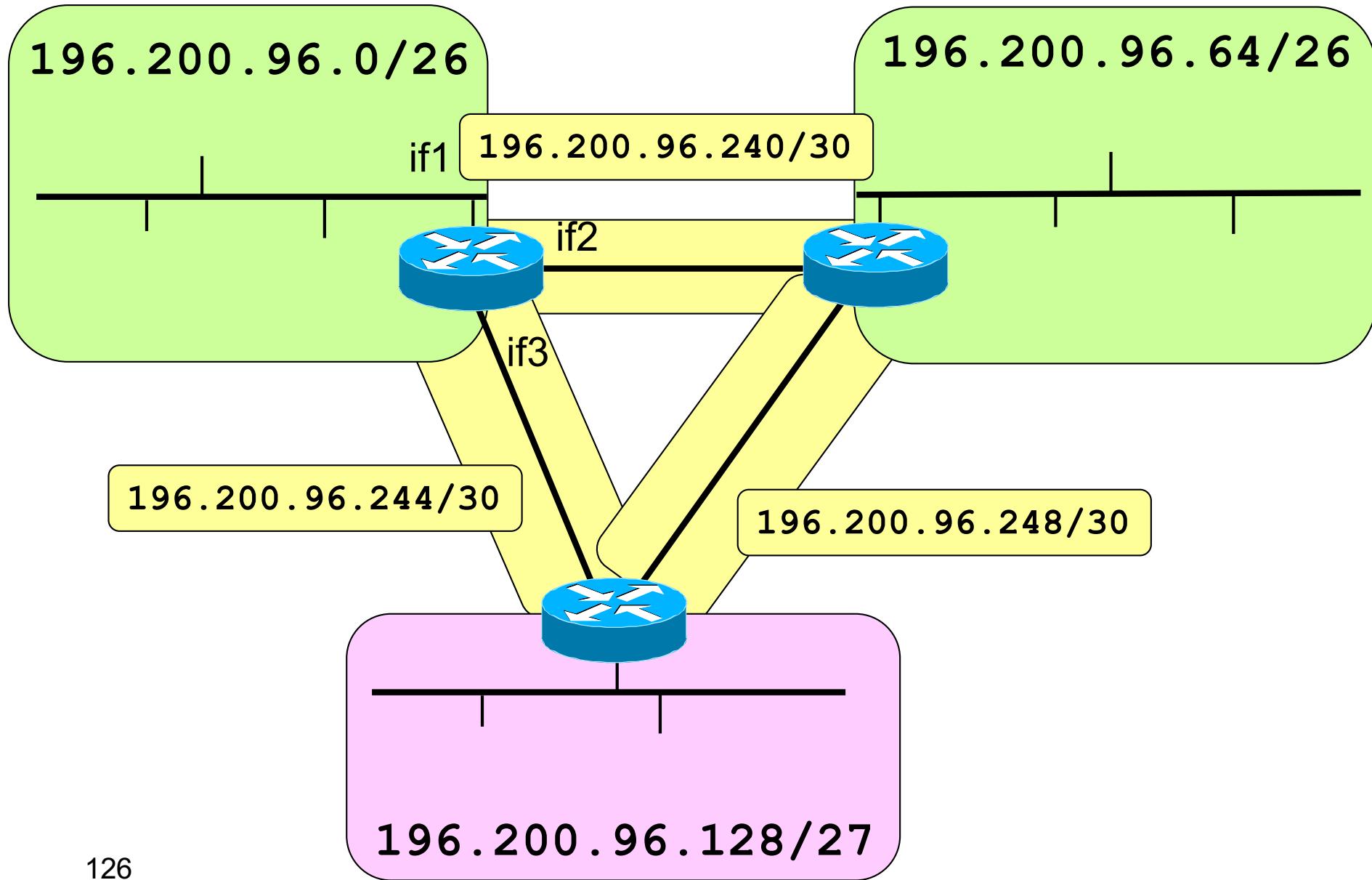
Ultimo byte netmask	# host	# subnets
00000000	254	1
10000000	126	2
11000000	62	4
11100000	30	8
11110000	14	16
11111000	6	32
11111100	2	64

Soluzione 3

Subnet	# host	Indirizzi	Broadcast
196.200.96.0/26	62	1 – 62	63
196.200.96.64/26	62	65 – 126	127
196.200.96.128/27	30	129 – 158	159
196.200.96.160/27	30	161 – 190	191
196.200.96.192/27	30	193 – 222	223
196.200.96.224/28	14	225 – 238	239
196.200.96.240/30	2	241 – 242	243
196.200.96.244/30	2	245 – 246	247
196.200.96.248/30	2	249 – 250	251
196.200.96.252/30	2	253 – 254	255



Soluzione 3



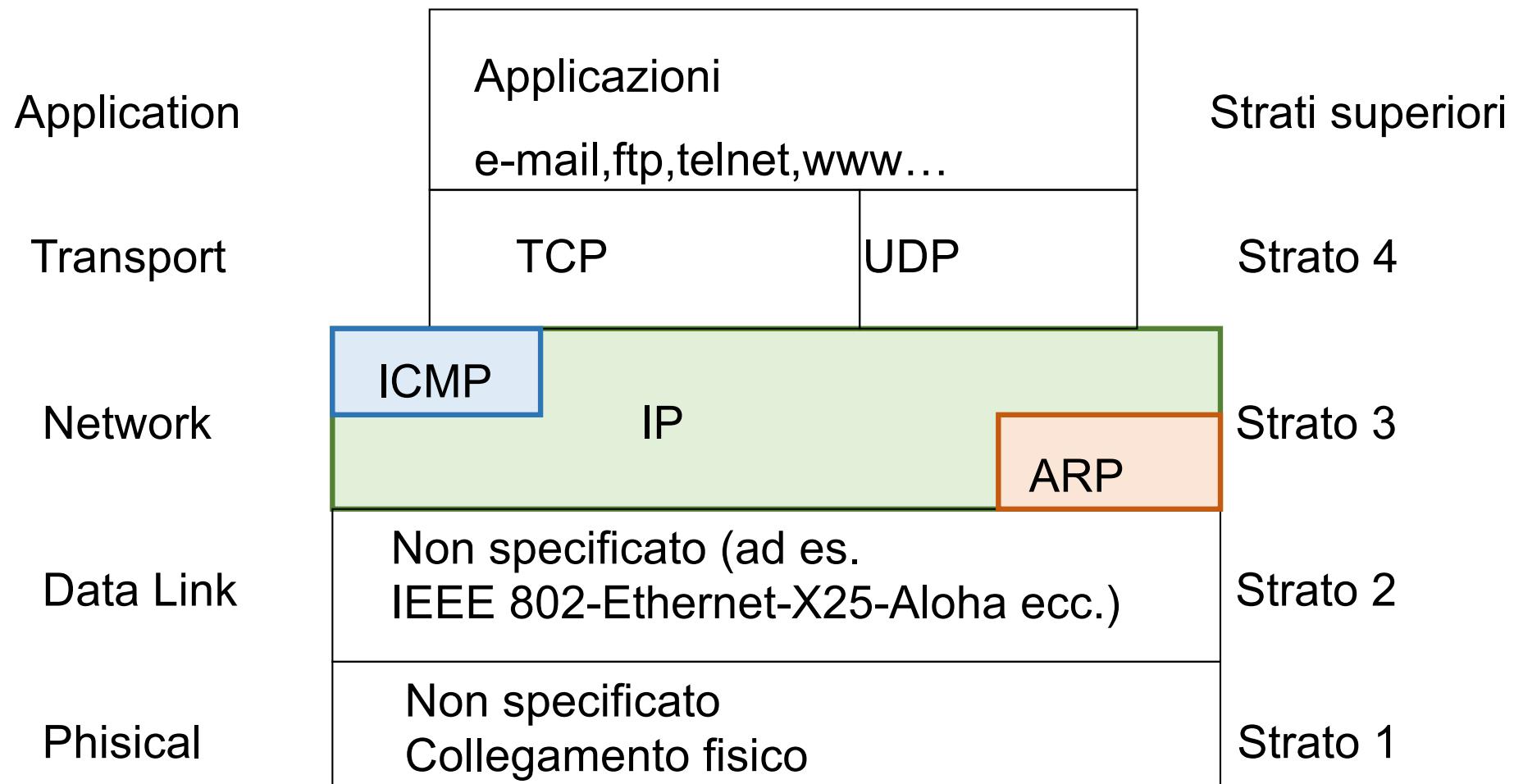


ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Il protocollo ICMP



Architettura



Il protocollo IP...

- offre un servizio di tipo best effort
 - non garantisce la corretta consegna dei datagrammi
 - se necessario si affida a protocolli affidabili di livello superiore (TCP)
- è comunque necessario un protocollo di controllo
 - gestione di situazioni anomale
 - notifica di errori o di irraggiungibilità della destinazione
 - scambio di informazioni sulla rete

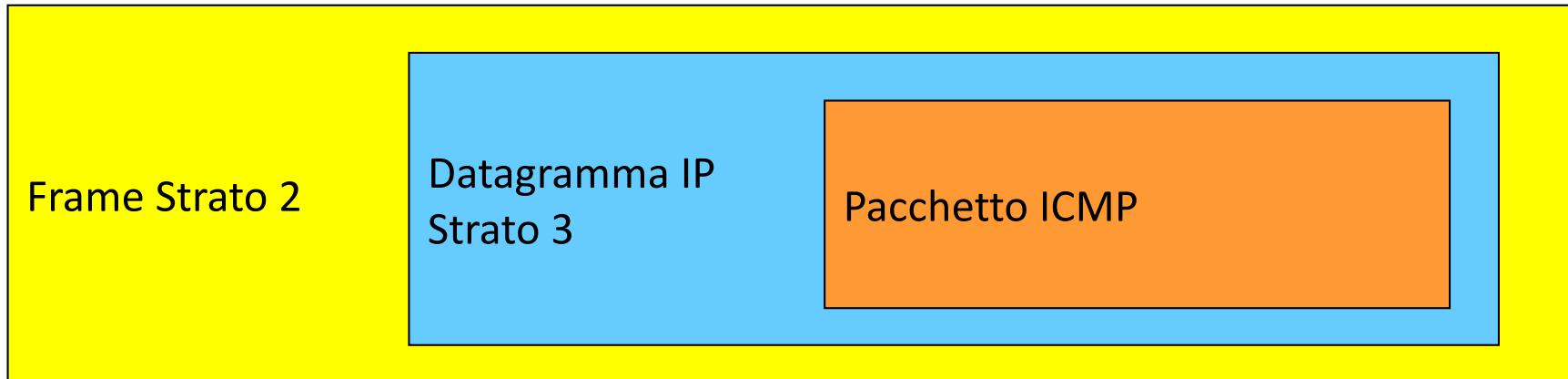
→ ICMP (Internet Control Message Protocol)

- ICMP segnala solamente errori e malfunzionamenti, ma non esegue alcuna correzione
- ICMP **non rende affidabile** IP



ICMP

- **Internet Control Message Protocol (RFC 792)**
svolge funzioni di controllo per IP
 - IP usa ICMP per la gestione di situazioni anomale, per cui ICMP offre un servizio ad IP
 - i pacchetti ICMP sono incapsulati in datagrammi IP, per cui ICMP è anche utente IP





Pacchetto ICMP

IP header	20 - 60 byte
Message Type	1 byte
Message Code	1 byte
Checksum	2 byte
Additional Fields (optional)	variabile
Data	variabile

- **Type** definisce il tipo di messaggio ICMP
 - messaggi di errore
 - messaggi di richiesta di informazioni
- **Code** descrive il tipo di errore e ulteriori dettagli
- **Checksum** controlla i bit errati nel messaggio ICMP
- **Add. Fields** dipendono dal tipo di messaggio ICMP
- **Data** intestazione e parte dei dati del datagramma che ha generato l'errore



Tipi di errori

- **Destination Unreachable** (Type = 3)
 - Generato da un gateway quando la sottorete o l'host non sono raggiungibili
 - Generato da un host quando si presenta un errore sull'indirizzo dell'entità di livello superiore a cui trasferire il datagramma
- Codici errore di Destination Unreachable
 - 0 = sottorete non raggiungibile
 - 1 = host non raggiungibile
 - 2 = protocollo non disponibile
 - 3 = porta non disponibile
 - 4 = frammentazione necessaria ma bit don't fragment settato



Tipi di errori

- Time Exceeded (Type = 11)
 - generato da un router quando il Time-to-Live di un datagramma si azzera ed il datagramma viene distrutto (Code = 0)
 - generato da un host quando un timer si azzera in attesa dei frammenti per riassemblare un datagramma ricevuto in parte (Code = 1)
- Source Quench (Type = 4)
 - i datagrammi arrivano troppo velocemente rispetto alla capacità di essere processati: l'host sorgente deve ridurre la velocità di trasmissione (obsoleto)
- Redirect (Type = 5)
 - generato da un router per indicare all'host sorgente un'altra strada più conveniente per raggiungere l'host destinazione



Informazioni

- Echo (Type = 8)
 - Echo Reply (Type = 0)
 - l'host sorgente invia la richiesta ad un altro host o ad un gateway
 - la destinazione deve rispondere immediatamente
 - metodo usato per determinare lo stato di una rete e dei suoi host, la loro raggiungibilità e il tempo di transito nella rete
 - Additional Fields:
 - Identifier: identifica l'insieme degli echo appartenenti allo stesso test
 - Sequence Number: identifica ciascun echo nell'insieme
 - Optional Data: usato per inserire eventuali dati di verifica



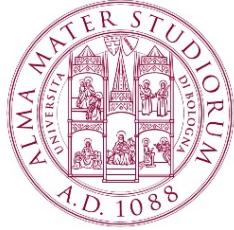
Informazioni

- Timestamp Request (Type = 13)
- Timestamp Reply (Type = 14)
 - l'host sorgente invia all'host destinazione un Originate Timestamp che indica l'istante in cui la richiesta è partita
 - l'host destinazione risponde inviando un
 - Receive Timestamp che indica l'istante in cui la richiesta è stata ricevuta
 - Transmit Timestamp che indica l'istante in cui la risposta è stata inviata
 - serve per valutare il tempo di transito nella rete, al netto del tempo di processamento = $T_{Transmit} - T_{Receive}$



Informazioni

- Address Mask Request (Type = 17)
- Address Mask Reply (Type = 18)
inviato dall'host sorgente all'indirizzo di broadcast (255.255.255.255) per ottenere la subnet mask da usare dopo aver ottenuto il proprio indirizzo IP tramite RARP o BOOTP
- Router Solicitation (Type = 10)
- Router Advertisement (Type = 9)
utilizzato per localizzare i router connessi alla rete



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

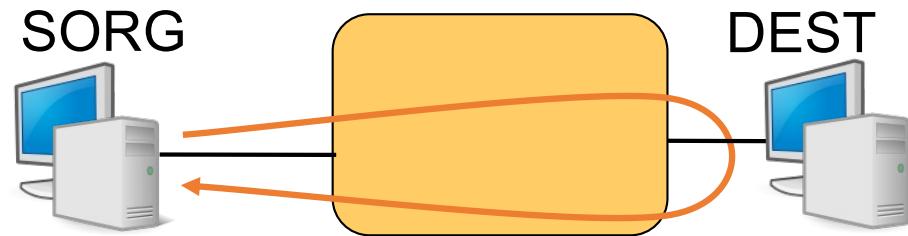
Applicazioni di ICMP



Comando PING

ping DEST

Permette di controllare se l' host DEST è raggiungibile o meno da SORG



- SORG invia a DEST un pacchetto ICMP di tipo “echo”
- Se l'host DEST è raggiungibile da SORG, DEST risponde inviando indietro un pacchetto ICMP di tipo “echo reply”



Opzioni

- **-n N** permette di specificare quanti pacchetti inviare (un pacchetto al secondo)
- **-l M** specifica la dimensione in byte di ciascun pacchetto
- **-t Ctrl-C** esegue **ping** finché interrotto con **Ctrl-C**
- **-a** traduce l' indirizzo IP in nome DNS
- **-f** setta il bit *don't fragment* a 1
- **-i T** setta *time-to-live* = **T**
- **-w T_{out}** specifica un timeout in millisecondi
- Per maggiori informazioni consultare l' help: **ping /?**

Comando PING – Output

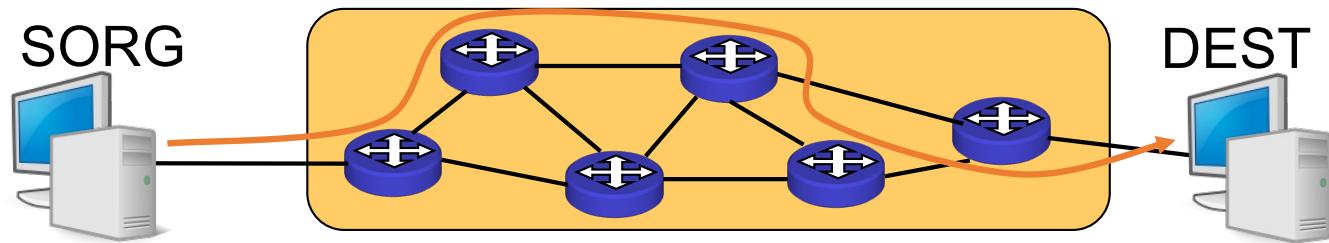
L' output mostra

- la dimensione del pacchetto “echo reply”
- l' indirizzo IP di DEST
- il numero di sequenza della risposta (solo UNIX-LINUX)
- il “time-to-live” (TTL)
- il “round-trip time” (RTT)
- alcuni risultati statistici: N° pacchetti persi, MIN, MAX e media del RTT

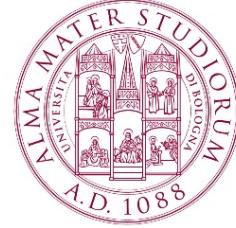
Comando TRACEROUTE

tracert DEST

Permette di conoscere il percorso seguito dai pacchetti inviati da SORG e diretti verso DEST



- SORG invia a DEST una serie di pacchetti **ICMP** di tipo **ECHO** con un **TIME-TO-LIVE (TTL)** progressivo da **1 a 30** (per default)
- Ciascun nodo intermedio decrementa **TTL**
- Il nodo che rileva **TTL = 0** invia a SORG un pacchetto **ICMP** di tipo **TIME EXCEEDED**
- SORG costruisce una lista dei nodiattraversati fino a DEST
- L'output mostra il **TTL**, il nome **DNS** e l'indirizzo **IP** dei nodi intermedi ed il **ROUND-TRIP TIME (RTT)**



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Gestione della numerazione



Dispositivi di rete

- DHCP
 - Permette ad un Host di ottenere una configurazione IP
- Packet Filter
 - Permette/blocca l' invio di pacchetti da/verso determinati indirizzi
 - Protegge la rete dal traffico "vagante"
- Application Layer Gateway (ALG) / Proxy
 - Controlla la comunicazione a livello applicativo
- Firewall
 - Combinazione dei dispositivi descritti sopra
 - Protegge le risorse interne da accessi esterni
- Network Address Translator (NAT)
 - Riduce la richiesta dello spazio di indirizzamento Internet
 - Nasconde gli indirizzi IP interni
 - Esegue un packet filtering per il traffico sconosciuto



DHCP – RFC 2131,2132

Dynamic Host Configuration Protocol

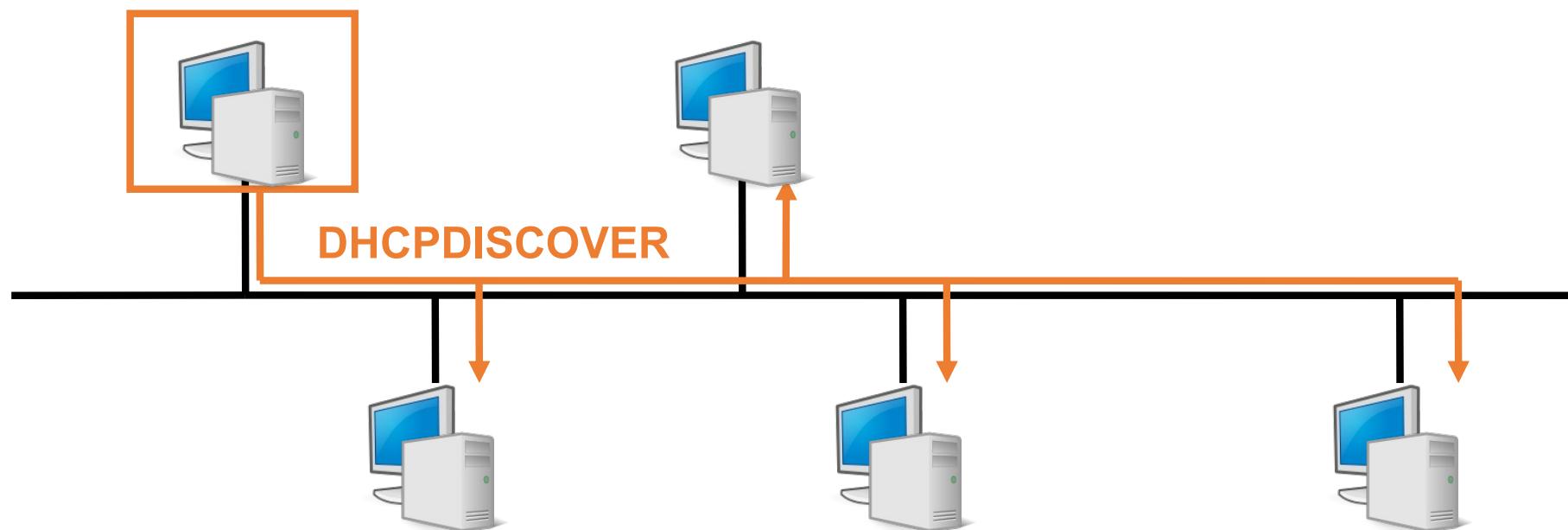
Configurazione **automatica** e **dinamica** di

- Indirizzo IP
- Netmask
- Broadcast
- Host name
- Default gateway
- Server DNS

Server su porta **67** UDP

DHCP – 1

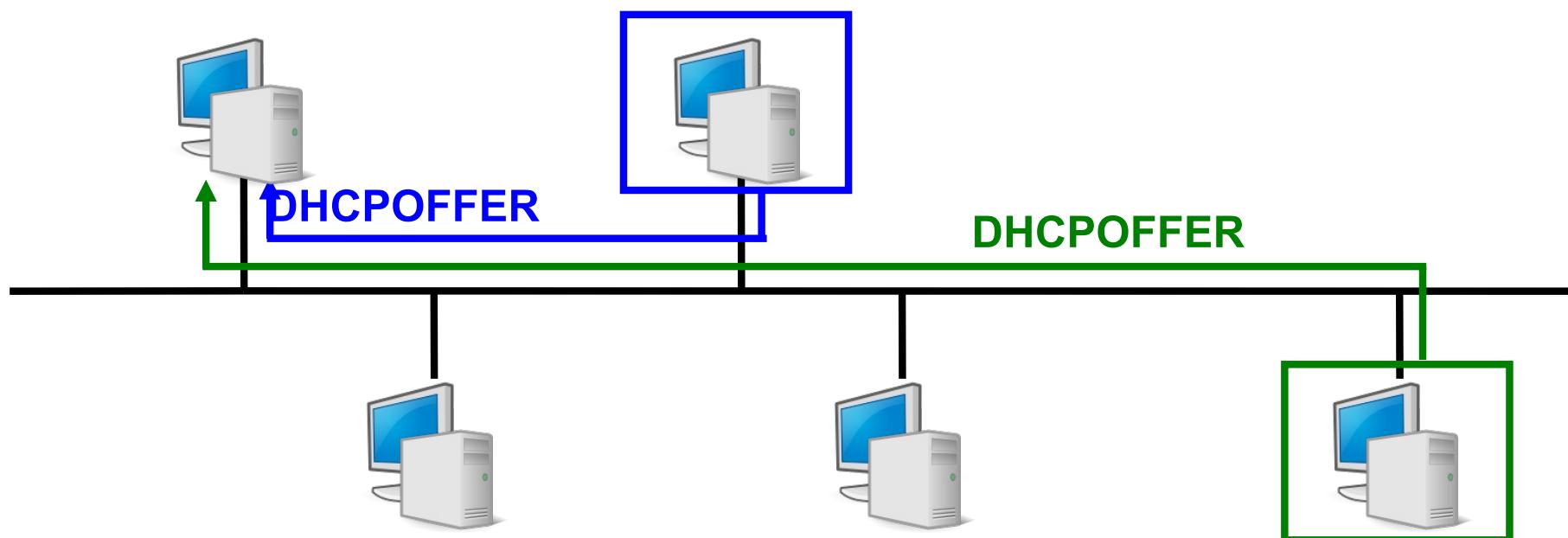
- Quando un host attiva l'interfaccia di rete, invia in modalità broadcast un messaggio **DHCPDISCOVER** in cerca di un server DHCP





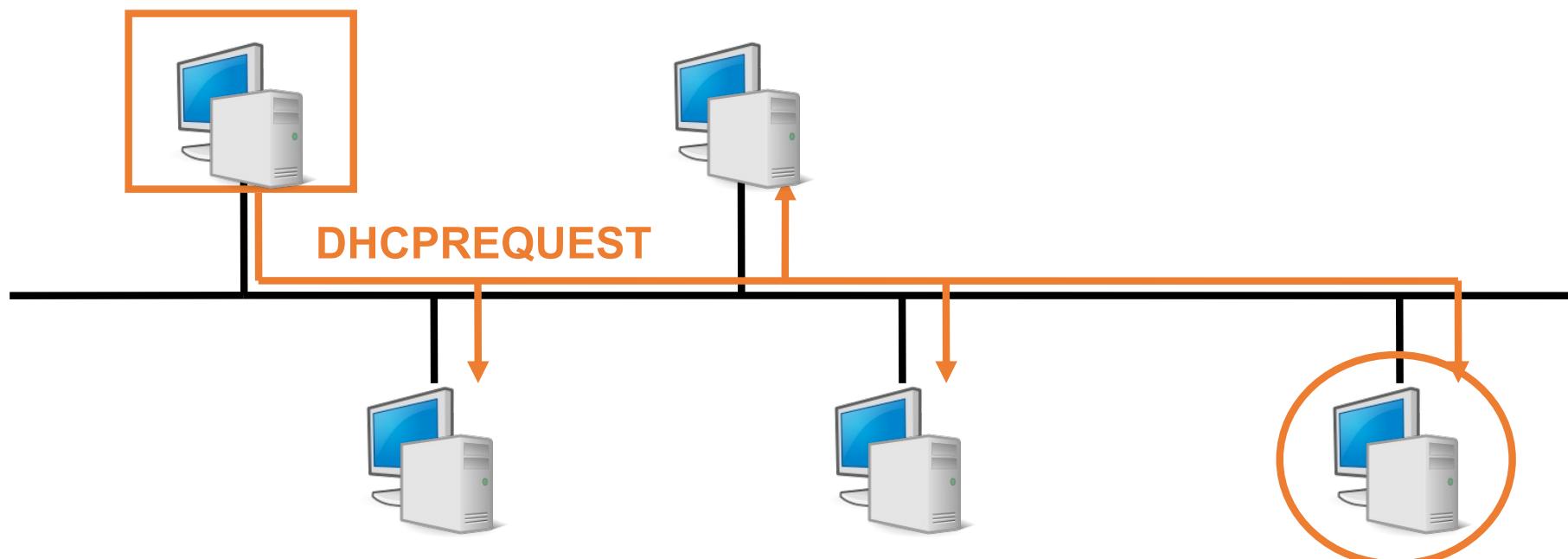
DHCP – 2

- Ciascun server DHCP presente risponde all'host con un messaggio **DHCPOFFER** con cui propone un indirizzo IP



DHCP – 3

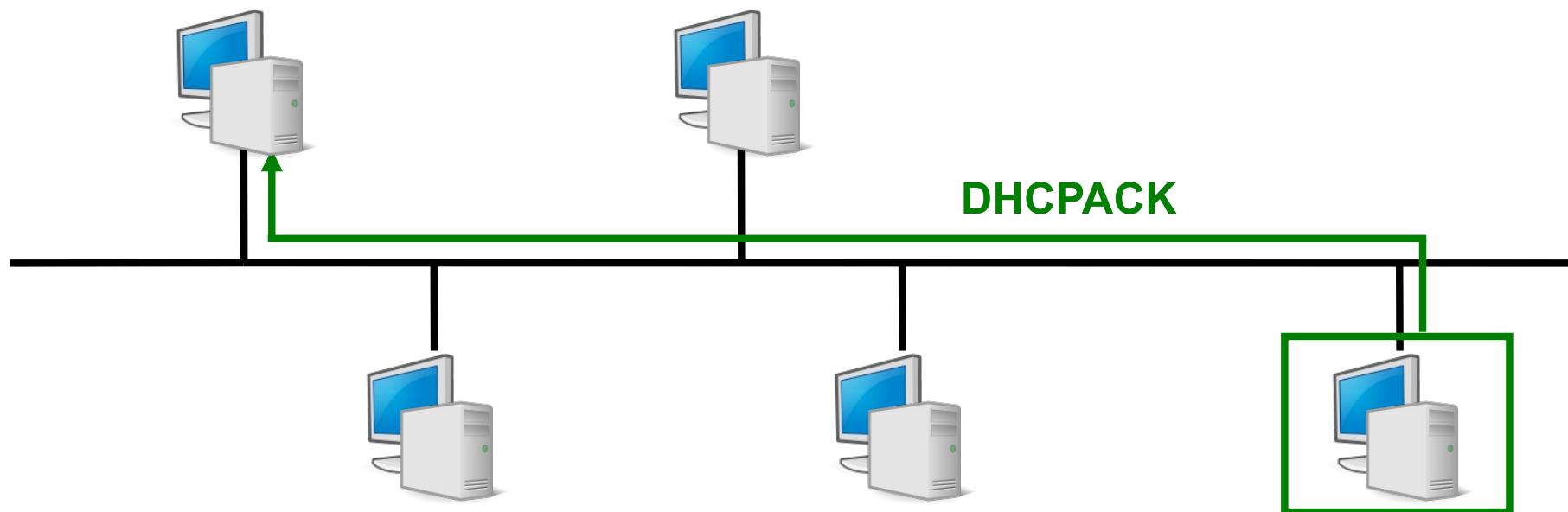
- L' host accetta una delle offerte proposte dai server e manda un messaggio **DHCPREQUEST** in cui richiede la configurazione, specificando il server





DHCP – 4

- Il server DHCP risponde all'host con un messaggio **DHCPACK** specificando i parametri di configurazione





Ulteriori dettagli

- Un'analisi dettagliata del protocollo DHCP che include:
 - Esempi operativi
 - Catture di traffico
- Si può trovare su [virtuale](#)