Reti di Calcolatori

Prof. Roberto Canonico

Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione

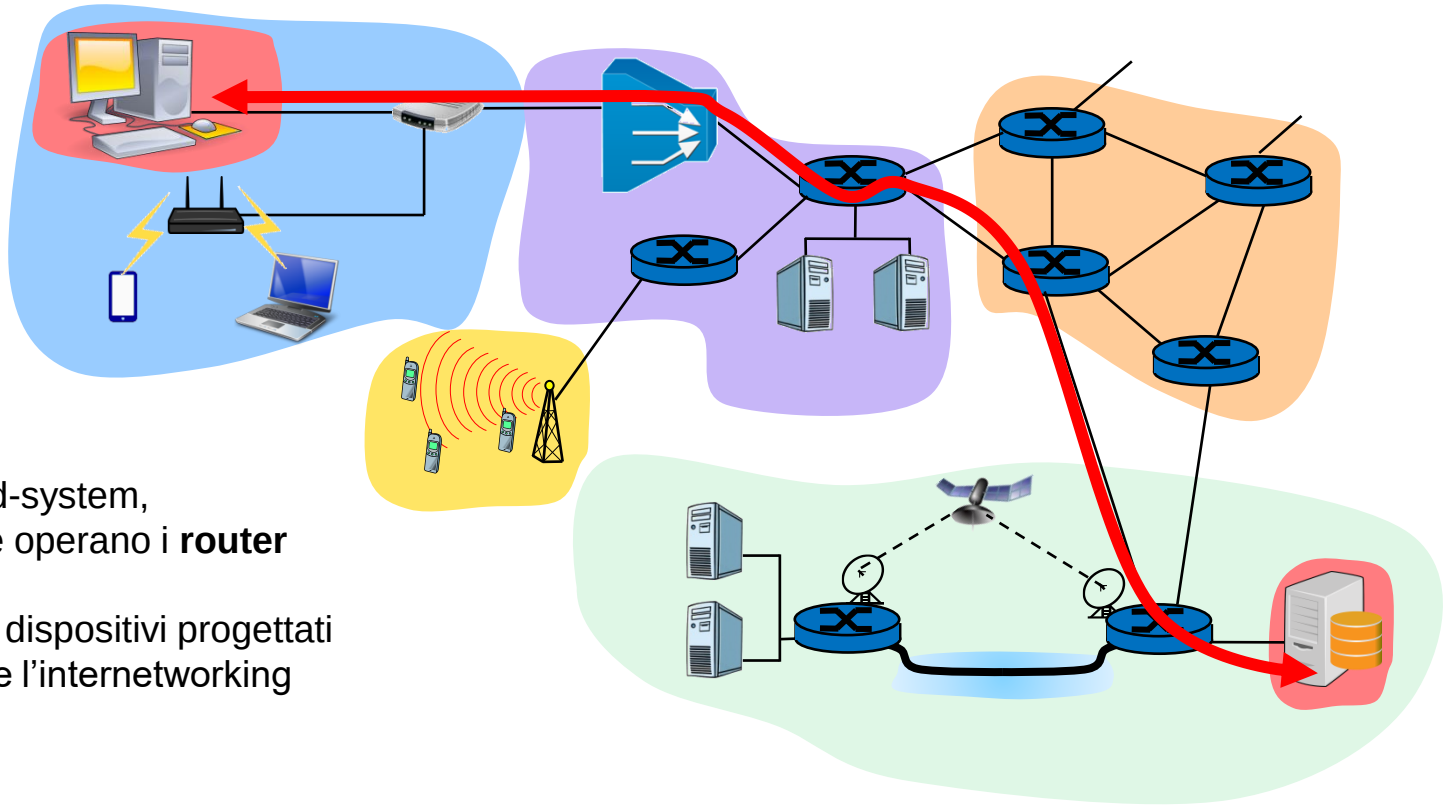
Corso di Laurea in Ingegneria Informatica

Il protocollo di rete IPv4

Formato dei pacchetti ed indirizzamento

Il compito del Livello Rete (layer 3)

In una rete di computer ottenuta attraverso la interconnessione di reti distinte (*internetwork*), il compito del ***livello rete*** è quello di definire i percorsi dei pacchetti nel loro transito da host mittente a host destinazione



Oltre agli end-system,
al livello rete operano i **router**

I router sono dispositivi progettati per realizzare l'internetworking

Reti di calcolatori e packet switching

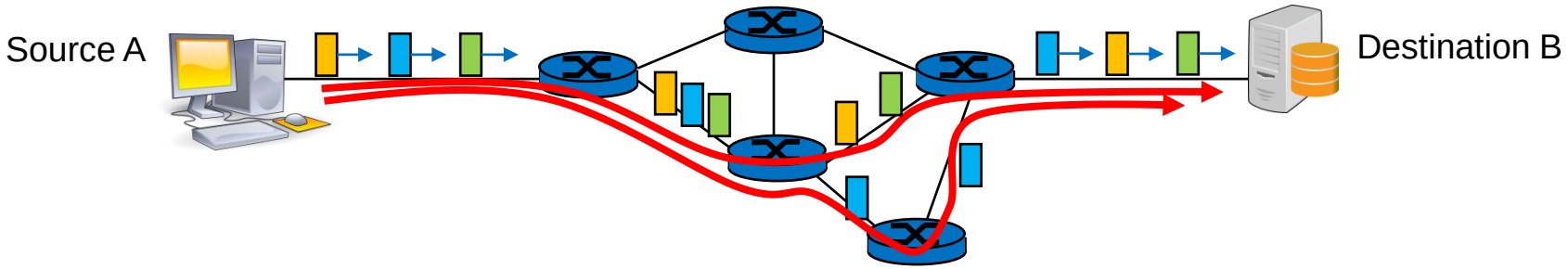
- Le reti di calcolatori operano secondo il modello detto ***packet switching o commutazione di pacchetto***
- In una rete a commutazione di pacchetto l'informazione è trasmessa in ***pacchetti*** formati da una intestazione (***header***) ed un ***payload***
  - l'header contiene informazioni di controllo, tra le quali un indirizzo destinazione che serve ad identificare il terminale a cui il pacchetto deve essere consegnato

  - In uno stack protocollare, ciascun layer aggiunge un suo proprio header (L2 anche un trailer); la struttura risultante è la seguente:

- I dispositivi intermedi che operano al livello rete funzionano in una modalità detta ***store-and-forward***
  - ogni pacchetto è ricevuto interamente, se ne controlla l'assenza di errori e se ne opera la ritrasmissione su un link di uscita
  - all'interno dei dispositivi intermedi, i pacchetti sono mantenuti in buffer di memoria gestiti come delle code

Packet switching: modello a datagram

- In una rete a commutazione di pacchetto basata sul ***modello a datagram***, ciascun pacchetto è inoltrato verso la sua destinazione indipendentemente dagli altri
  - Ogni volta che un pacchetto arriva ad un dispositivo intermedio che opera al livello rete (cioè un ***router***), il dispositivo inoltra il pacchetto verso un successivo dispositivo intermedio (o verso il destinatario finale del pacchetto, qualora esso sia direttamente raggiungibile)
  - Pacchetti inviati da un terminale A verso un terminale B in momenti successivi possono seguire percorsi differenti nella rete e, quindi, arrivare a destinazione in ordine diverso da quello con il quale sono stato trasmessi



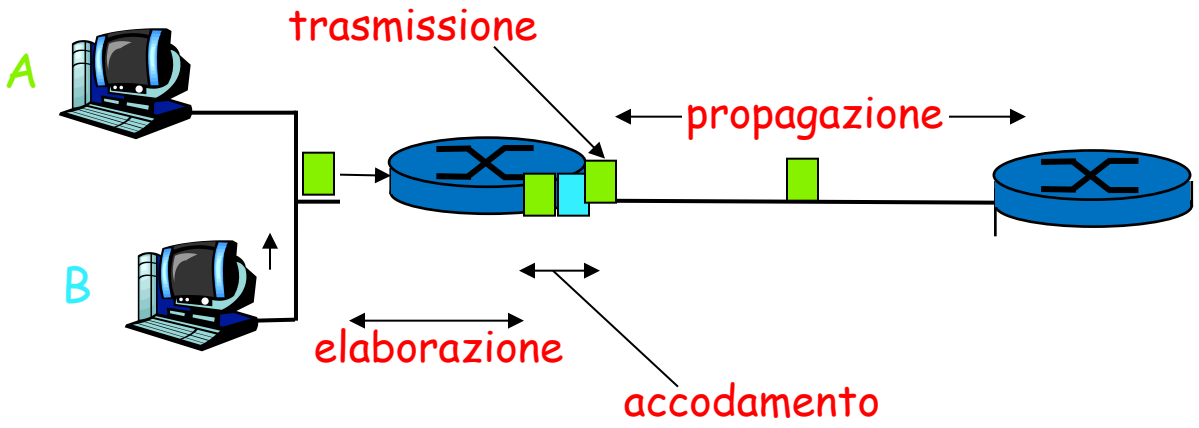
E' possibile che dei pacchetti non arrivino a destinazione

Qualità del Servizio

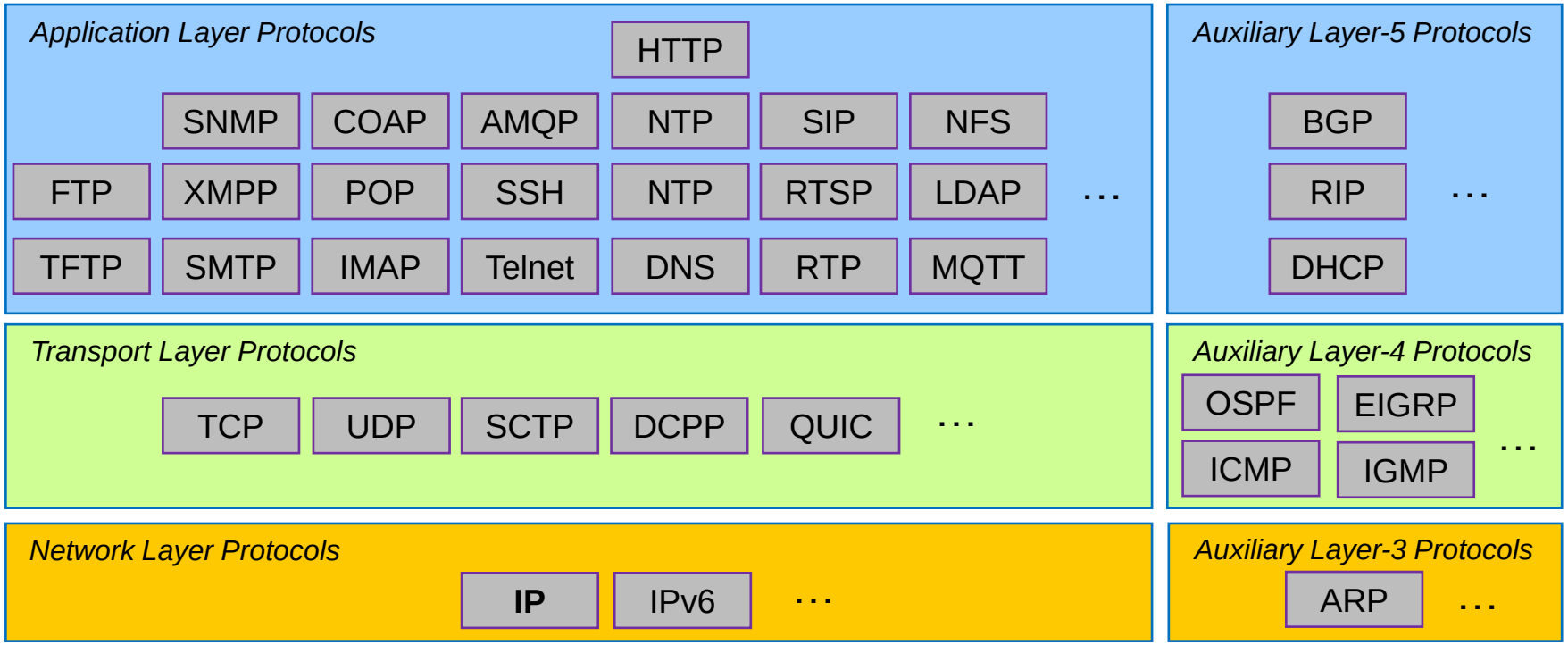
- Il servizio offerto da una rete a commutazione di pacchetto consiste nel recapitare pacchetti da un qualunque terminale mittente ad un qualunque terminale destinatario
- La **Qualità del Servizio (QoS)** di una rete a commutazione di pacchetto è misurata da una molteplicità di "indici di prestazione"
- Relativamente alla comunicazione tra due terminali collegati ad una rete, i parametri di QoS più comunemente utilizzati sono:
  - ***End-to-end delay***: ritardo nella consegna dei pacchetti [s]
  - ***Packet delay variation (PDV):*** variazione temporale del ritardo one-way (spesso anche indicata con il termine ***packet jitter)***
  - ***Throughput***: quantità di bit al secondo che la rete è in grado di trasferire tra i due terminali [b/s]
  - ***Loss-Rate***: probabilità che un pacchetto non venga consegnato a destinazione

Ritardo nelle reti a commutazione di pacchetto

- Il ritardo nella consegna di un pacchetto alla destinazione è determinato da:
  - Tempo di elaborazione nel nodo:
    - controllo di errori, determinazione link di uscita, …
  - Tempo di trasmissione su ciascun link = Lunghezza in bit / velocità in bps
  - Tempo di attesa nelle code dei router (variabile)
  - Tempo di propagazione sulle linee = lunghezza della linea / velocità del segnale



La Internet Protocol suite ed il protocollo IP



- Nella rete Internet, la funzione principale del livello rete è svolta dal protocollo IP
- La versione ancora oggi prevalentemente utilizzata è la versione 4 del protocollo IP
  - IP versione 6 è progressivamente introdotto ed utilizzato
- La caratteristica principale del protocollo IP è quella di offrire un servizio di consegna elementare e senza garanzie (*best effort*) di pacchetti
  - La semplicità rende IP adattabile ad un'ampia varietà di tecnologie di livello inferiore

Chi definisce come funziona Internet: l'IETF

- La rete Internet è una "rete di reti" basata su standard aperti
- I protocolli di comunicazione utilizzati nei livelli Rete, Trasporto ed Applicazione in Internet sono definiti da una comunità aperta di esperti detta

Internet Engineering Task Force (IETF)

- L'IETF è organizzata in gruppi di lavoro (*working groups*) che operano soprattutto tramite mailing list, aperte alla partecipazione di chiunque sia interessato
- Tre volte l'anno l'IETF organizza dei meeting plenari
  - IETF 101 a Londra – Marzo 2018
- I gruppi di lavoro si occupano ciascuno di uno specifico argomento e sono organizzati in aree (protocolli applicativi, sicurezza, ecc...)
- Ogni gruppo produce dei documenti detti RFC (*Request For Comments*) che vengono sottoposti alla IESG (*Internet Engineering Steering Group*) per il loro avanzamento a standard ufficiale
  - Prima di arrivare allo stato di RFC i documenti condivisi nei working group sono denominati *Internet Draft* (I-D)

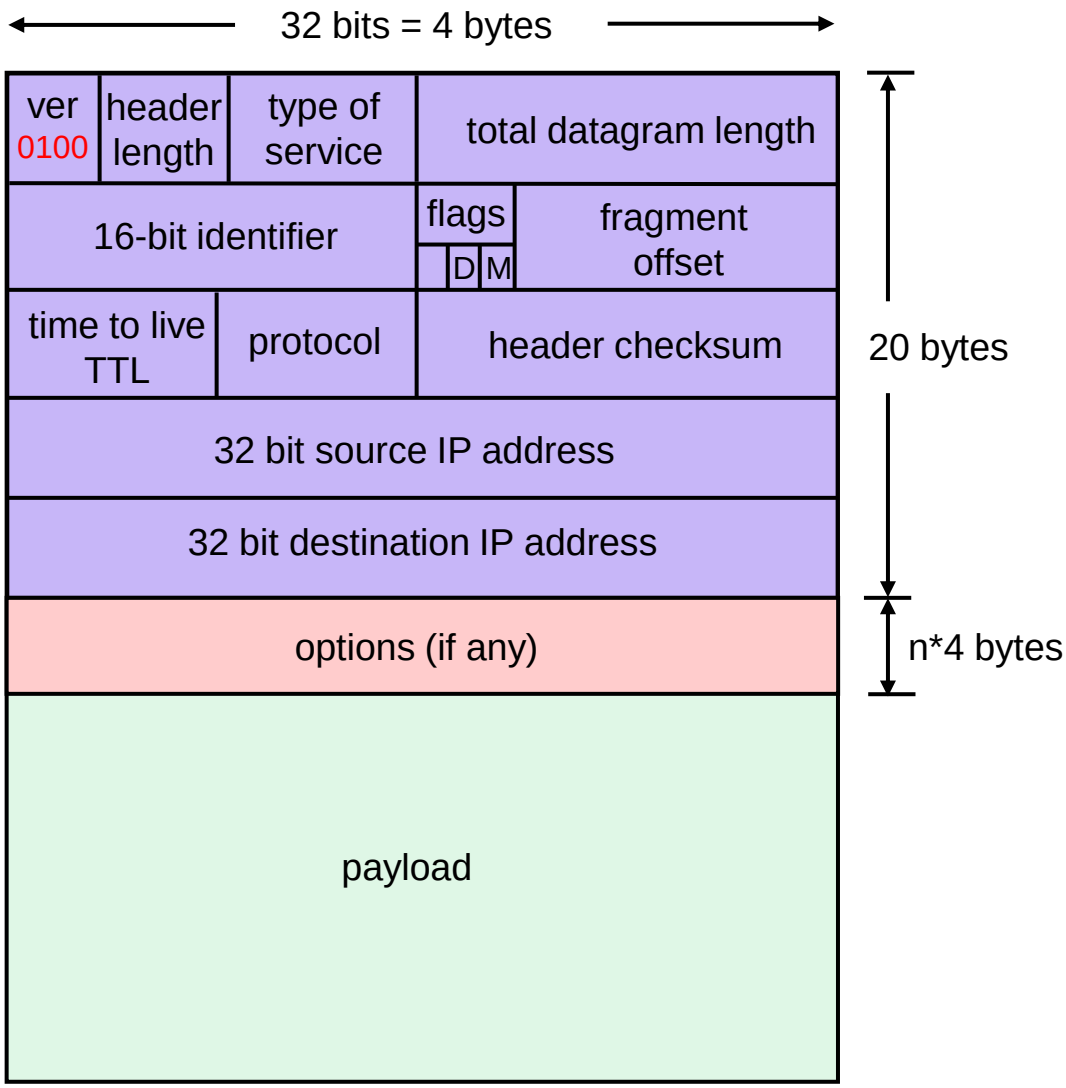
Il protocollo IP versione 4

- Nella rete Internet, la funzione principale del livello rete è svolta dal protocollo IP
  - IPv4 definito in RFC 791 (settembre 1981)
- IP realizza un servizio di consegna best-effort di pacchetti singoli (*datagram*)
- Al di sopra di IP, nello stack TCP/IP (*Internet Protocol Suite*), operano i protocolli di livello trasporto (UDP e TCP)
- Il protocollo IP gestisce indirizzamento, frammentazione, ri-assemblaggio e multiplexing dei protocolli
- E' implementato sia negli end-system (terminali) che nei router
- È responsabile dell'***instradamento*** dei pacchetti, cioè della scelta dell'interfaccia sulla quale un pacchetto deve essere trasmesso per arrivare a destinazione
- Un datagramma IPv4 può avere una dimensione massima di 65535 byte ($2^{16} – 1$) ed è costituito da un header ed un payload
- In IPv4 l'***header*** è costituito da una parte a struttura fissa (20 byte) ed una opzionale
- Il ***payload*** è creato di norma da un protocollo di trasporto (TCP o UDP)
  - In circostanze particolari, il payload di un pacchetto IP può contenere un altro pacchetto IP: *incapsulamento IP in IP*
  - Alcuni protocolli ausiliari (cioè non intesi a supportare la comunicazione di applicazioni eseguite nei terminali) inviano i loro messaggi inserendoli direttamente in un payload IP: ICMP, IGMP, OSPF

IP: servizio best effort

- IP non garantisce di prevenire:
  - pacchetti duplicati
  - consegna ritardata o fuori ordine
  - corruzione di dati
  - perdita di pacchetti
- La consegna affidabile dei messaggi alle applicazioni può avvenire grazie a meccanismi di controllo realizzati nei protocolli di livello superiore (negli end-system)
- Ogni router che riceve un pacchetto IP decide a quale altro nodo inoltrarlo, sulla base dell'indirizzo destinazione contenuto nel pacchetto, in maniera indipendente …
  - rispetto agli altri router
  - rispetto agli altri pacchetti passati in precedenza per lo stesso router
- Il protocollo IP è stato progettato per realizzare un servizio *best-effort*
- Servizio best-effort significa che la rete
  - non fornisce alcuna garanzia sulla consegna di un pacchetto
  - ma non discrimina un pacchetto rispetto ad altri
    - ***network neutrality***

Struttura di un datagram IP versione 4



Campi dell'header IP versione 4

- In IPv4 l'***header*** è costituito da una parte a struttura fissa (20 byte) ed una opzionale di lunghezza multipla di 4 byte
- ***IP header length (4 bit):*** lunghezza dell'header, in multipli di 32 bit (max 60 byte)
- ***Type-of-Service (8 bit):*** specifica il tipo di servizio che si richiede alla rete
  - usato, in pratica, per scopi differenti
- ***Total length (16 bit):*** indica la lunghezza in byte dell'intero pacchetto (header+dati)
- ***Time-to-live TTL (8 bit):*** numero residuo di router attraversabili
  - viene decrementato di 1 da ogni router, a 0 il pacchetto viene scartato
  - serve, in caso di percorsi circolari (*loop*), ad evitare che un pacchetto resti perennemente in circolo
- ***Protocol (8 bit):*** indica il protocollo di livello superiore associato al payload
  - il valore 6 indica TCP, 17 indica UDP
  - serve al de-multiplexing dei pacchetti a destinazione
- ***Header checksum (16 bit):*** serve a verificare l'integrità dell'header IP
- ***Source IP Address (32 bit):*** indirizzo IP del nodo mittente del pacchetto
- ***Destination IP Address (32 bit):*** indirizzo IP del nodo destinatario del pacchetto
- ***Identification (16 bit), Flags (3 bit), Fragment Offset (13 bit):*** sono usati in caso di frammentazione del pacchetto da parte di un router
  - consentono al nodo destinatario di ricostruire il pacchetto originario

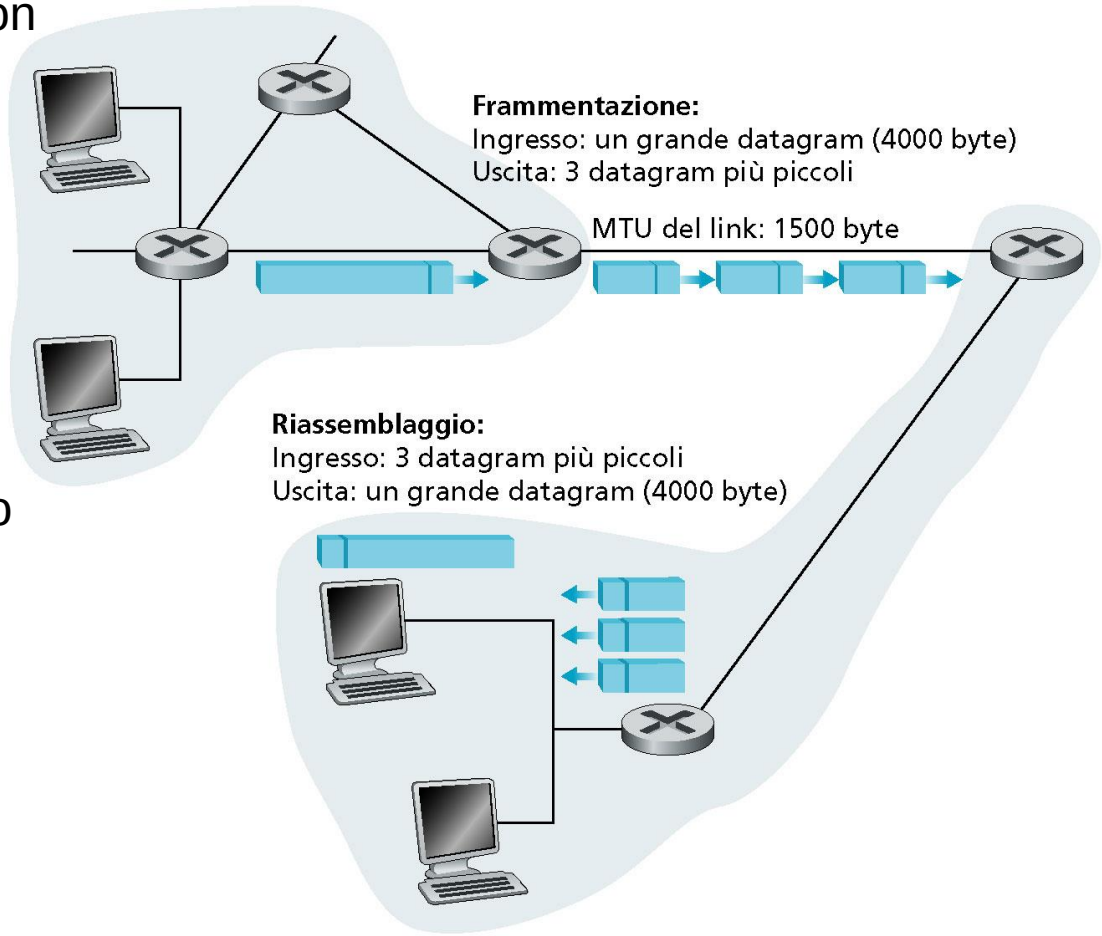
Campi Identification, Flags e Fragment offset

- Questi campi servono a gestire la frammentazione dei pacchetti IPv4
- Un pacchetto IPv4 può essere “spezzato” da un router in una sequenza di pacchetti che singolarmente viaggiano verso il destinatario
- Il livello IP del destinatario finale si occupa del “riassemblaggio” del pacchetto originario prima di consegnarlo allo strato superiore
- Un pacchetto può essere frammentato anche più volte lungo il percorso
- La necessità di frammentare un pacchetto si presenta quando la dimensione del pacchetto supera la Maximum Transmissible Unit (MTU) sul link di uscita
- Il valore di MTU dipende dalla tecnologia usata al livello 2
  - Es. in Ethernet la MTU è 1500 byte

- **Identification**
  - Questo campo (16 bit) è un identificativo del datagramma
  - Serve ad associare diversi frammenti ad un unico pacchetto originario
- **Flags**
  - Il bit D (*don’t fragment*) indica se il pacchetto può essere frammentato
  - Il bit M (*more fragments*) indica se il pacchetto è l’ultimo frammento
- **Fragment offset**
  - 13 bit, identifica la posizione del frammento all’interno del pacchetto

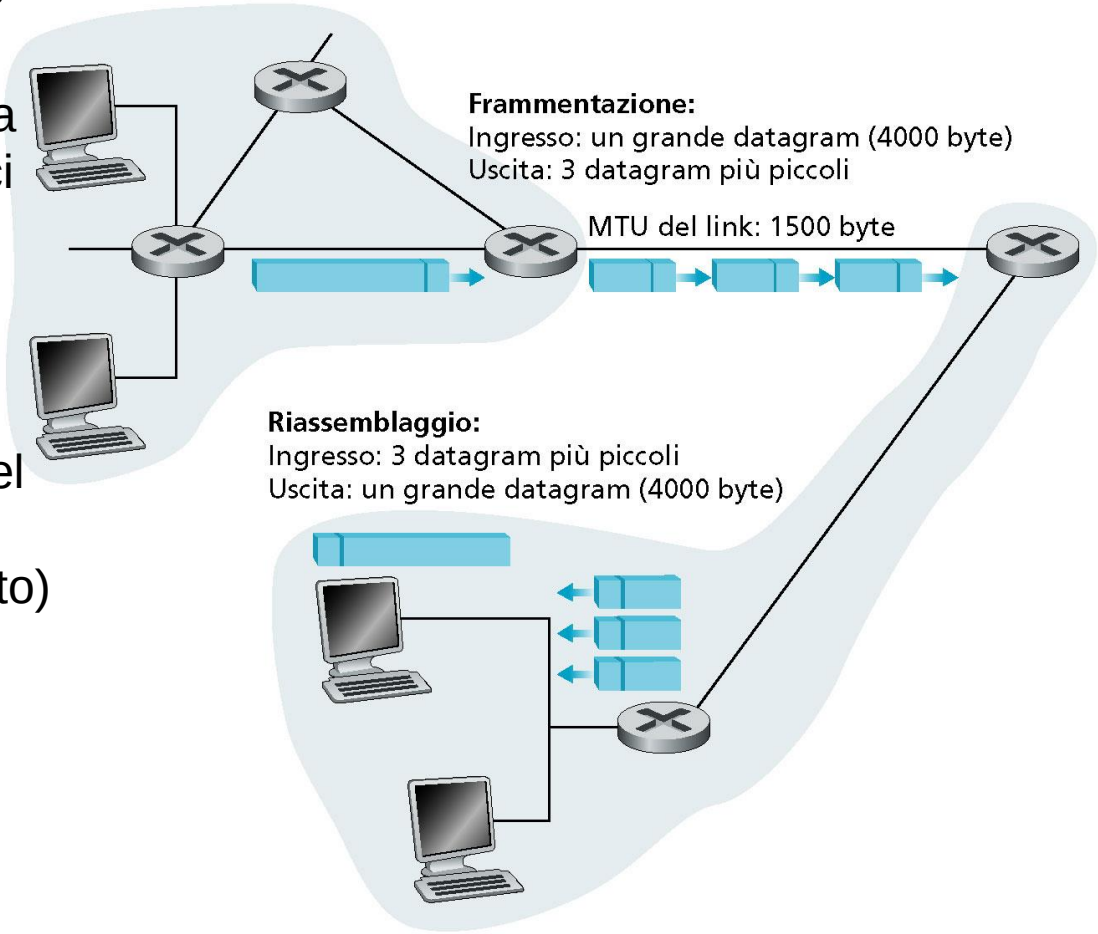
Frammentazione e riassemblaggio IP

- Se un pacchetto di dimensione N arriva ad un router e deve essere trasmesso su un link di uscita con MTU M<N, il pacchetto è ***frammentato***
- Ogni frammento è trasmesso come singolo pacchetto IP
- La dimensione del payload di ogni frammento è un multiplo di 8 byte
- Tutti i frammenti hanno lo stesso ID number



Frammentazione e riassemblaggio IP (2)

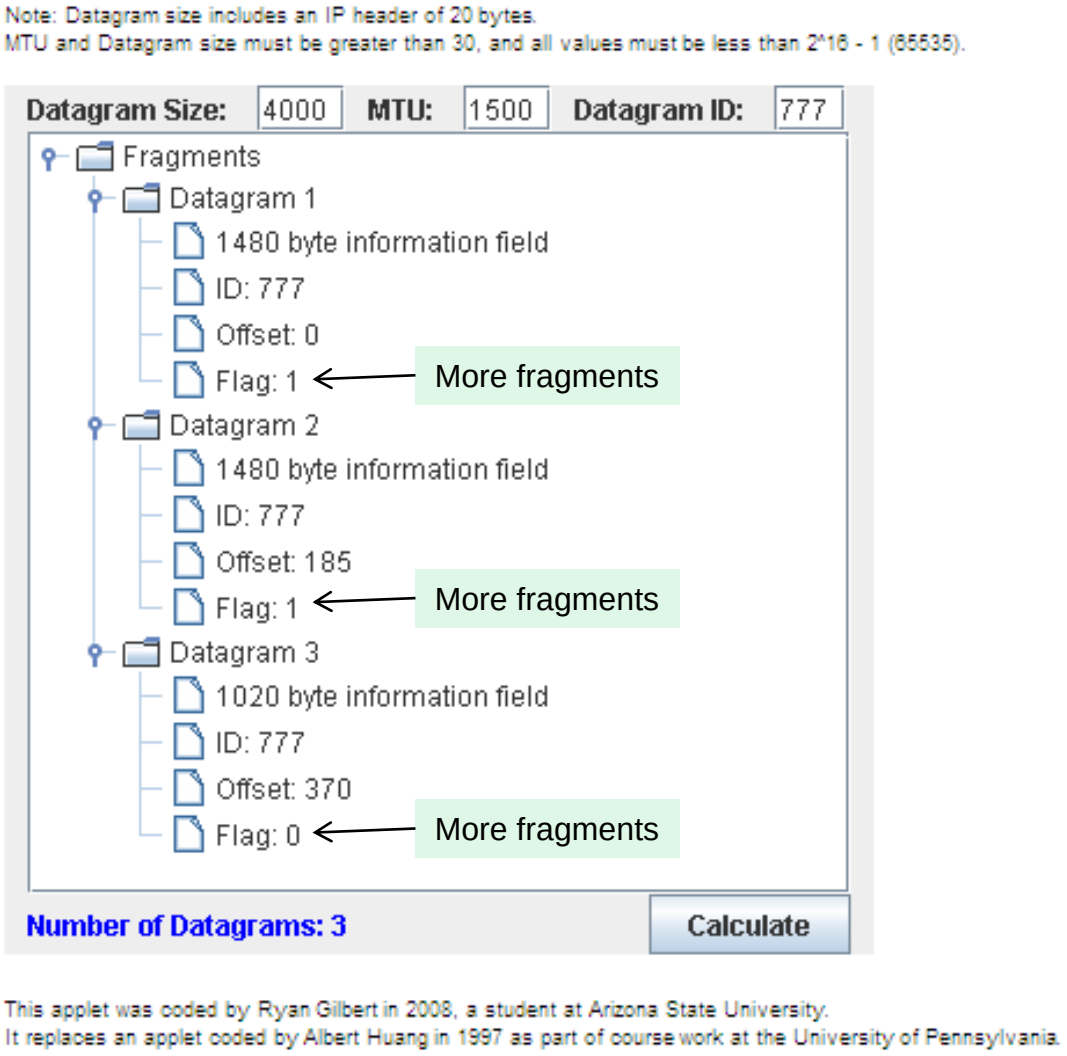
- Tutti i frammenti (tranne l'ultimo) hanno un payload di dimensione multipla di 8 byte
- Essendo la dimensione massima di un datagramma 65535 byte, ci possono essere al massimo 65536/8 cioè 8192 frammenti per ogni datagramma
- La posizione del payload di un frammento rispetto al payload del pacchetto originario è espressa mediante un offset (spiazzamento) di 13 bit



Frammentazione IP: esempio 1

- N = 4000, MTU = 1500
- Tre frammenti, ciascuno con header 20 byte
- Frammento 1:
  - payload 1480
  - offset 0
- Frammento 2:
  - payload 1480
  - offset (1480/8)=185
- Frammento 3:
  - payload 1020
  - offset (1480+1480)/8=370

NOTA:
20+1480+1480+1020=4000



http://media.pearsoncmg.com/aw/aw_kurose_network_2/applets/ip/ipfragmentation.html

Frammentazione IP: esempio 2

- Il pacchetto IP raffigurato di seguito deve attraversare un link avente Maximum Transfer Unit (MTU) pari a 1500 bytes. Come verrà trattato?

Original IP Datagram

Sequence	Identifier	Total Length	DF May / Don't	MF Last / More	Fragment Offset
0	345	5140	0	0	0

IP Fragments (Ethernet)

Sequence	Identifier	Total Length	DF May / Don't	MF Last / More	Fragment Offset
0-0	345	1500	0	1	0
0-1	345	1500	0	1	185
0-2	345	1500	0	1	370
0-3	345	700	0	0	555

Frammentazione: problemi e come evitarli

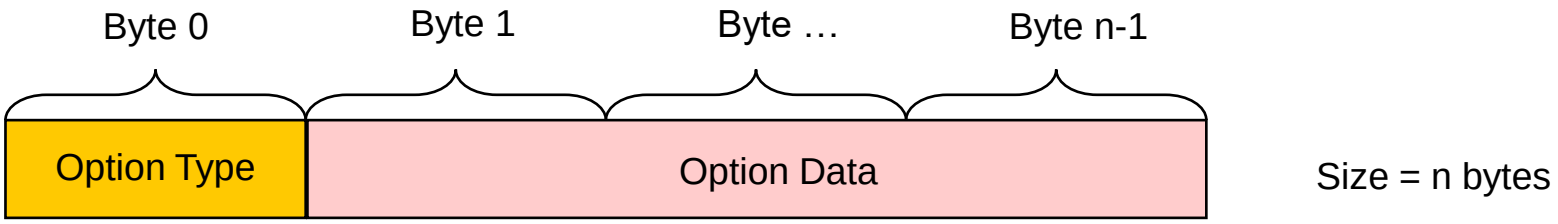
- Il compito di riassemblaggio è oneroso
- Il destinatario deve collezionare tutti i frammenti del pacchetto originario prima di consegnare il payload al livello superiore
- Se non termina entro un determinato tempo, tutti i frammenti arrivati sono scartati
- Può essere una tecnica per attaccare un host bersaglio
- Per evitare la frammentazione dei pacchetti lungo il percorso, talvolta si effettua un *path MTU discovery*, cioè si determina la più piccola MTU lungo il percorso da un host A ad un host B
  - Conoscendo il path MTU, A evita del tutto la frammentazione se invia pacchetti di dimensione minore a tale valore
- Un esempio di path MTU discovery
  - A invia un pacchetto ICMP echo request a B di massima dimensione con flag D=1
  - Se il pacchetto incontra sul percorso un router che non riesce a trasmettere il pacchetto, A riceve un messaggio ICMP "Destination unreachable: Fragmentation needed"
  - A dimezza la dimensione e ritrasmette, se riceve da B l'echo reply incrementa la dimensione di un quarto, altrimenti dimezza
  - Ecc…

Opzioni dell'header IPv4

- L'header IP può essere esteso con dei campi "Opzione" mediante le quali si intende chiedere una elaborazione "speciale" del pacchetto da parte dei router
  - Security
  - Source routing
  - Route recording
  - Stream identification
  - Timestamping
- Per la presenza delle opzioni, l'header IP può essere di lunghezza variabile
  - Questo è il motivo della presenza del campo Header Length
  - Se l'opzione non occupa 4 byte (o un suo multiplo), vengono inseriti dei bit di riempimento (tutti zero)
  - Nei router in cui il dataplane è implementato in hardware, l'elaborazione di questi campi non è effettuata in hardware (*fast path*) ma in software (*slow path*), oppure questi campi sono ignorati
    - Gli attacchi DoS di tipo "Christmas Tree" consistono nel trasmettere pacchetti IP con diverse opzioni (inutili) nell'header al fine di sovraccaricare i router

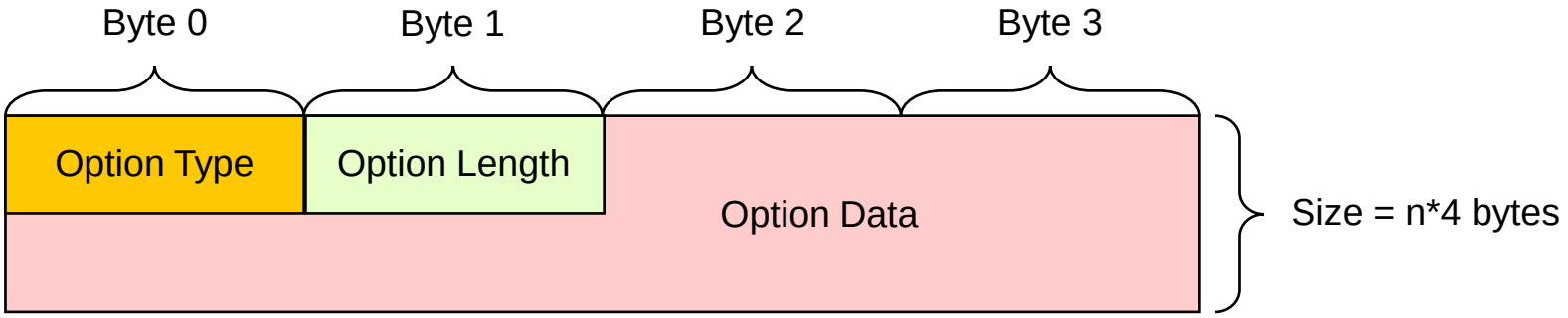
IPv4: header option format - Case 1 (single byte)

- Le opzioni di questo formato hanno una lunghezza di n byte implicitamente definita dal valore di Option Type
- Esempi: "End of Option List" (Type = 0) e "No Operation" (Type = 1)



IPv4: header option format - Case 2

- Le opzioni di questo formato hanno una lunghezza multipla di 4 byte esplicitamente indicata nel campo Option Length
- I numeri di IP Option Type standard sono registrati in una lista gestita da IANA
  - "IP OPTION NUMBERS": https://www.iana.org/assignments/ip-parameters/ip-parameters.xhtml

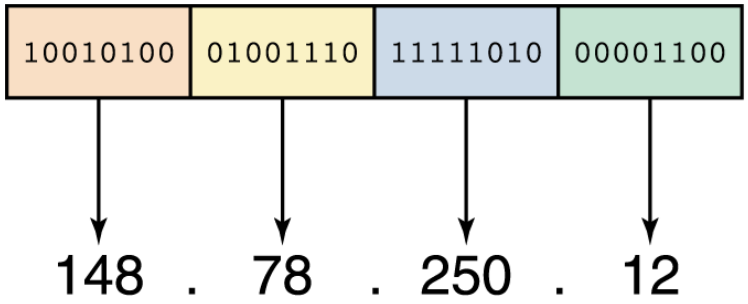


Option Type byte

Subfield Name	Size (bits)	Description
Copied	1	If 1: Option to be copied in all fragments If 0: Option only kept in first fragment
Option Class	2	0: Control Options 1: Unused 2: Debugging/Measurement 3: Unused
Option Number	5	Up to 32 different Options for each class

Indirizzi IP

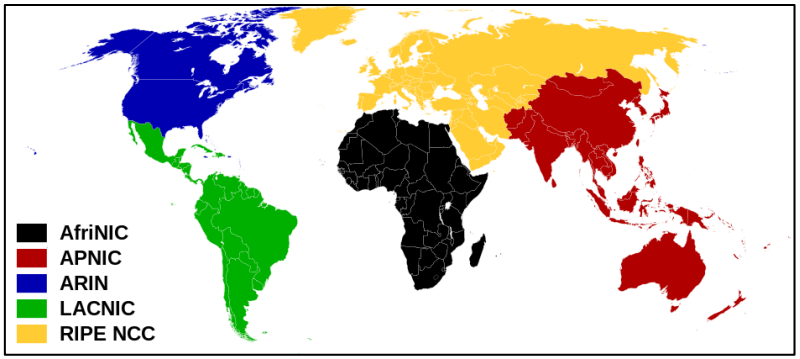
- Un indirizzo IP è una sequenza di 32 bit
- Un pacchetto IP ha, nell'header, l'indirizzo IP del mittente e quello del destinatario
- In forma testuale, per un uso da parte di un utente umano, un indirizzo IP è solitamente rappresentato nella ***notazione dotted decimal***:
  - i 32 bit sono decomposti in 4 byte, il valore di ciascuno dei quali è riportato in decimale come numero naturale tra 0 e 255
  - i quattro numeri decimali sono scritti in sequenza separati dal punto



- In una rete IP (ad esempio, la rete Internet) un indirizzo IP serve ad identificare univocamente un'interfaccia di rete di un dispositivo
  - Un end system può avere una sola interfaccia di rete, un router almeno due
  - I terminali moderni hanno diverse interfacce di rete (***multi-homed***) e dunque diversi indirizzi IP (es. interfaccia Ethernet, WiFi, Bluetooth, ecc.)

Chi assegna gli indirizzi IP

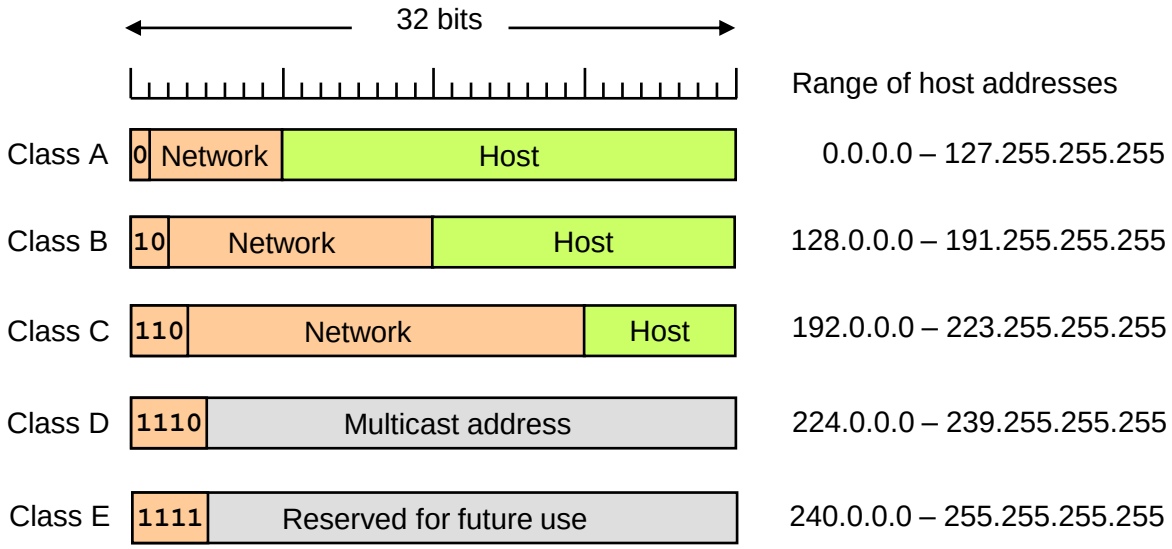
- L'assegnazione degli indirizzi IP avviene attraverso un sistema gerarchico di autorità
- Il gestore globale dell'intero spazio di indirizzamento è IANA
  - IANA - Internet Assigned Numbers Authority
  - In origine IANA era una persona: Jon Postel
- IANA dipartimento di ICANN (*Internet Corporation for Assigned Names and Numbers*)
- IANA delega la gestione degli indirizzi IP a cinque autorità regionali (RIR)
  - In Europa opera come *Regional Internet Registry* il RIPE NCC



- I registry regionali assegnano blocchi di indirizzi agli Internet Service Provider (ISP) ed alle grosse organizzazioni
- Questi, a loro volta, sono responsabili della assegnazione unica degli indirizzi di loro pertinenza ai singoli dispositivi delle proprie reti

Indirizzi IP e classi (1)

- Un indirizzo IP è costituito da due parti: un identificatore ***Network*** della rete di appartenenza e un identificatore ***Host*** che identifica il terminale all'interno della rete
- Nella rete Internet, inizialmente, si adottò una ***gestione degli indirizzi per classi***
- Nella gestione per classi, la demarcazione tra i campi Network ed Host è fissa e determinata dal valore dei primi bit
- ***Significato di rete:*** tutti gli host di una stessa rete possono comunicare direttamente a livello 2, senza l'ausilio di un router

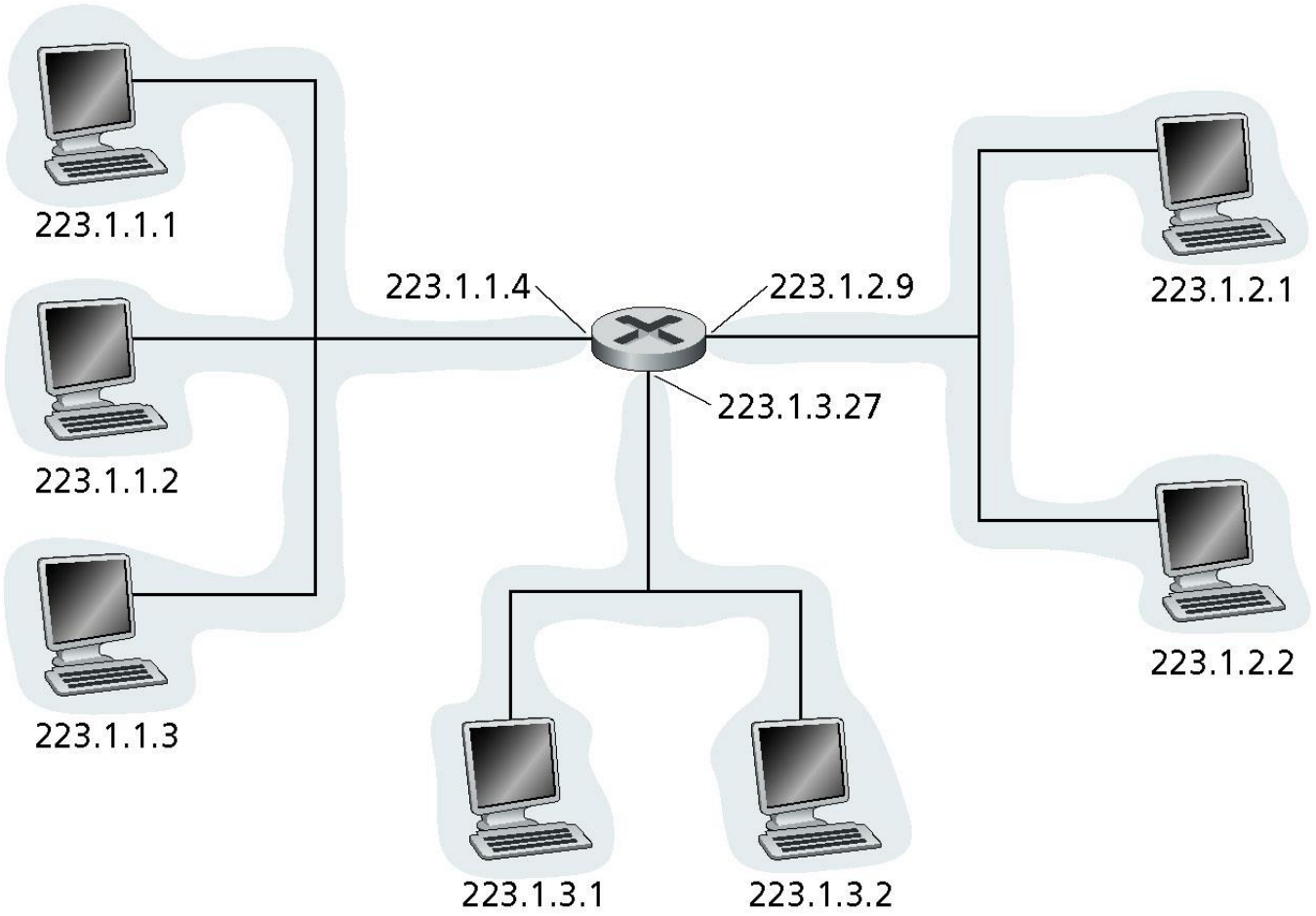


Indirizzi IP e classi (2)

- Un indirizzo IP di **classe A** usa il primo byte per identificare la rete ed i restanti tre byte per identificare l'host
  - Una rete di classe A è un blocco di 2^{24} = 16.777.216 indirizzi consecutivi
  - Esistono 256 reti di classe A distinte
- Un indirizzo IP di **classe B** usa i primi due byte per identificare la rete ed i restanti due byte per identificare l'host
  - Una rete di classe B è un blocco di 2^{16} = 65.536 indirizzi consecutivi
  - Esistono 2^{16} = 65.536 reti di classe B distinte
- Un indirizzo IP di **classe C** usa i primi tre byte per identificare la rete ed il restante byte per identificare l'host
  - Una rete di classe C è un blocco di 2^{8} = 256 indirizzi consecutivi
  - Esistono 2^{24} = 16.777.216 reti di classe C distinte
- Gli indirizzi di **classe D** (nel range 224.0.0.0-239.255.255.255) sono usati per identificare gruppi di trasmissione multicast (RFC1112)
  - Possono essere usati solo come indirizzo destinazione
- Gli indirizzi di **classe E** (nel range 240.0.0.0-255.255.255.255) sono stati riservati per usi futuri e mai utilizzati

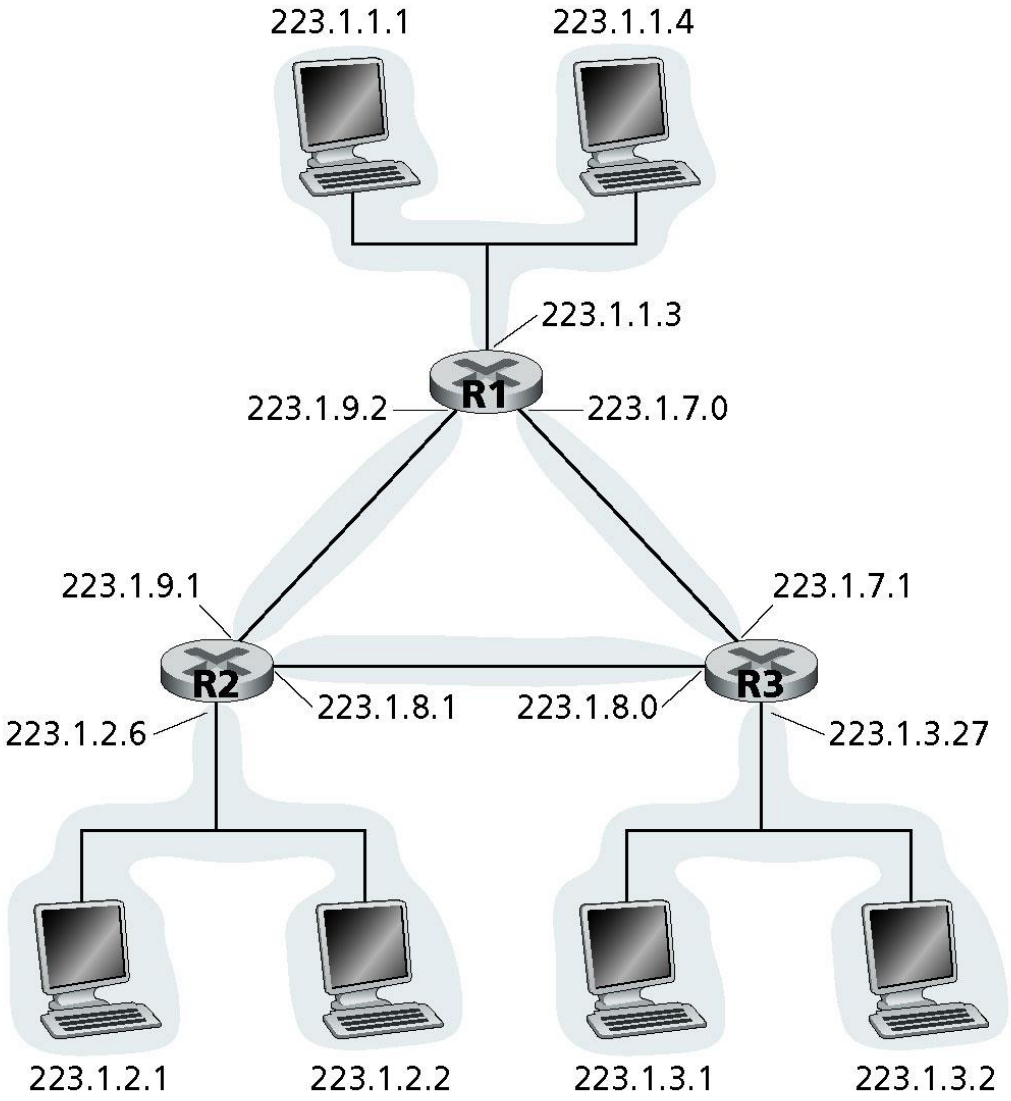
Assegnazione indirizzi IP alle interfacce di rete (1)

- Scenario con tre reti fisiche associate a tre diverse reti di classe C
  - 223.1.1.X è il prefisso per la rete a sinistra
  - 223.1.2.X è il prefisso per la rete a destra
  - 223.1.3.X è il prefisso per la rete in basso



Assegnazione indirizzi IP alle interfacce di rete (2)

- Scenario con sei distinte reti fisiche associate a sei diverse reti di classe C



Indirizzi IP speciali

- L'indirizzo **0.0.0.0** è usato per scopi speciali in vari contesti
  - Ad esempio, all'interno di un host, identifica "qualunque indirizzo IP assegnato alle sue proprie interfacce"
- Tutto il blocco di indirizzi **0.X.Y.Z** (con X, Y, e Z qualsiasi) è riservato e non può essere assegnato specificamente ad un'interfaccia
- Gli indirizzi della rete **127.0.0.0**, cioè del tipo **127.X.Y.Z** (con X, Y, e Z qualsiasi), sono tutti associati ad un'interfaccia virtuale che è presente in qualunque sistema e che può essere usata per la comunicazione tra processi in esecuzione nella stessa macchina (***interfaccia di loopback***)
  - L'interfaccia di loopback è, di solito, configurata con l'indirizzo **127.0.0.1**
- L'indirizzo **255.255.255.255** (usato come destinazione) indica il ***broadcast*** a tutti gli host nella rete locale del mittente
- L'indirizzo che ha tutti zero nel campo host serve ad identificare la rete
  - Es. la rete 148.78.0.0
- L'indirizzo che ha tutti uno nel campo host serve ad identificare (come destinatario) tutti gli host della rete specificata nel campo network (***broadcast diretto***)
  - Es. un pacchetto con indirizzo 148.78.255.255 è consegnato a tutti i sistemi che hanno un'interfaccia nella rete 148.78.0.0
- Data una rete qualsiasi, gli indirizzi che hanno nel campo host tutti zero e tutti uno sono considerati speciali e quindi non assegnabili a specifici host

L'interfaccia di loopback negli end-system

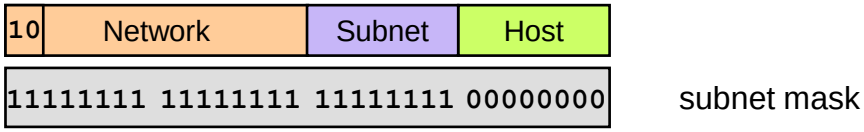
- Tipicamente un qualunque end-system è configurato in modo da avere almeno un'interfaccia di rete virtuale (cioè non associata ad una scheda di rete fisica) detta **interfaccia di loopback**
- Lo scopo di un'interfaccia di loopback è quello di consentire la comunicazione tra processi attivi nello stesso end-system mediante i protocolli TCP/IP anche quando il sistema è fisicamente disconnesso da una rete
- Nei sistemi Linux:
  - questa interfaccia è denominata **`lo0`** e ad essa è assegnato staticamente l'indirizzo 127.0.0.1
  - A questo indirizzo IP è associato, nel file di sistema **`/etc/hosts`**, il nome **localhost**
  - è possibile creare ulteriori interfacce di loopback (**`lo1`**, **`lo2`**, …) a cui sono assegnati indirizzi del tipo 127.0.0.n
- I dispositivi di rete scartano un qualunque pacchetto che abbia un indirizzo 127.X.Y.Z come indirizzo mittente o destinatario

Indirizzi IP: netmask

- La gestione per classi degli indirizzi IP condusse ad un uso inefficiente dello spazio di indirizzamento e ad una conseguente difficoltà ad assegnare indirizzi IP a nuove reti collegate ad Internet
  - Nella gestione per classi, una rete con più di 256 host necessita di un blocco di indirizzi di classe B che, però, comprende 65.536 indirizzi
- Nel 1992 una nuova tecnica di gestione degli indirizzi IP fu introdotta: CIDR
- In CIDR, la separazione tra campo network e campo host all'interno di una rete è fatta attraverso una stringa di 32 bit ausiliaria, detta ***network mask*** o ***netmask***
- La netmask contiene una sequenza di *k* '1' in testa che identificano la parte di bit che costituiscono l'identificatore di rete, ed una restante sequenza di (32-*k*) '0' che identificano l'host nella rete
- Una netmask si rappresenta o in notazione dotted decimal,
  oppure con la notazione */k*, dove *k* è il numero di '1' consecutivi in testa
- Esempi:
  - 255.255.0.0 o /16
  - 255.255.128.0 o /17
  - 255.255.255.0 o /24
  - 255.255.255.240 o /28
  - 255.255.255.252 o /30

Indirizzi IP: gestione *classless* (CIDR)

- Nella gestione CIDR ciascuna delle reti originariamente definite dalle classi è stata suddivisa in ***sottoreti***, ovvero in blocchi di indirizzi consecutivi
- Una sottorete è identificata usando un campo ***subnet*** sottratto al campo host
- La demarcazione tra i campi subnet e host è realizzata mediante la netmask
- Gli host di una stessa sottorete comunicano direttamente a livello 2 senza l'ausilio di un router
- Tutti gli host della stessa sottorete devono essere configurati con la stessa netmask
- La figura seguente mostra blocco di indirizzi di classe B ripartito in $2^8 = 256$ sottoreti da 256 indirizzi ciascuna



- La subnet dell'esempio può contenere fino a 254 host distinti, perché gli indirizzi che hanno tutti zero e tutti uno nel campo host sono usati per scopi speciali

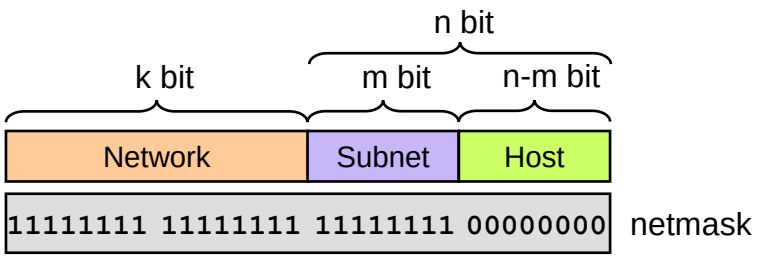
Blocchi di indirizzi IP per usi speciali

- IANA (RFC 1918) ha riservato i seguenti tre blocchi di indirizzi per reti TCP/IP private
  - 10.0.0.0 - 10.255.255.255 (**10.0.0.0/8**) un blocco di 2^{24} indirizzi
  - 172.16.0.0 - 172.31.255.255 (**172.16.0.0/12**) un blocco di 2^{20} indirizzi
  - 192.168.0.0 - 192.168.255.255 (**192.168.0.0/16**) un blocco di 2^{16} indirizzi
- Una ***rete privata*** è una rete non collegata a livello 3 alla rete Internet
- Un'organizzazione può assegnare nella propria rete interna gli indirizzi specificati in RFC 1918 senza dover riceve alcuna autorizzazione
  - Questo però impedisce la possibilità di comunicare con host in Internet
  - A meno di non usare una soluzione di *address translation* (NAT) che vedremo in seguito
- In RFC 5737 sono indicati tre blocchi di indirizzi che sono considerati riservati per l'uso in manuali e documentazione
  - **192.0.2.0/24** (TEST-NET-1)
  - **198.51.100.0/24** (TEST-NET-2)
  - **203.0.113.0/24** (TEST-NET-3)
- I router di Internet sono configurati per eliminare (cioè non inoltrare) pacchetti aventi come indirizzo mittente o destinazione uno degli indirizzi riservati di RFC 1918 ed RFC 5737
- Altri indirizzi IPv4 riservati per usi speciali sono elencati in:

  https://www.iana.org/assignments/iana-ipv4-special-registry/iana-ipv4-special-registry.xhtml

Subnetting con Fixed Length Subnet Mask (FLSM)

- Un blocco di $N=2^n$ indirizzi consecutivi è identificato dal prefisso /*k* con **$k = 32 - n$**
- Il termine ***subnetting con fixed length subnet mask (FLSM)*** indica la ripartizione di un blocco di N indirizzi consecutivi in M sottoinsiemi disgiunti ciascuno formato da (N/M) indirizzi consecutivi (***subnet***)
- In ciascun blocco di (N/M) indirizzi, due indirizzi saranno riservati per scopi speciali:
  - l'indirizzo che ha tutti zero nel campo host indica l'intera subnet
  - l'indirizzo che ha tutti uno nel campo host indica il broadcast alla subnet
- Pertanto, solo (N/M)-2 indirizzi saranno attribuibili alle interfacce degli host che appartengono alla subnet (*host range*)
- All'interno del blocco, ciascuna subnet sarà identificata da **$m = \log_2(M)$** bit
- Tutte le interfacce dei dispositivi della rete saranno configurate con una netmask avente:
  - **k + m** bit '1' per identificare globalmente ciascuna subnet
  - **n - m** bit '0' per identificare ciascun host all'interno di una subnet



Esempio di fixed length subnetting (1)

- Si abbia assegnato il blocco di $N=2^8=256$ indirizzi **192.168.20.0/24**
- Lo si voglia ripartire in **M=8** blocchi uguali (subnet) da N/M=32 indirizzi ciascuno
- In ciascuna subnet al più 30 indirizzi sono assegnabili agli host ed ai router perché due indirizzi sono riservati

- Ciascuna subnet è identificata da $m=\log_2 8=3$ bit
- Occorre usare una netmask con
  - 24+3= 27 bit ‘1’
  - 8-3 = 5 bit ‘0’

- Netmask rappresentata in binario:
  11111111.11111111.11111111.11100000
- Netmask rappresentata in notazione dotted decimal:
  255.255.255.224
- Netmask rappresentata come prefisso: **/27**

Esempio di fixed length subnetting (2)

- Blocco di **N=2^8=256** indirizzi **192.168.20.0/24**
  ripartito in **M=8** subnet da **N/M=32** indirizzi ciascuna con netmask **/27**

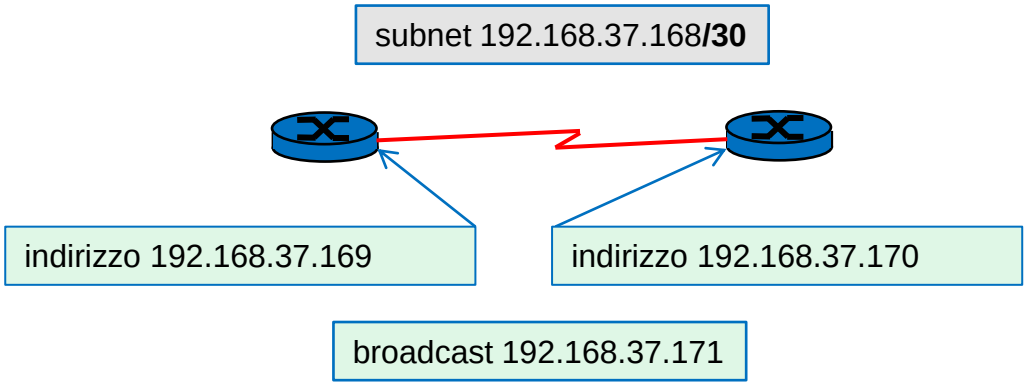
Subnet	Subnet Address	Host Range	Broadcast Address
0	192.168.20.0 /27	192.168.20.1 to 192.168.20.30	192.168.20.31
1	192.168.20.32 /27	192.168.20.33 to 192.168.20.62	192.168.20.63
2	192.168.20.64 /27	192.168.20.65 to 192.168.20.94	192.168.20.95
3	192.168.20.96 /27	192.168.20.97 to 192.168.20.126	192.168.20.127
4	192.168.20.128 /27	192.168.20.129 to 192.168.20.158	192.168.20.159
5	192.168.20.160 /27	192.168.20.161 to 192.168.20.190	192.168.20.191
6	192.168.20.192 /27	192.168.20.193 to 192.168.20.222	192.168.20.223
7	192.168.20.224 /27	192.168.20.225 to 192.168.20.254	192.168.20.255

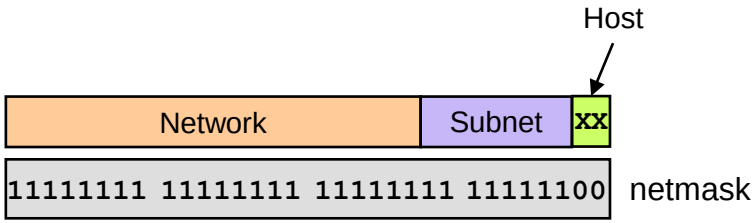
Subnetting VLSM

- Sia disponibile un blocco di $N=2^n$ indirizzi consecutivi identificato dal prefisso ***/k*** con ***k = 32 – n***
- Il termine ***subnetting con variable length subnet mask (VLSM)*** indica la ripartizione del blocco di N indirizzi consecutivi in M sottoinsiemi disgiunti di differente dimensione
  - Tutti i blocchi devono avere come dimensione una potenza di due
- La ripartizione avviene in maniera gerarchica
  - Si ripartisce il blocco in M_1 blocchi "grandi" identificati da un prefisso di **$m_1=\log_2(M_1)$** bit
  - Uno o più dei blocchi ottenuti dalla prima ripartizione sono suddivisi in M_2 blocchi più piccoli identificati da un prefisso di **$m_2=\log_2(M_2)$** bit
  - La ripartizione può essere ulteriormente effettuata in blocchi ancora più piccoli se necessario
- Ciascun blocco sarà associato ad una propria netmask
- Subnet associate ai blocchi "grandi": netmask **$/k+m_1$**
- Subnet associate ai blocchi ottenuti dalla seconda suddivisione: netmask **$/k+m_1+m_2$**
- … e così via

Subnet di dimensione minima /30: point-to-point link

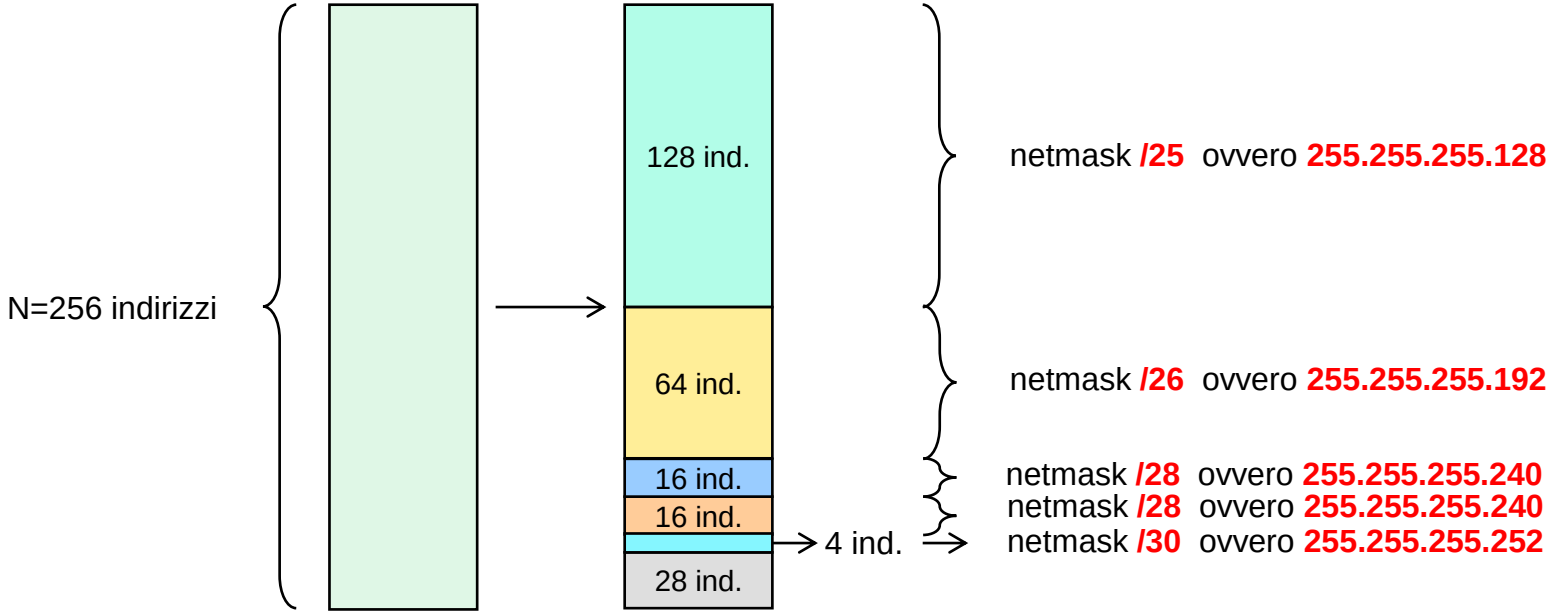
- Quando si usa la tecnica VLSM, alle reti associate ai link punto-punto che collegano due router conviene assegnare una subnet che comporti il minor spreco possibile di indirizzi IP
- Tale subnet deve comprendere quattro indirizzi IP consecutivi
  - indirizzo che ha nel campo host la configurazione di bit **00** riservato per la subnet
  - indirizzo che ha nel campo host la configurazione di bit **11** riservato per il broadcast
- Sono associabili alle interfacce dei due router gli indirizzi che hanno nel campo host le configurazioni di bit **01** ed **10**
- La netmask da usare per una tale subnet è quindi **/30** ovvero **255.255.255.252**





Esempio di variable length subnetting (1)

- Blocco di N=2^8=256 indirizzi consecutivi identificato dal prefisso ***/24***
- Si debba ripartire il blocco in 5 sottoreti di dimensione: 128, 64, 16, 16, 4
  - In ciascun blocco due indirizzi sono riservati e pertanto non assegnabili a host
- Il numero totale di indirizzi usati è 128+64+16+16+4=228
  - Dei 256 indirizzi disponibili ne avanzeranno 28
- La ripartizione è effettuata in blocchi di dimensione decrescente



Ripartizione di un blocco di N=256 indirizzi in subnet di dimensione diversa

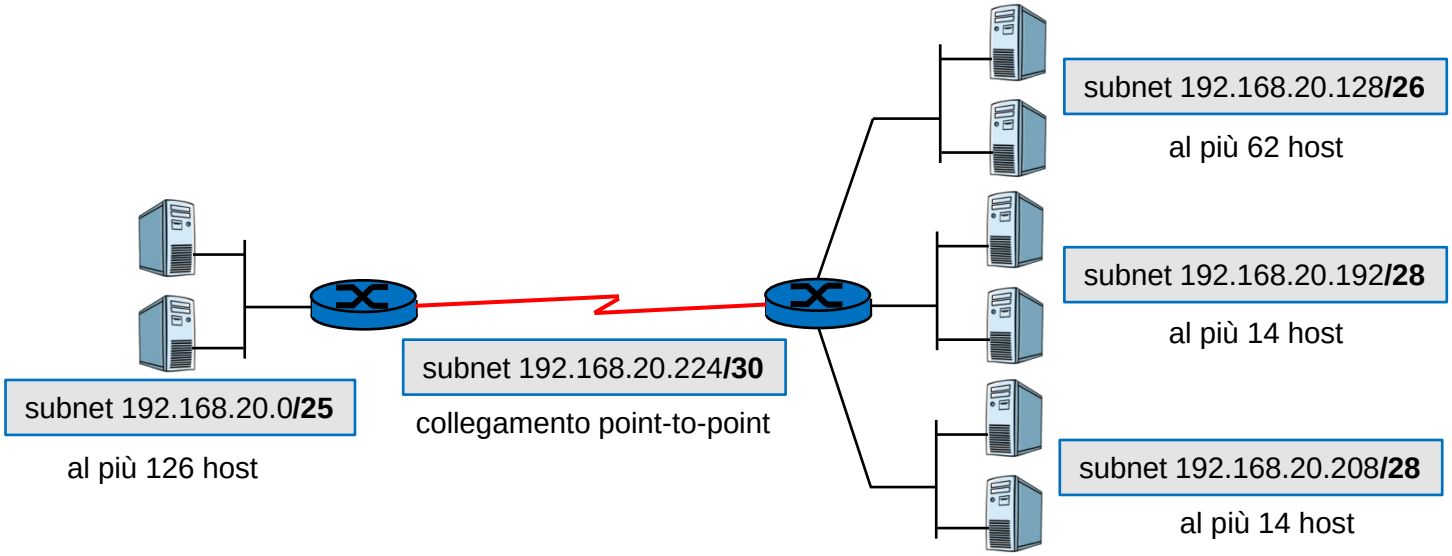
Esempio di variable length subnetting (2)

- Blocco di **N=2^8=256** indirizzi **192.168.20.0/24**
  ripartito 5 in subnet di dimensione: 128, 64, 16, 16, 4 indirizzi

Subnet	Subnet Address	Host Range	Broadcast Address
0	192.168.20.0 /25	192.168.20.1 to 192.168.20.126	192.168.20.127
1	192.168.20.128 /26	192.168.20.129 to 192.168.20.190	192.168.20.191
2	192.168.20.192 /28	192.168.20.193 to 192.168.20.206	192.168.20.207
3	192.168.20.208 /28	192.168.20.209 to 192.168.20.222	192.168.20.223
4	192.168.20.224 /30	192.168.20.225 to 192.168.20.226	192.168.20.227
Unused	192.168.20.228 to 192.168.20.255		

Esempio di variable length subnetting (3)

- Blocco di **N=2^8=256** indirizzi **192.168.20.0/24** ripartito 5 in subnet di dimensione: 128, 64, 16, 16, 4 indirizzi
- Esempio di rete a cui si applica il piano di indirizzamento determinato dal subnetting



Funzioni di un router: forwarding e routing

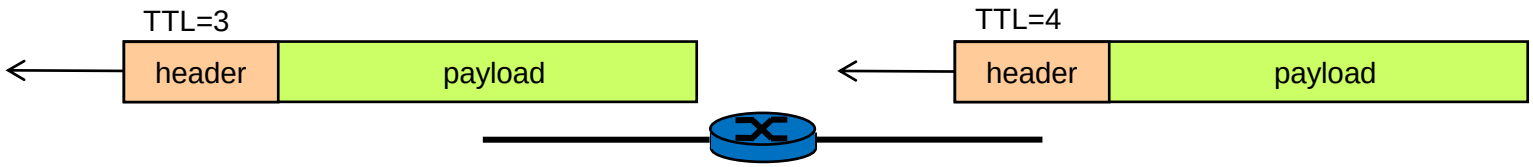
- Un router è un dispositivo dotato di più interfacce di rete che serve a collegare due o più reti tra di loro
- Un router ha il compito di inoltrare pacchetti nella rete verso la destinazione finale
- All'interno del router sono esplicate due funzioni fondamentali: **forwarding** e **routing**
- La funzione di ***forwarding*** consiste nell'inoltrare ciascun pacchetto che entra da un'interfaccia verso un'altra interfaccia
  - L'azione di forwarding effettuata dai router deve essere coordinata, in modo da far sì che un pacchetto, generato da un qualunque host mittente, possa arrivare verso un qualsiasi host destinatario
- La funzione di ***routing*** ha il compito di determinare i percorsi (*path*)
- Le due funzioni sono svolte contemporaneamente da due distinte sezioni del router:

Forwarding: funzione esplicata dal ***data plane***
Routing: funzione esplicata dal ***control plane***

- Il data plane deve essere in grado di operare alla velocità dei link
  - La funzione di forwarding è tipicamente realizzata mediante hardware specializzato
- Il control plane può operare a velocità più bassa (le scelte di percorso cambiano nell'ordine dei secondi)
  - La funzione di routing è tipicamente realizzata mediante software eseguito da CPU

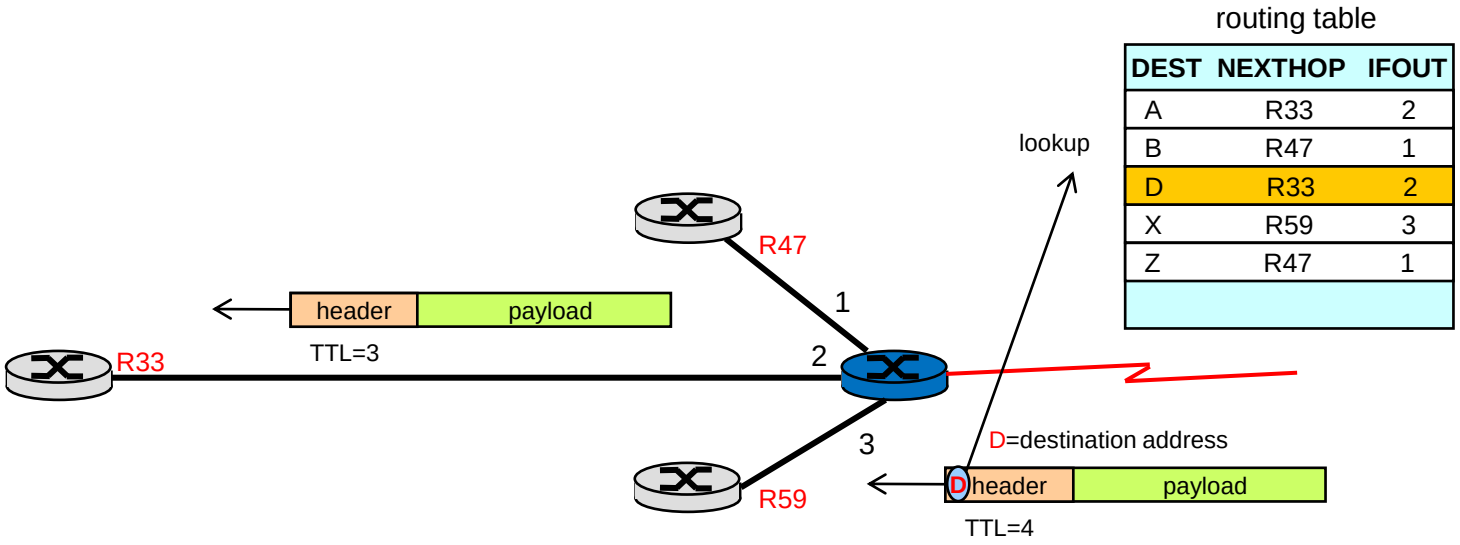
Funzioni di un router IP: forwarding (1)

- Un router IP è un dispositivo dotato di più interfacce di rete che serve a collegare due o più reti tra di loro
- A ciascuna interfaccia di un router è assegnato un indirizzo IP appartenente alla subnet associata alla rete a cui l'interfaccia si collega
- Internamente, il router identifica le proprie interfacce mediante degli identificatori locali come fa0, eth0, eth1, ecc.
- La funzione di ***forwarding*** svolta da un router IP è la seguente:
  - Per ciascun pacchetto, viene determinata l'interfaccia di uscita sulla base dell'indirizzo IP destinazione contenuto nel pacchetto
  - Prima della ritrasmissione, il campo TTL (*time-to-live*) nell'header del pacchetto inoltrato viene decrementato di 1
    - Se il TTL diventa zero, il pacchetto non è inoltrato ma viene eliminato
  - La modifica del TTL impone il ricalcolo del valore del campo *header checksum*



Funzioni di un router IP: forwarding (2)

- La scelta dell'interfaccia verso la quale il router realizza la ritrasmissione è determinata dall'indirizzo IP del destinatario del pacchetto
- Tale scelta è operata sulla base delle regole di instradamento contenute in una tabella: la ***tabella di routing***
- Ogni volta che il router deve inoltrare un pacchetto, viene consultata la tabella di routing per determinare l'interfaccia di uscita del pacchetto
- Il router effettua un'operazione di ricerca nella tabella (*lookup*) per determinare la regola da applicare



Funzioni di un router IP: forwarding (3)

- Nella tabella di routing c'è scritto, per ogni destinazione:
  - L'indirizzo IP del nexthop router
  - L'identificativo locale dell'interfaccia tramite la quale si raggiunge il nexthop
- Non è plausibile avere una regola per ciascun possibile indirizzo IP di destinazione:

  2^{32} = circa 4 miliardi di indirizzi
- Occorrono tecniche che consentano di compattare le regole nelle tabelle di routing
- Tutti i blocchi di indirizzi consecutivi che hanno lo stesso prefisso e lo stesso nexthop router sono rappresentati nella tabella di routing da una sola regola
- L'operazione di lookup nella tabella di routing viene effettuata con il criterio detto

  longest prefix match
- Una ***regola di default*** è di solito presente e si applica a tutte le destinazioni per le quali non c'è una regola esplicita nella tabella
- Esempio:

Destination Prefix (binary)	Destination Prefix (decimal)	Output Interface
11001000 00010111 00010	200.23.16.0/21	0
11001000 00010111 00011	200.23.24.0/21	2
11001000 00010111 00011000	200.23.24.0/24	1
default	default	3

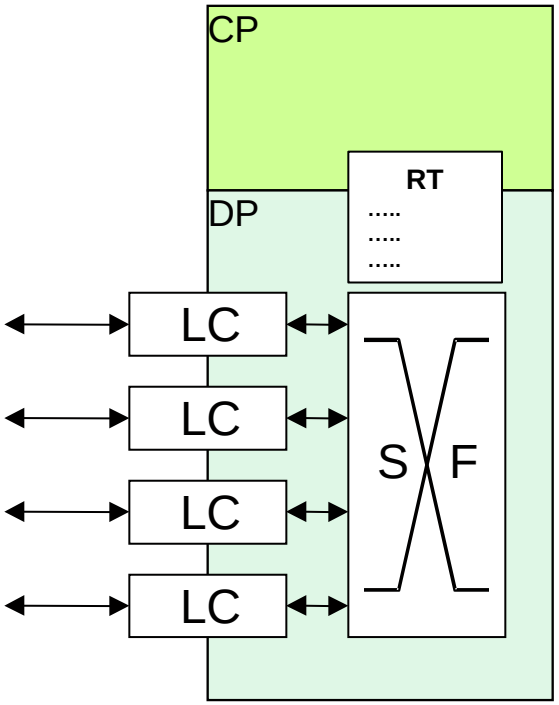
Pacchetto con destinazione: **200.23.24.17** cioè 11001000 00010111 00011000 00010001

Longest prefix match con la terza regola → **output interface = 1**

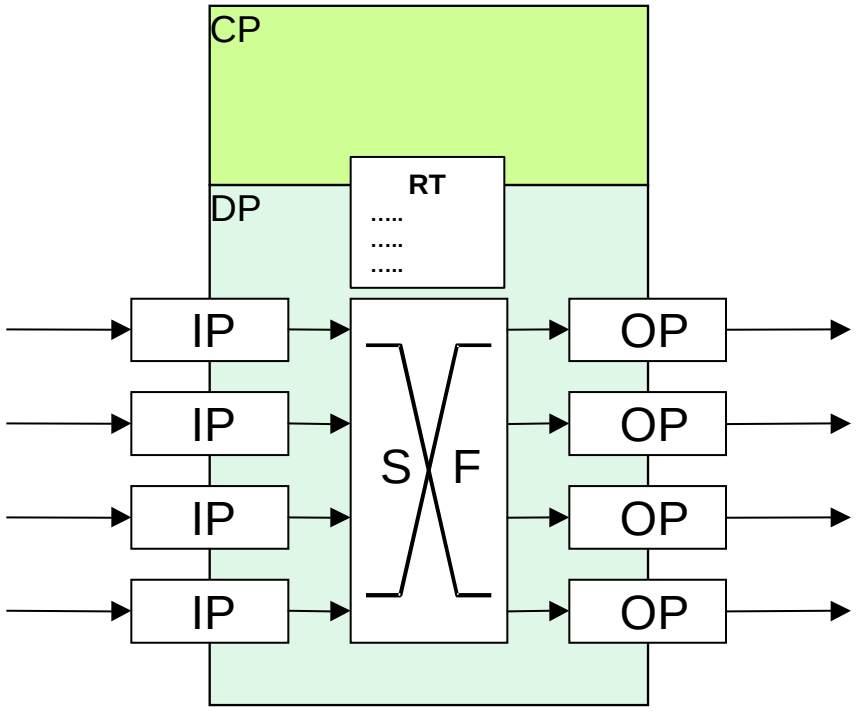
Pacchetto con destinazione: **200.23.25.11** cioè 11001000 00010111 00011001 00001011

Longest prefix match con la seconda regola → **output interface = 2**

Struttura interna di un router (1)



struttura fisica

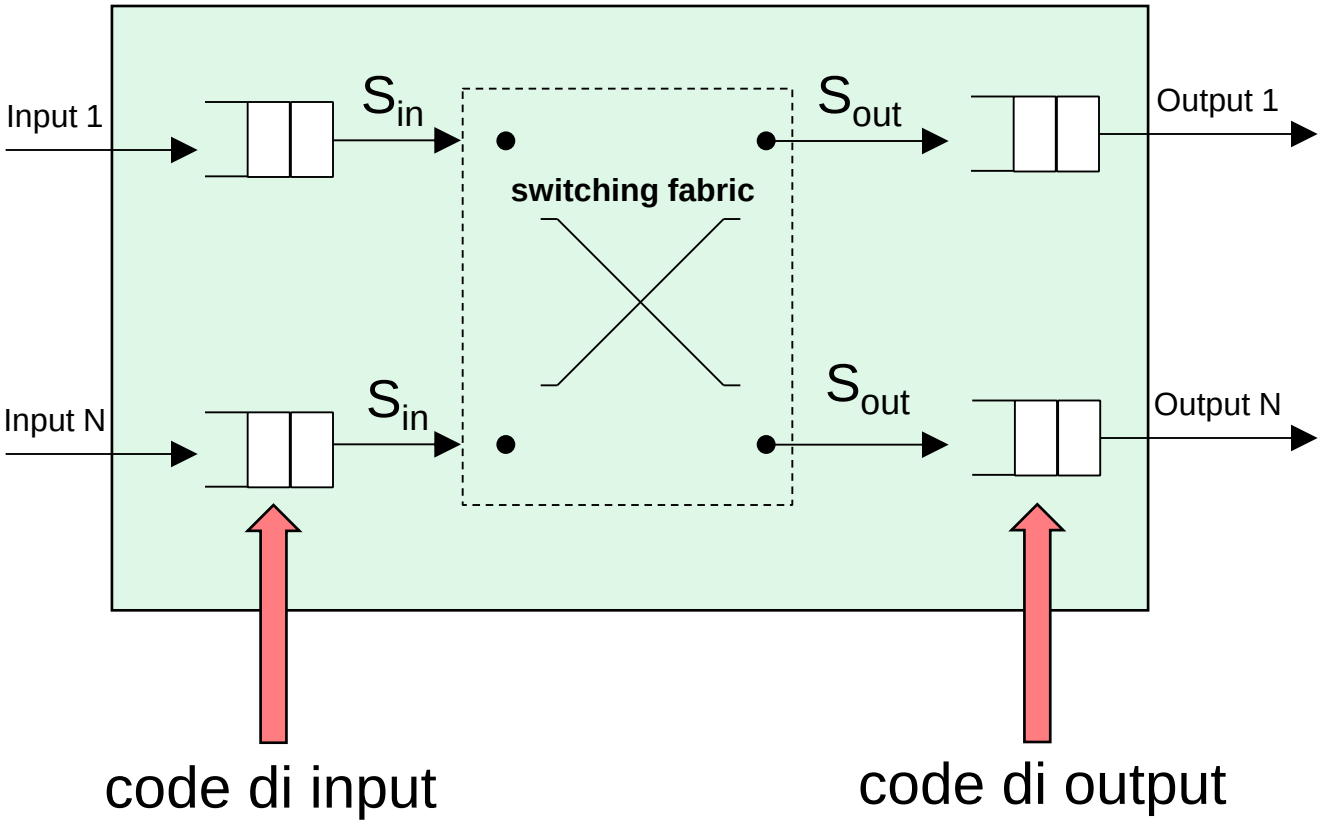


struttura logica

CP: Control Plane
DP: Data Plane
RT: Routing Table

LC: Line Card
SF: Switching Fabric
IP: Input Processor
OP: Output Processor

Struttura interna di un router (2)



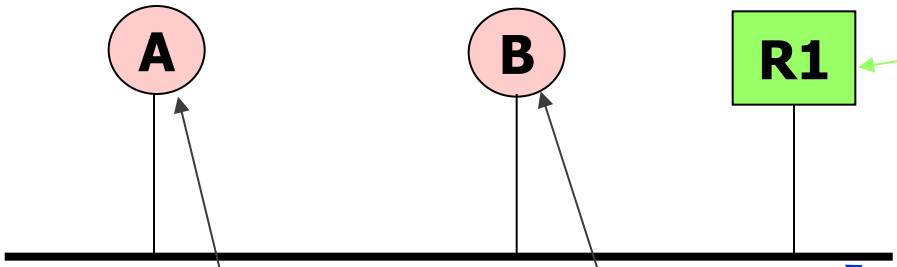
Router IP and options processing

- Taken from RFC 7126
- *From about 1995 onwards, a growing number of IP routers have incorporated silicon specialized for IP packet processing (i.e., Field-Programmable Gate Array (FPGA) or Application-Specific Integrated Circuit (ASIC)), thereby separating the function of IP packet forwarding from the other functions of the router.*
- *Routers with a silicon packet-forwarding engine can handle high volumes of IP packets (per second) containing IP options without any adverse impact on packet-forwarding rates or on the router's control plane (e.g., general-purpose CPU).*
- *Some implementations have a configuration knob simply to forward all IP packets containing IP options at wire-speed in silicon, as if the IP packet did not contain any IP options ("ignore options & forward").*
- *Other implementations support wire-speed silicon-based packet filtering, thereby enabling packets containing certain IP options to be selectively dropped ("drop"), packets containing certain other IP options to have those IP options ignored ("ignore options & forward"), and other packets containing different IP options to have those options processed, either on a general-purpose CPU or using custom logic (e.g., FPGA, ASIC), while the packet is being forwarded ("process option & forward").*
- *Broadly speaking, any IP packet that requires processing by an IP router's general-purpose CPU can be (used by) a DDoS to the routes*

Routing statico e dinamico

- Un router esplica la funzione di forwarding dei pacchetti consultando, per ogni pacchetto processato, la ***tabella di routing***
- La costruzione della tabella di routing è un compito che può essere svolto in 2 modi:
  - **routing statico:** l'amministratore di rete, conoscendo la topologia della rete, determina i percorsi tra qualunque coppia sorgente-destinazione e conseguentemente configura ciascun router con le opportune regole di inoltro
  - **routing dinamico:** in ciascun router, nel *control plane*, opera un programma il quale, mediante lo scambio di informazioni con i router vicini, determina (attraverso un algoritmo) i percorsi verso qualunque destinazione e conseguentemente crea nella tabella di routing le regole corrispondenti
    - Lo scambio di informazioni tra i router necessario all'esecuzione dell'algoritmo di routing è regolato da appositi protocolli di comunicazione: i **protocolli di routing**

ARP: primo caso A→B (1/3)



Default router

IP: 192.168.1.254
netmask: 255.255.255.0

Rete IP di classe C

rete: 192.168.1.0
netmask: 255.255.255.0

IP: 192.168.1.2
netmask: 255.255.255.0

IP: 192.168.1.1
netmask: 255.255.255.0

A ha intenzione di inviare un pacchetto a **B**
Domanda: come fa **A** a sapere che **B** è nella sua stessa sottorete IP?

Risposta: attraverso la netmask!

ARP: primo caso (2/3)

- Ogni computer ha un indirizzo IP ed una netmask
- La netmask serve ad individuare la propria sottorete IP
  - In Windows digitare il comando: **`ipconfig /all`**
- Il computer **A** esegue una AND bit-a-bit tra l'indirizzo IP destinazione e la propria netmask.
  - Nel caso precedente:

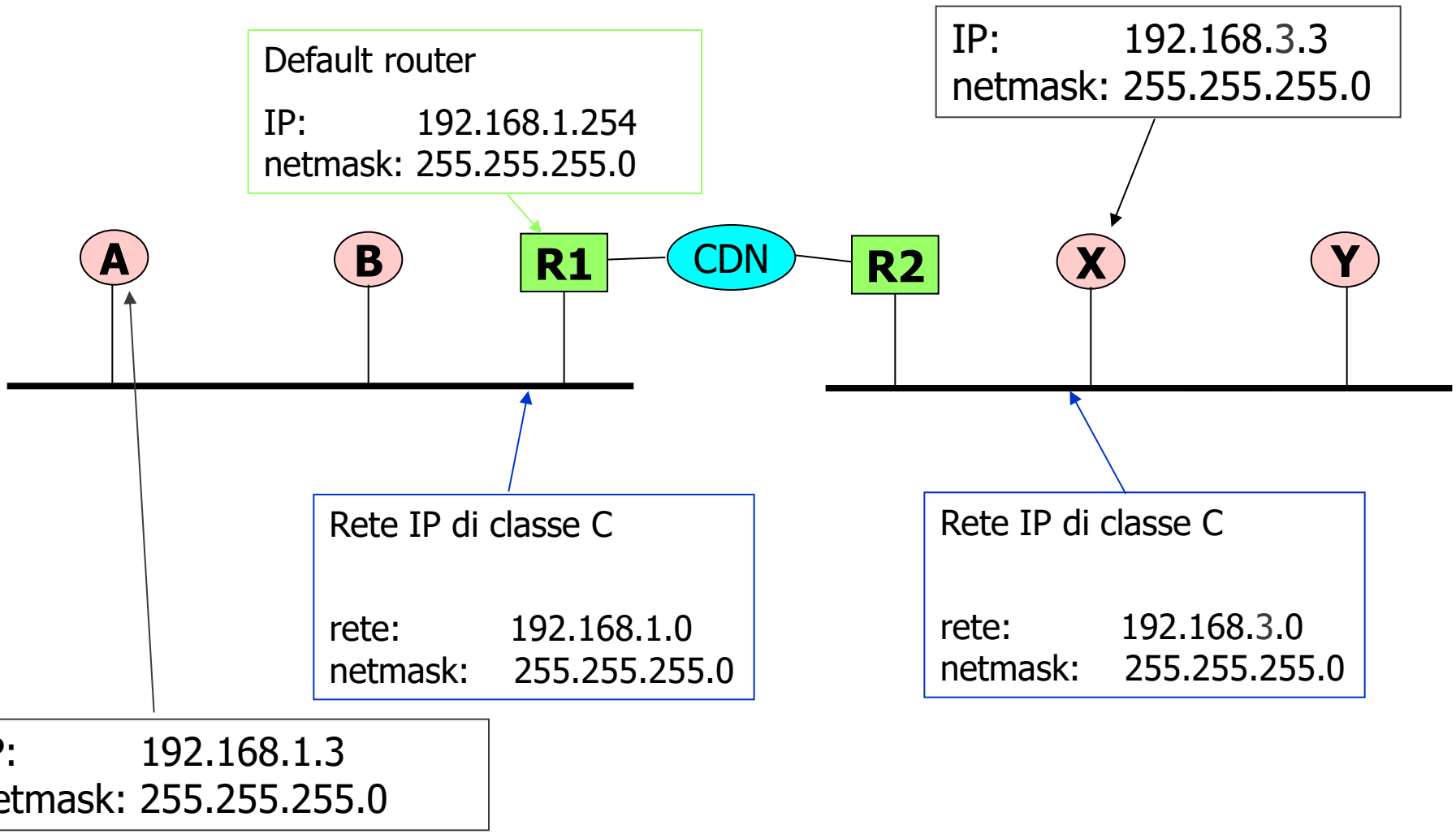
IP di B	192.168.1.2
	AND
netmask A	255.255.255.0
	=
	192.168.1.0

E' proprio l'indirizzo della sottorete IP cui appartiene A

ARP: primo caso (3/3)

- Se il computer **B** è sulla stessa sottorete IP la comunicazione avviene direttamente da **A** a **B**
  - A manda un pacchetto *ARP request* in broadcast per conoscere il MAC address di B
    - tale pacchetto contiene, nel campo **DEST IP,** l'indirizzo IP di B

ARP: secondo caso A→X (1/2)



ARP: secondo caso (2/2)

- Se **A** intende mandare un pacchetto a **X,** l'operazione di AND bit-a-bit tra la netmask e l'indirizzo IP di **X** fornisce un risultato differente
  - il destinatario non è nella stessa subnet IP del mittente

IP di X	192.168.3.3
	AND
netmask A	255.255.255.0
	=
	192.168.3.0

Non è l'indirizzo della sottorete cui appartiene A →
Occorre inviare il pacchetto al router

- In questo caso, pertanto, al primo hop il destinatario del livello 2 è l'interaccia del router che appartiene alla subnet di A

- A prepara un pacchetto ARP in cui si specifica come indirizzo IP DEST proprio l'indirizzo IP del router

ARP: ricapitolando…

- Operazione di AND logico tra l'indirizzo IP della destinazione e la propria netmask:
  - Se il risultato fornisce l'indirizzo della propria subnet IP:
    - Invia una richiesta ARP per risolvere l'indirizzo della destinazione
  - …altrimenti:
    - Il pacchetto deve essere inviato al router di default:
      - Nel caso in cui l'indirizzo MAC del router non sia noto:
        - Invia una richiesta ARP per risolvere l'indirizzo IP del router