



*Corso di*

***SISTEMI TELEMATICI***

*a.a. 2011-2012*

**Il protocollo IPv4**  
(RFC 791)



# Il protocollo IP

- IP è un protocollo di strato 3 e fornisce le seguenti funzionalità:
  - × definisce lo schema di indirizzamento
  - × definisce il percorso di un'unità dati verso la destinazione (instradamento)
  - × definisce l'unità base per il trasferimento dei dati e ne specifica il formato
  - × definisce le modalità per la segmentazione e l'aggregazione delle unità dati (il risultato dell'operazione di segmentazione verrà chiamato "frammento")
- Il protocollo IP fornisce un servizio di trasferimento delle unità informative con modalità a datagramma, senza connessione e inaffidabile



# Il protocollo IP

- Il termine “senza connessione” indica che IP tratta ciascuna unità informativa indipendentemente dalle altre
  - ✗ ognuna può seguire una strada diversa per arrivare a destinazione (non esiste il concetto di circuito logico e di connessione)
  - ✗ IP non mantiene informazioni di stato sulle unità dati inoltrate
- Il termine “inaffidabile” indica che non è garantita la consegna di un’unità informativa a destinazione (servizio best-effort)
  - ✗ un’unità dati può essere persa, duplicata, ritardata o consegnata fuori sequenza



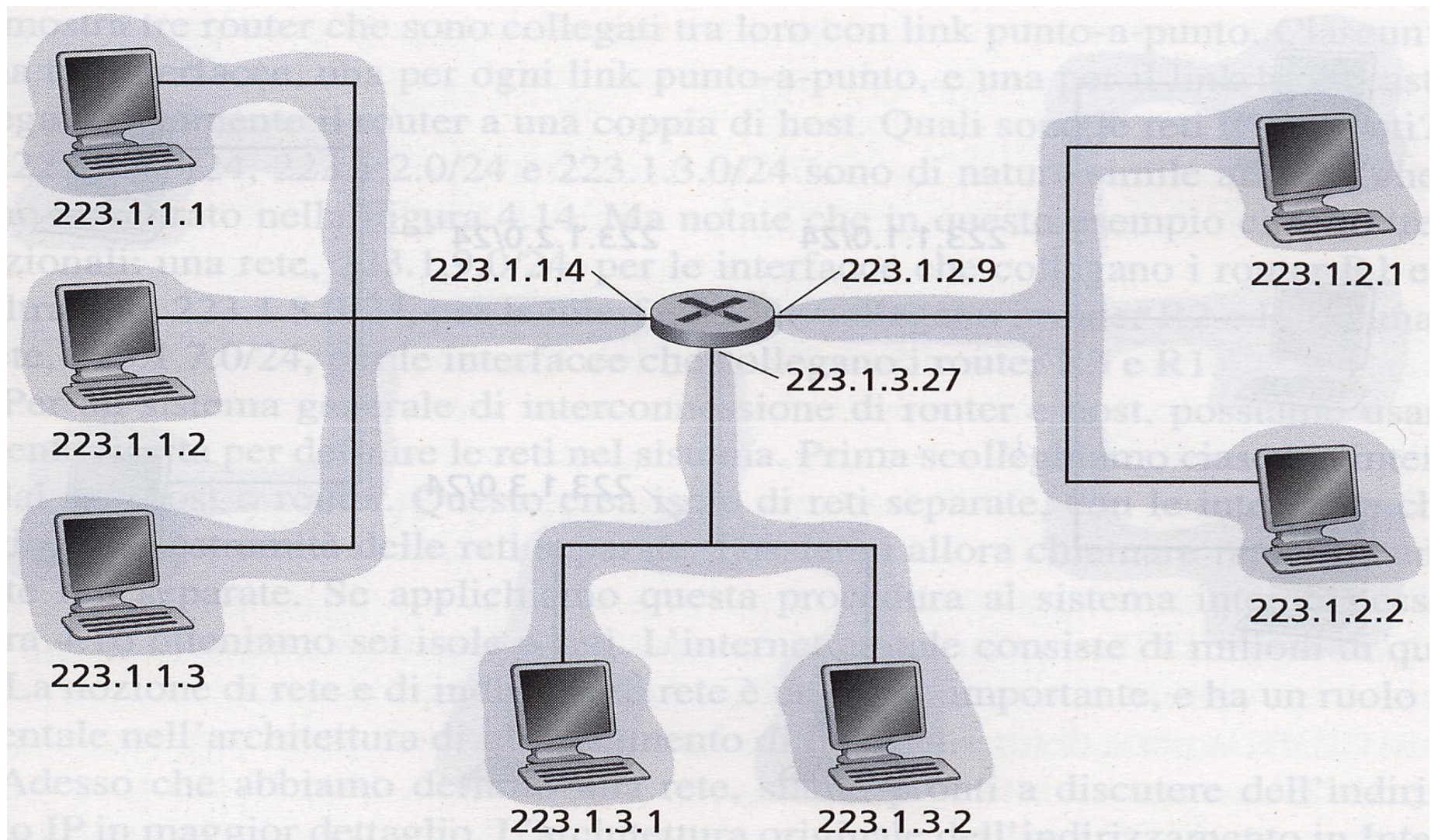
# Il protocollo IP

- Lo strato IP non fornisce garanzia sulla qualità di servizio (integrità informativa, trasparenza temporale, etc.)
  - ✗ il compito di garantire la qualità di servizio è demandato agli strati superiori residenti negli host
  - ✗ la qualità dipende dalle caratteristiche delle sotto-reti attraversate





# Il protocollo IP



**Esempio di comunicazione a livello rete tramite indirizzamento IP**



# Formato dell'unità dati

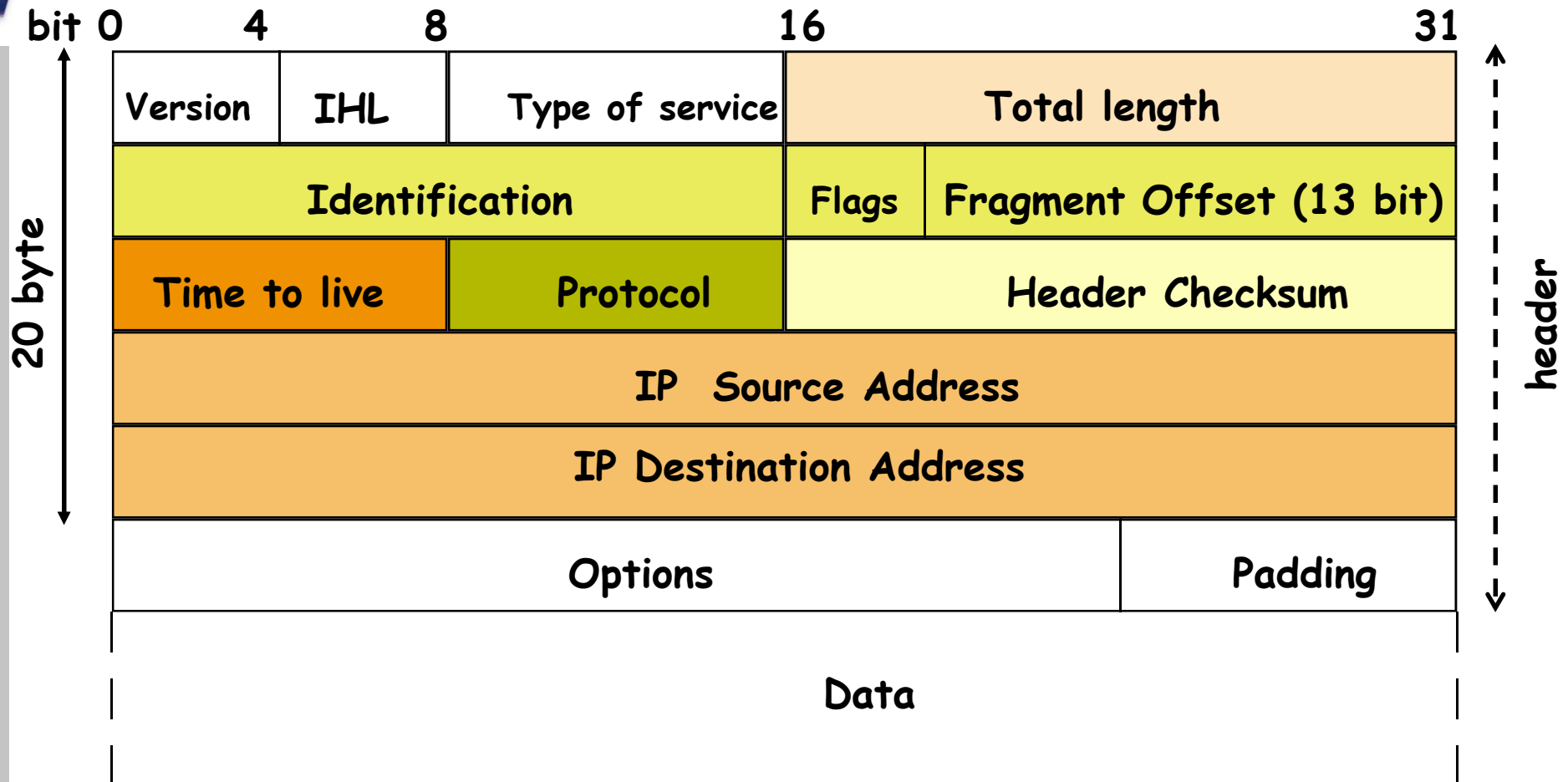
- Le unità-dati dello strato IP sono dette datagrammi
- Il datagramma è composto da un campo informativo che contiene i dati di utente e da un'intestazione (header)
- Il datagramma IP ha lunghezza variabile

header: 20 ÷ 60 byte

- La lunghezza massima di un datagramma IP è 65536 ottetti



# Formato dell'unità dati





# Header IP

- ✕ **Vers (4 bit):** versione del protocollo IP
- ✕ **IHL (Internet Header Length) (4 bit):** lunghezza dell'intestazione (in parole di 32 bit)
  - Il valore minimo nel caso senza opzioni è 5 (20 byte)
- ✕ **Type of Service (TOS) (8 bit):** specifica parametri della qualità di servizio richiesti dall'utente
  - nella versione originale di IP (RFC791), TOS era formato da 2 sottocampi: un sub-campo (precedence) di 3 bit (valori 0-7) per specificare l'importanza del datagramma; un sub-campo di 3 bit per richiedere particolari prestazioni (se=1):
    - D-bit (delay):** basso ritardo; **T-bit (throughput):** alta portata; **R-bit (reliability):** alta affidabilità
  - i 2 bit restanti sono riservati per usi futuri

0	2	3	4	5	6	7
precedence	D	T	R	0	0	





# Header IP

- nella versione aggiornata del campo TOS (RFC1349), uno dei 2 bit è aggiunto al secondo sub-campo per indicare:

**1000** minimizzare il ritardo (delay)

**0100** massimizzare la portata (throughput)

**0010** massimizzare l'affidabilità (reliability)

**0001** minimizzare il costo

**0000** servizio normale

0	2	3	6	7
precedence	Type			0

- Specificare il TOS significa che la rete offrirà un servizio migliore se è possibile, ma non significa che negherà il servizio se ciò non è possibile  
cioè se **TOS=1000**, la rete proverà a scegliere il percorso con ritardo minore tra quelli disponibili, ma non scarterà il datagramma se il ritardo è troppo alto



# Header IP

- ✕ **Total length (16 bit):** specifica la lunghezza del datagramma, misurata in ottetti, includendo l'intestazione ed i dati ( $2^{16}=65536$  byte); la lunghezza è sempre multipla di 4 byte
  - tutti gli host devono essere preparati ad accettare datagrammi di lunghezza fino a 576 byte (interi o frammentati); che permette di trasmettere una quantità ragionevole di dati oltre all'intestazione
- ✕ **Identification (16 bit):** numero del datagramma; è assegnato dal processo sorgente al datagramma o ai suoi frammenti
  - il numero è generato da un contatore nell'host sorgente e incrementato ogni volta che viene generato un nuovo datagramma
  - ogni router che segmenta il datagramma ricopia questo campo nell'intestazione di ogni frammento del datagramma di partenza



# Header IP

## ✕ **Flags (3 bit):** è un campo di bit di controllo

- X (bit 0): non usato e posto a zero
- DF (bit 1): Don't Fragment (se è 1); se 0 indica che il datagramma può essere frammentato
- MF (bit2): More Fragment (se è 1) indica che seguono altri frammenti; se 0 indica che è l'ultimo frammento

## ✕ **Fragment Offset (13 bit):** indica la posizione del frammento all'interno del datagramma originario

- misurato in unità di 8 byte (la lunghezza di un frammento è pari a un multiplo di 8 byte); il campo può numerare 8192 frammenti ( $2^{13}$ ) di 8 byte ciascuno (per un totale di 65536 byte); il primo frammento ha offset 0
- ogni sistema deve essere in grado di inoltrare datagrammi di 68 byte senza ulteriore frammentazione



# Header IP

✖ **Time to Live (8 bit):** indica quanto tempo il datagramma può rimanere all'interno della rete

- è inizializzato dall'host, quando genera il datagramma, col tempo concesso per attraversare l'inter-rete; questo valore viene decrementato da ogni router incontrato dal datagramma; quando il valore diventa zero il datagramma viene scartato
- così si impedisce a un datagramma di circolare all'infinito nella rete (in caso di instradamento errato su un cammino chiuso)
- il campo è decrementato a passi minimi di 1 s, il valore max è 255s ( $2^8$ )
- nelle implementazioni più recenti, il campo è definito in numero di salti (hop); per salto si intende l'attraversamento di un router e quindi un datagramma può attraversare al max 256 router prima di essere scartato



# Header IP

- ✖ **Protocol (8 bit):** indica a quale protocollo dello stato superiore deve essere trasferito il contenuto informativo del datagramma
- ✖ **Header Cecksum (16 bit):** l'intestazione è protetta da un controllo di errore
  - il contenuto del campo si ottiene considerando i bit dell'intestazione a gruppi di 16, effettuandone la somma e memorizzando nel campo il complemento a 1 del risultato
  - I controlli non vengono eseguiti sul flusso dei dati dell'utente. Se da un lato ciò consente di usare un algoritmo di checksum piuttosto semplice, in quanto non deve operare su molti byte, dall'altro richiede che un protocollo di livello superiore esegua un controllo degli errori sui dati dell'utente.
- **Source/Destination Address (32 bit):** indirizzo di sorgente/destinazione IP (dell'host, non dell'utente finale)



# Header IP

✖ **Options** (lunghezza variabile a multipli di 8 bit): è opzionale. La fine del campo è delimitata da un byte di 0

- Record Route Option (RRO): permette al mittente di creare una lista vuota di indirizzi IP; ogni nodo attraversato inserisce il suo indirizzo nella lista
- Source Route Option (SRO): consente al mittente di specificare i nodi attraverso i quali vuole che transiti il datagramma (per scopi gestionali e di test)

Type (8)	Length (8)	Pointer (8)	Route data (var)
----------	------------	-------------	---------------------

- Timestamp Option: come RRO, ma in più ogni nodo specifica l'istante temporale in cui il datagramma attraversa i diversi nodi

✖ **Padding (riempitivo)**: rende la lunghezza dell'intestazione un multiplo intero di 32 bit mediante introduzione di zeri





# Header IP

✖ **Options** (lunghezza variabile a multipli di 8 bit): è opzionale. La fine del campo è delimitata da un byte di 0

- Record Route Option (RRO): permette al mittente di creare una lista vuota di indirizzi IP; ogni nodo attraversato inserisce il suo indirizzo nella lista
- Source Route Option (SRO): consente al mittente di specificare i nodi attraverso i quali vuole che transiti il datagramma (per scopi gestionali e di test)

Type (8)	Length (8)	Pointer (8)	Route data (var)
----------	------------	-------------	---------------------

- Timestamp Option: come RRO, ma in più ogni nodo specifica l'istante temporale in cui il datagramma attraversa i diversi nodi

✖ **Padding (riempitivo)**: rende la lunghezza dell'intestazione un multiplo intero di 32 bit mediante introduzione di zeri