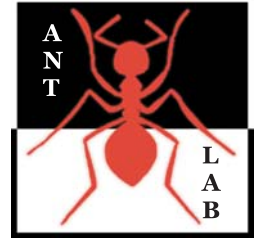




Politecnico di Milano

Advanced **N**etwork **T**echnologies **L**aboratory



Il protocollo IP (Internet Protocol)

- Servizi offerti da IP
- Formato del pacchetto IP

Il servizio di comunicazione offerto da IP

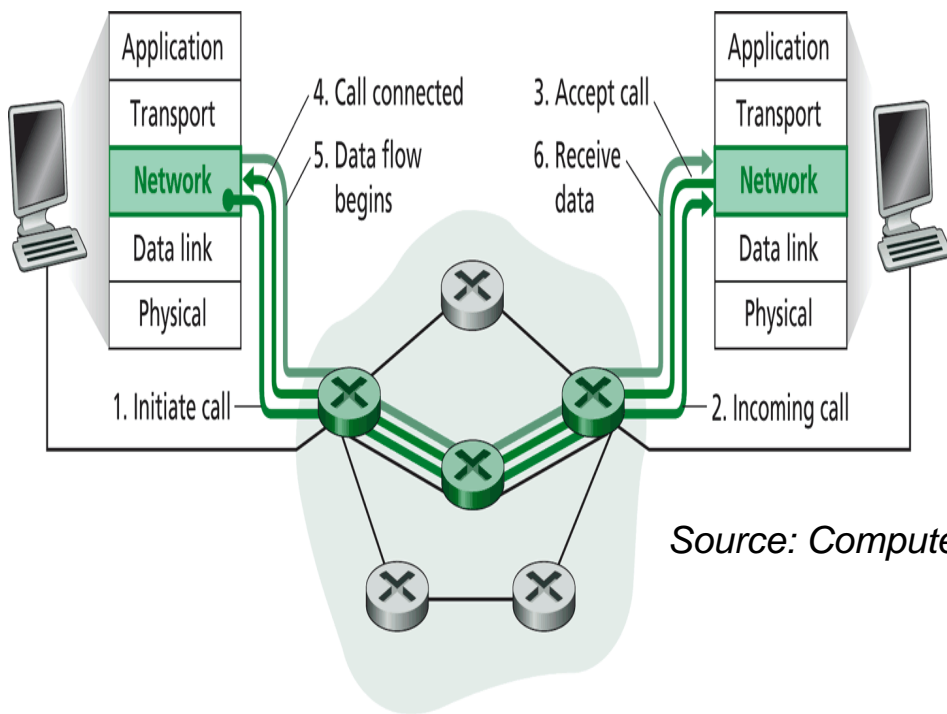
□ *Connectionless*

- progettato secondo un paradigma *packet-oriented* (o *datagram*)
- Due pacchetti (o datagrammi) destinati alla stesso host possono “essere trattati” in maniera diversa

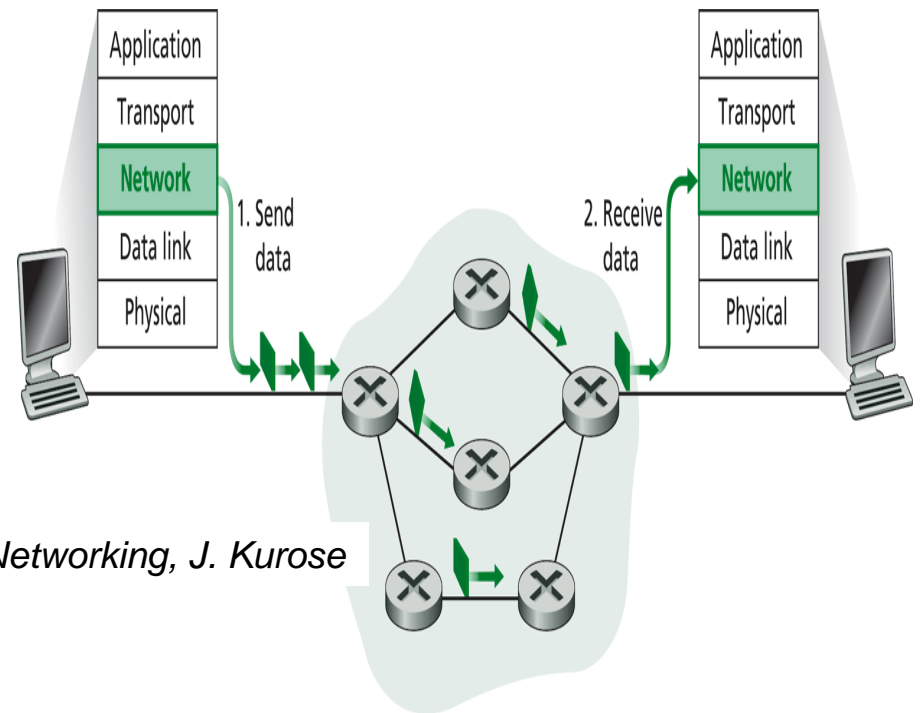
□ Non affidabile

- Consegna *best-effort* dei datagrammi senza garanzia di successo
- Analogia con il servizio postale ordinario

Pacchetto vs Circuito Virtuale



**Approccio a circuito virtuale:
ATM, X25, Frame Relay**

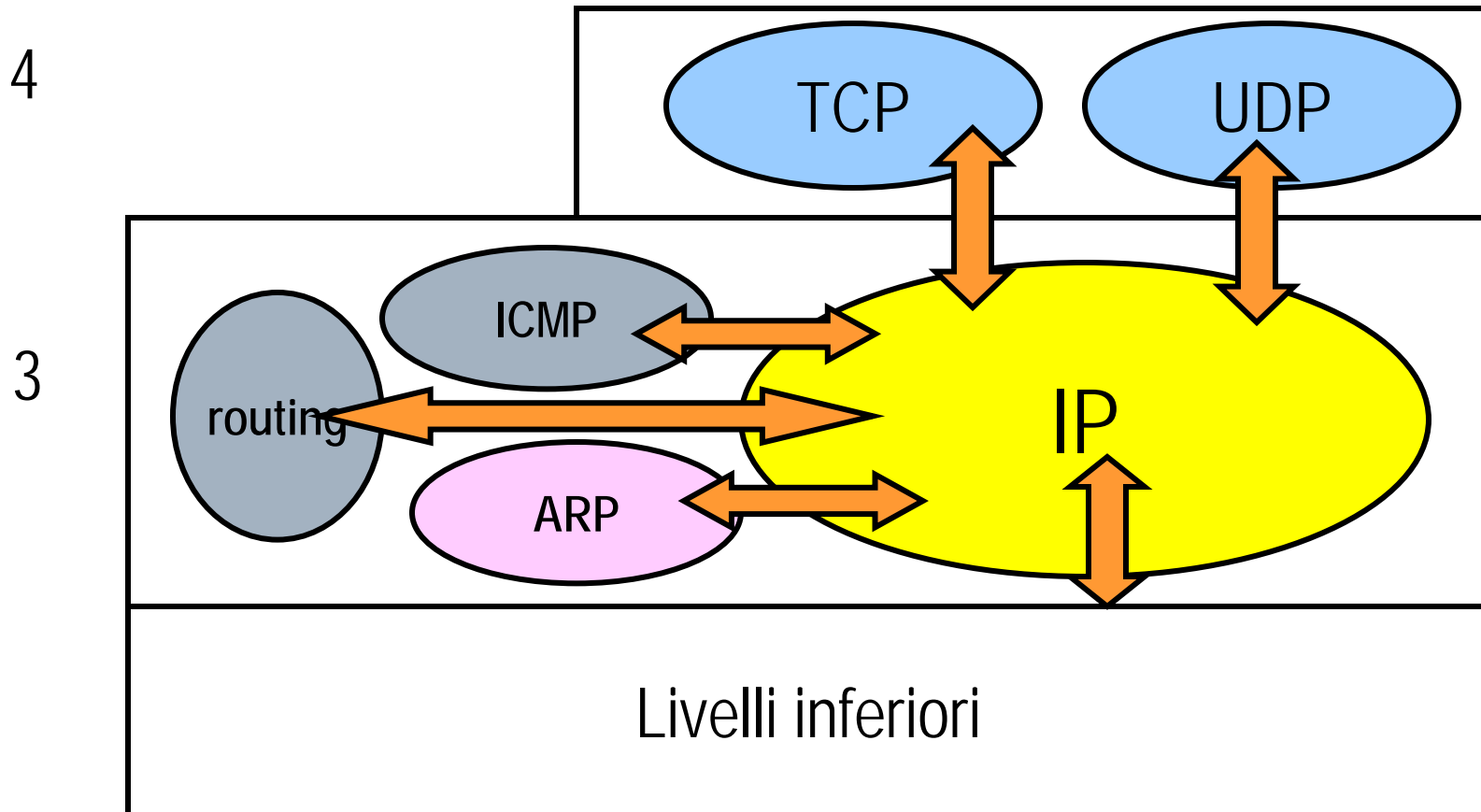


**Approccio a pacchetto:
IP**

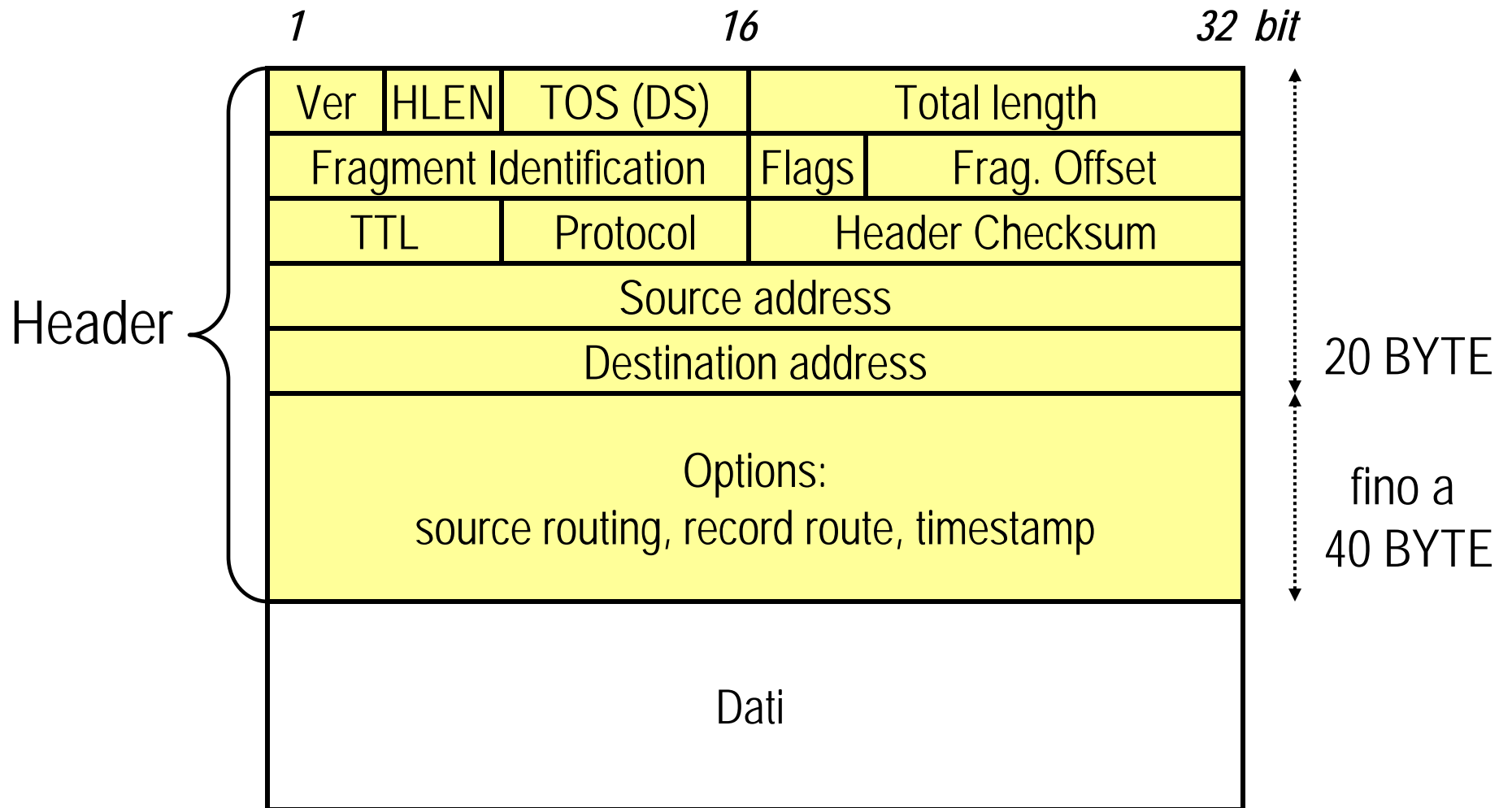
Altri Servizi Offerti da IP

- **Indirizzamento:** assegna un'indirizzo universalmente riconosciuto
- **Frammentazione/Deframmentazione:** frammenta/deframmenta i pacchetti se il livello locale lo richiede (IP è pensato per funzionare su molteplici tecnologie di livello inferiore)

Lo stack IP base



Il pacchetto (datagramma) IP



I campi dell'header IP

□ Ver (4 bit):

- *version*: indica la versione del protocollo; IPv4, IPv6. Se il campo VER non corrisponde alla versione del protocollo implementata sul router ricevente, il pacchetto viene scartato.

□ HLEN (4 bit)

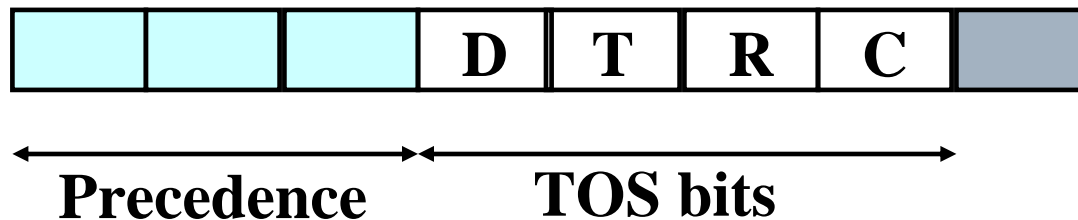
- *header length*: indica la lunghezza dell'header del pacchetto espressa in parole da 32 bit (max 64 byte)

□ Total length (16 bit):

- indica la lunghezza totale del pacchetto in byte: valore massimo $2^{16}=65536$; una volta sottratta la dimensione dell'header dà la lunghezza del payload. Serve solo se il livello sottostante effettua padding riempitivo.

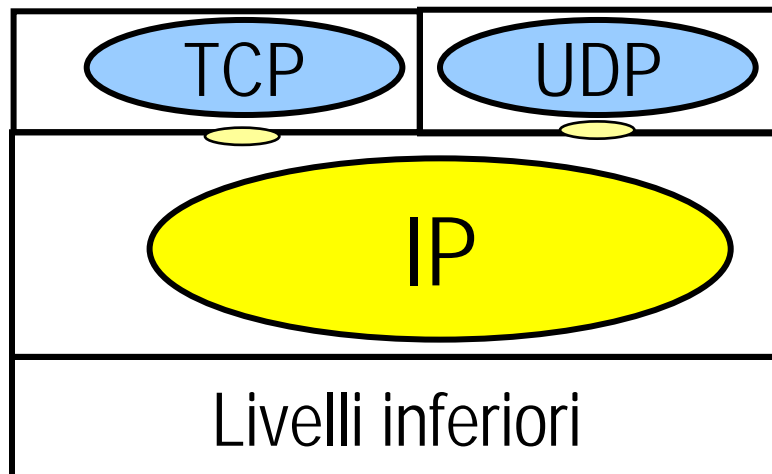
I campi dell'header IP

- *TOS type of service (8 bit)*
 - Recentemente cambiato in *Differentiated Services* usato per la gestione delle priorità nelle code dei router, e per garantire QoS



Il campo *Protocol*

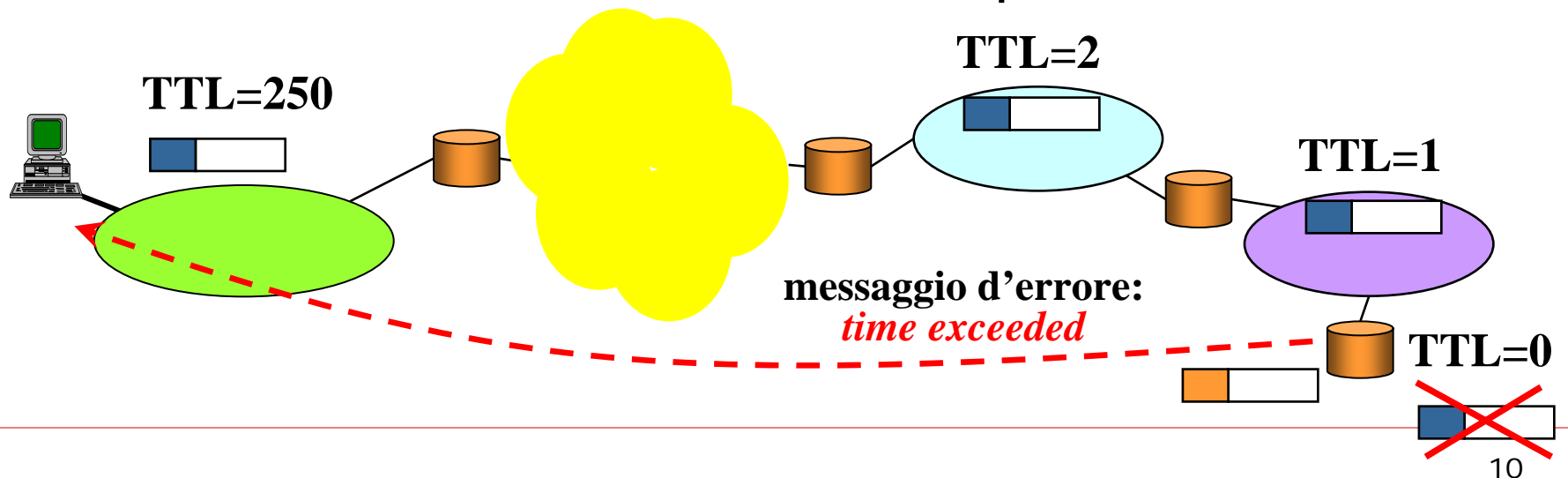
- ❑ E' un codice che indica il protocollo di livello superiore
- ❑ più protocolli di livello superiore possono usare IP (multiplazione)
- ❑ il codice identifica il SAP (*Service Access Point*) tra IP e il protocollo di livello superiore



Valore	Protocollo
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

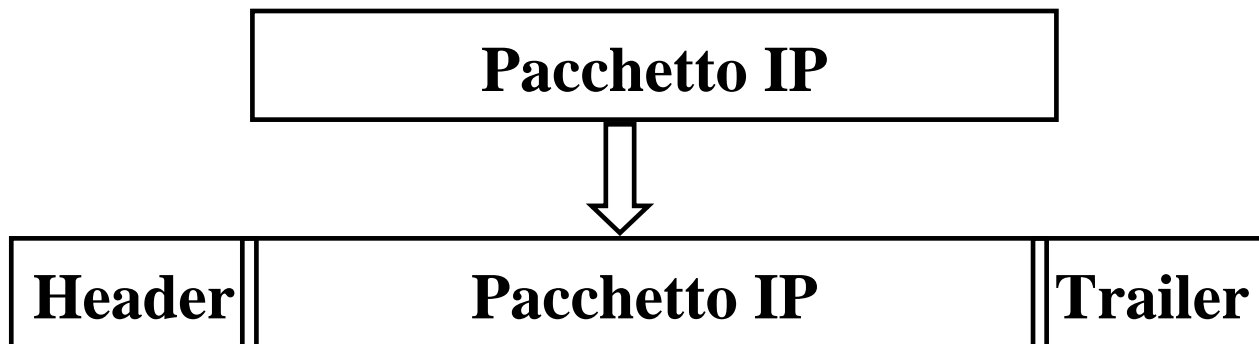
Il campo *Time To Live* (TTL)

- Il campo *TTL* viene settato ad un valore elevato da chi genera il pacchetto e viene decrementato da ogni router attraversato
- Se un router decrementa il valore e questo va a zero, il pacchetto viene scartato e viene generato un messaggio di errore verso la sorgente
- *Time-out* sulla validità di un pacchetto



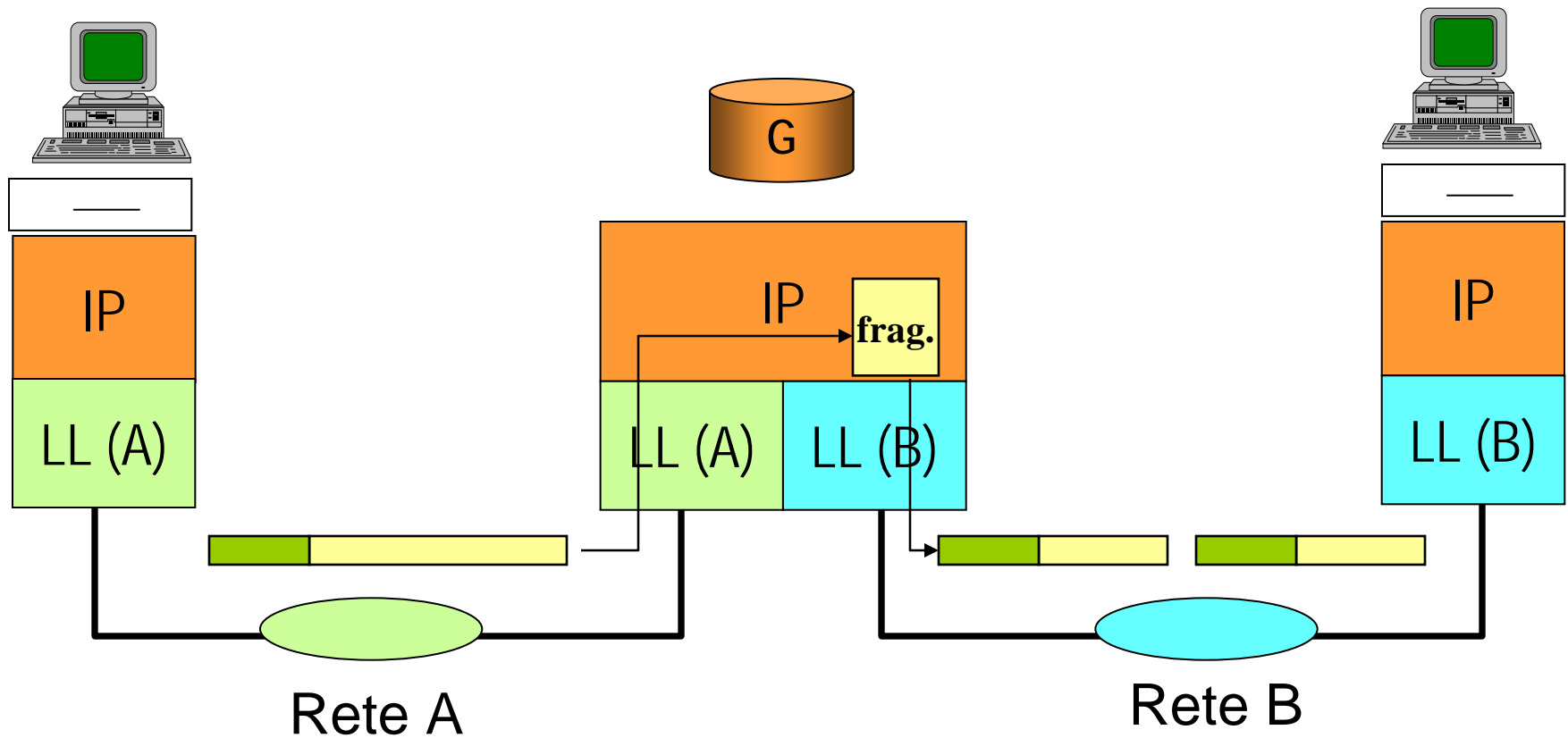
La Frammentazione (1)

- Alcuni protocolli di livello inferiore a cui IP si appoggia richiedono una dimensione massima del pacchetto IP (*Maximum Transfer Unit, MTU*) inferiore a 65536 bytes



Protocollo	MTU (byte)
Token Ring 16Mb/s	17914
FDDI	4352
Ethernet	1500

La Frammentazione (2)



La Frammentazione (3)

- prima di passare il pacchetto al livello inferiore IP divide il pacchetto in frammenti ciascuno con il suo header
- un frammento di un pacchetto può essere frammentato ulteriormente lungo il cammino
- i frammenti verranno ricomposti dall'entità IP del destinatario (frammenti di uno stesso pacchetto possono seguire diversi percorsi)
- i campi *Identification*, *Flags* e *Frag. Offset* sono usati per questo scopo

I campi usati per la frammentazione (1)

□ Identification (16 bit)

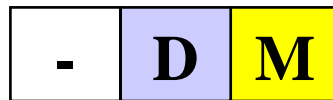
- è un campo che identifica tutti i frammenti di uno stesso pacchetto in modo univoco. E' scelto dall'IP che effettua la frammentazione

□ Frag. Offset (13 bit)

- I byte del pacchetto originale sono numerati da 0 al valore della lunghezza totale. Il campo *Frag. Offset* di ogni frammento riporta il numero di sequenza del primo byte del frammento.
- *esempio*: se un pacchetto di 2000 byte viene diviso in due da 1000 il primo frammento avrà un *Frag Offset* pari a 0 e il secondo pari a 1000/8

I campi usati per la frammentazione (2)

□ Flags



- il bit M (*More*) è pari a 0 solo nell'ultimo frammento
- il bit D (*Do not fragment*) viene posto a 1 quando non si vuole che lungo il percorso venga applicata la frammentazione
 - in questo caso, se la frammentazione fosse necessaria, il pacchetto sarebbe scartato e verrebbe generato un messaggio di errore

La Frammentazione in pratica

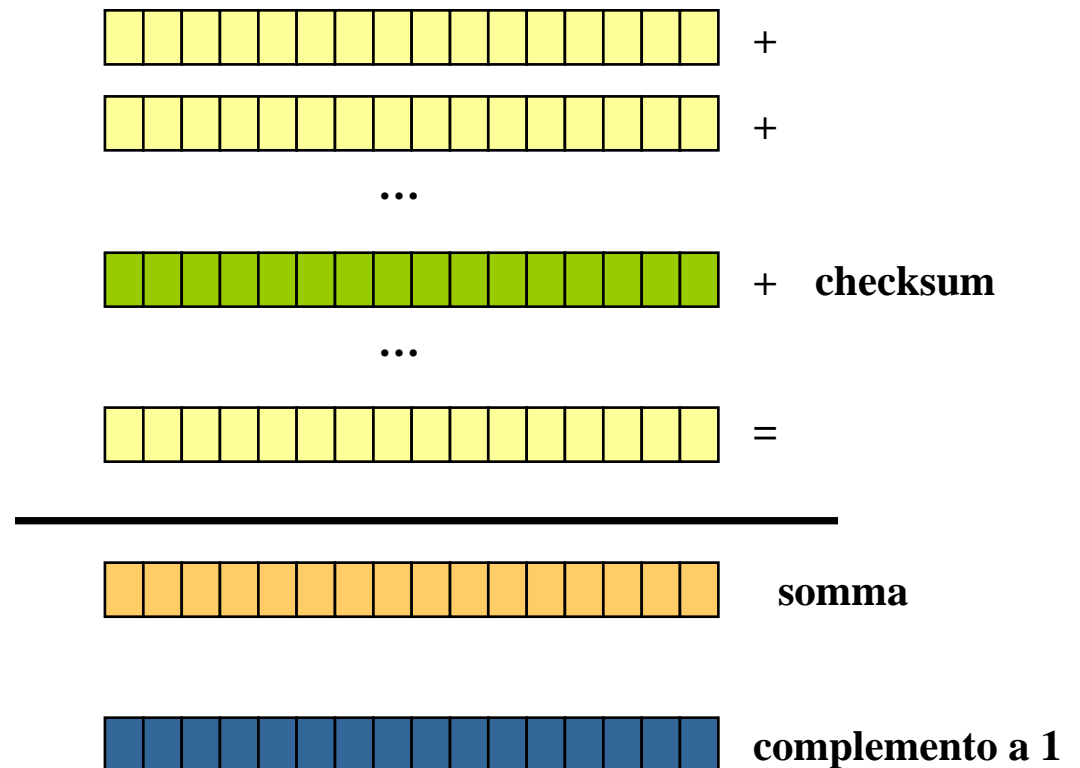
- ❑ L'overhead computazionale legato alla frammentazione può essere rilevante
- ❑ Si tende, quindi, a non frammentare limitando la dimensione dei segmenti che arrivano all'IP dal livello di trasporto.
- ❑ IP supporta tecnologie sottostanti in grado di gestire trame di almeno 576 byte
- ❑ Si pone la dimensione dei segmenti del livello di trasporto pari a 536 byte (+20byte TCP + 20byte IP)
- ❑ La maggior parte delle applicazioni per trasferimento di dati (HTTP) funzionano con dimensione di segmenti tra 512-536 byte.

Il campo *Checksum*: controllo di integrità

- ❑ Informazione ridondante inserita nell'header del pacchetto IP per controllo d'errore
- ❑ Il campo di *checksum* (16 bit) è calcolato dal trasmettitore ed inserito nell'header
- ❑ Il ricevitore ripete lo stesso calcolo sul pacchetto ricevuto (comprensivo di *checksum*)
- ❑ Se il risultato è soddisfacente accetta il pacchetto altrimenti lo scarta

Calcolo del Checksum lato ricevitore

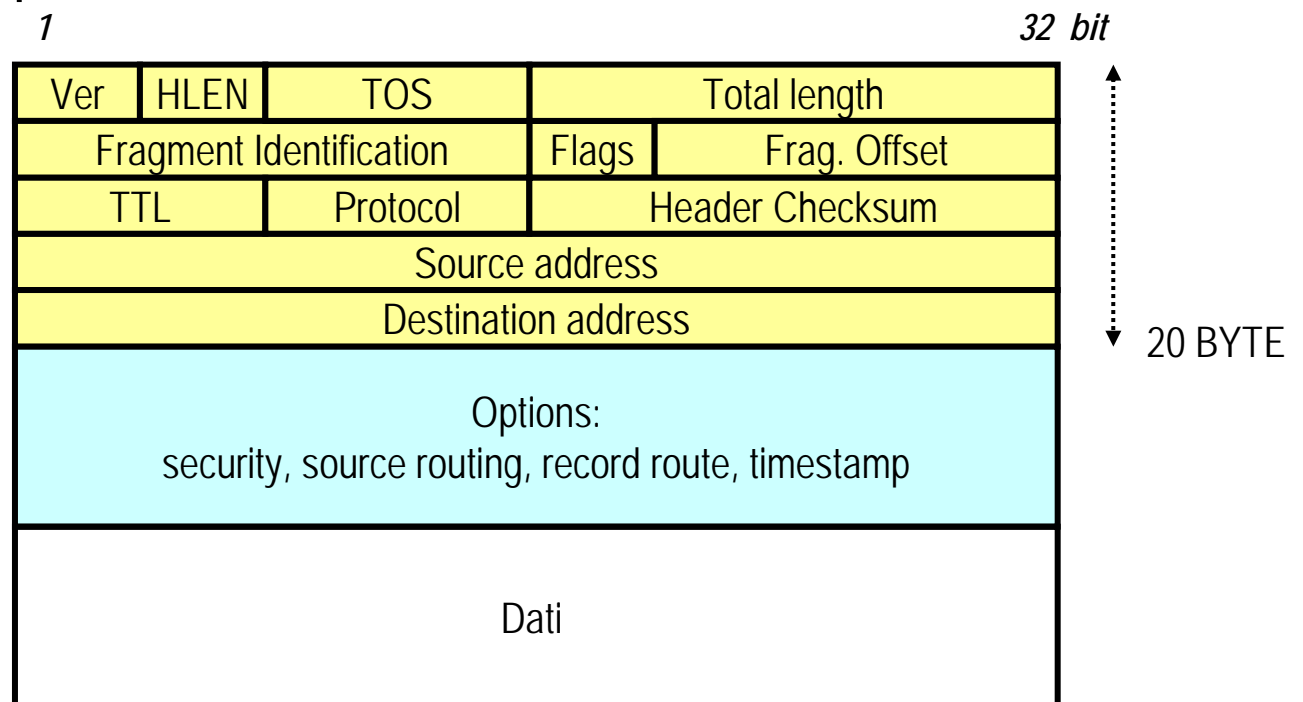
- L'header è diviso in blocchi da 16 bit
- Tutti i blocchi vengono sommati modulo 2
- Il risultato è complementato
 - Se sono tutti 0 il pacchetto è accettato
 - Altrimenti è scartato



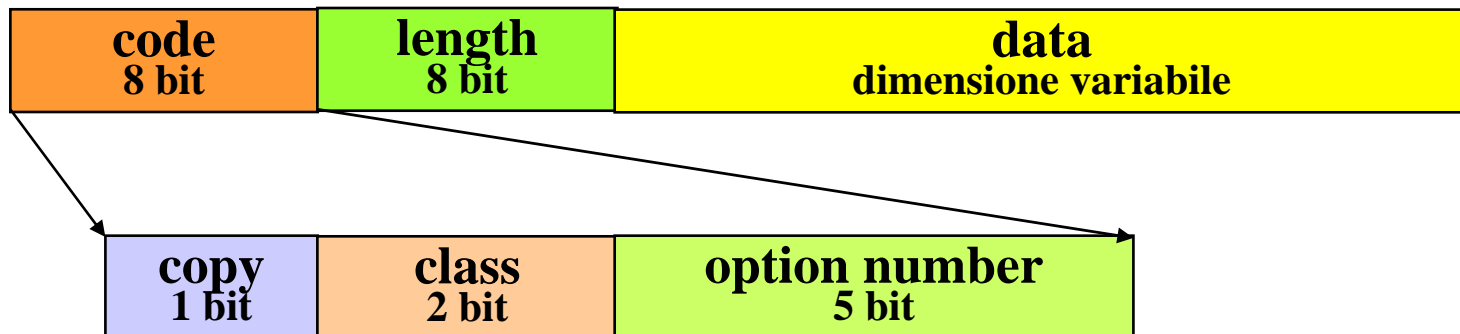
Le Opzioni

- La parte iniziale dell'header IP è di 20 byte ed è sempre presente
- Campi opzionali possono allungare l'header fino ad un massimo di 60 byte
- Opzioni usate per:

- *Testing*
- *Debugging*



Le Opzioni



Copy:

0 opzione copiata solo nel primo frammento
1 opzione copiata in tutti i frammenti

Class:

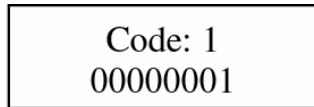
00 controllo del frammento
10 gestione e debugging

Option number:

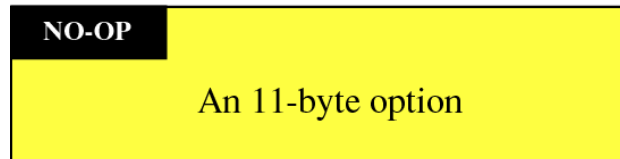
00000 end of option (1 byte)
00001 no operation (1 byte)
00011 loose source route
00100 time stamp
00111 record route
01001 strict source route

**Richiedono il
Campo dati**

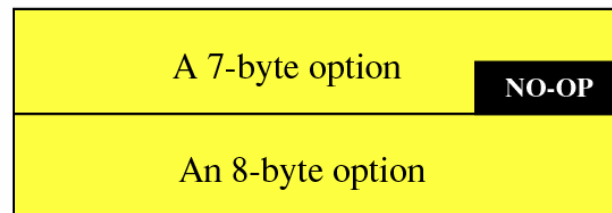
End Of Option e No operation



a. No operation option



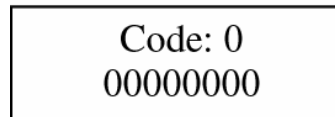
b. Used to align beginning of an option



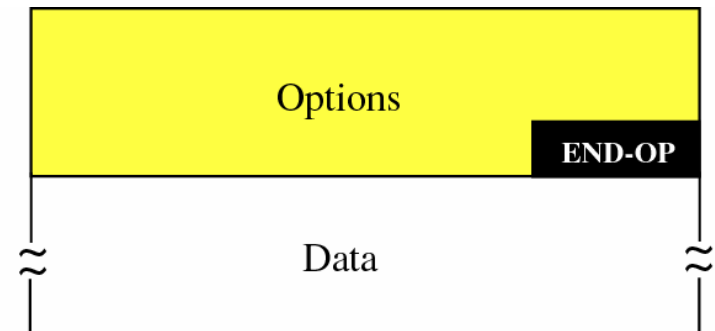
c. Used to align the next option

Source: *TCP/IP Protocol Suite*,
B. Forouzan.

- Sono campi opzione di 1 byte utilizzati per fare *padding*
- Non hanno la parte di dati

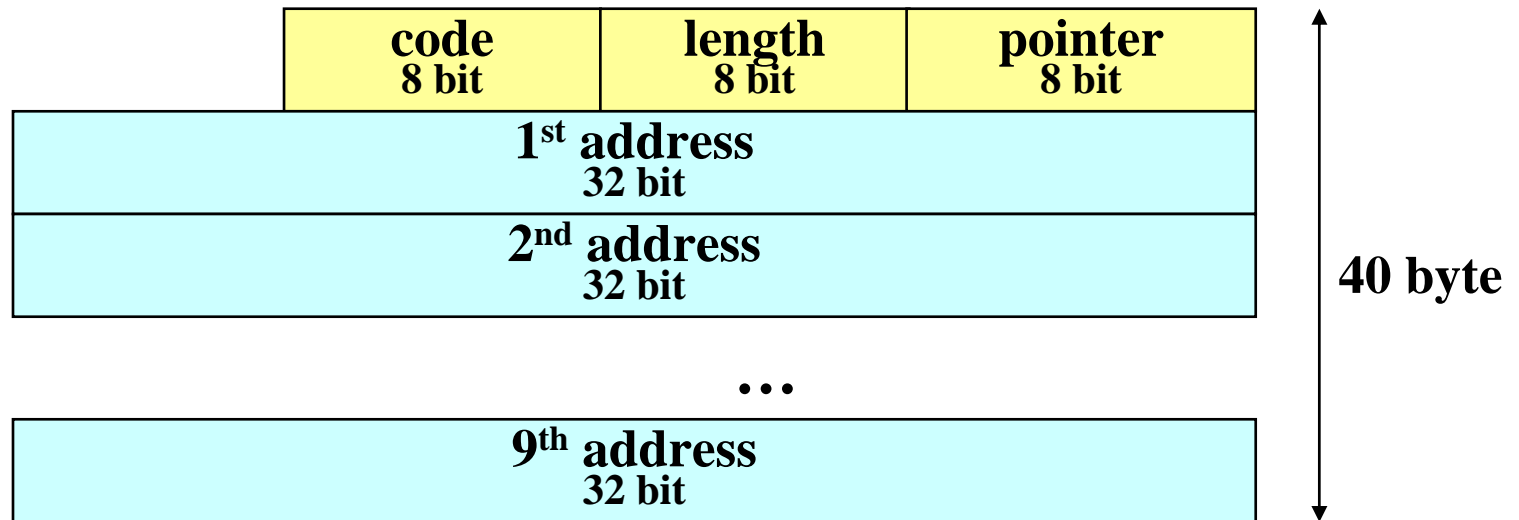


a. End of option



b. Used for padding

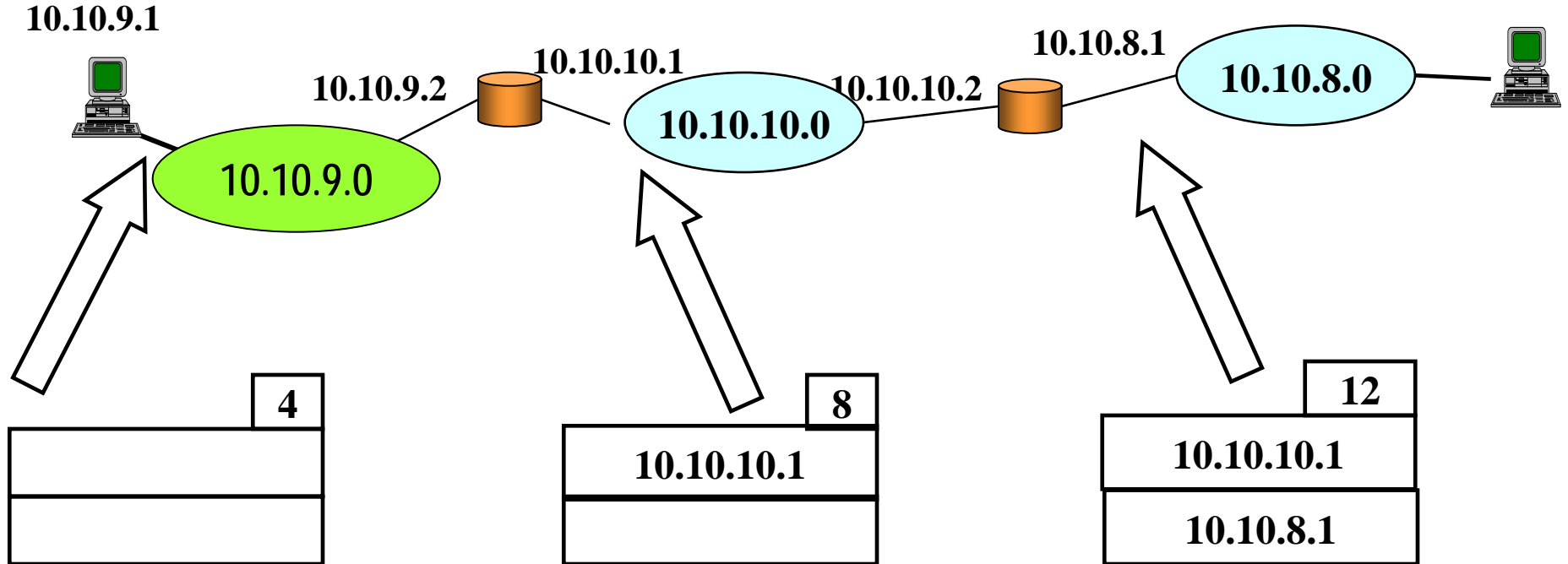
Record Route (1)



- Serve per registrare il percorso del pacchetto
- Il campo *pointer* punta al primo byte libero nella lista degli *address*
- Ogni volta che viene attraversato un router il suo indirizzo IP d'uscita viene registrato nel campo puntato e il *pointer* viene aumentato di 4

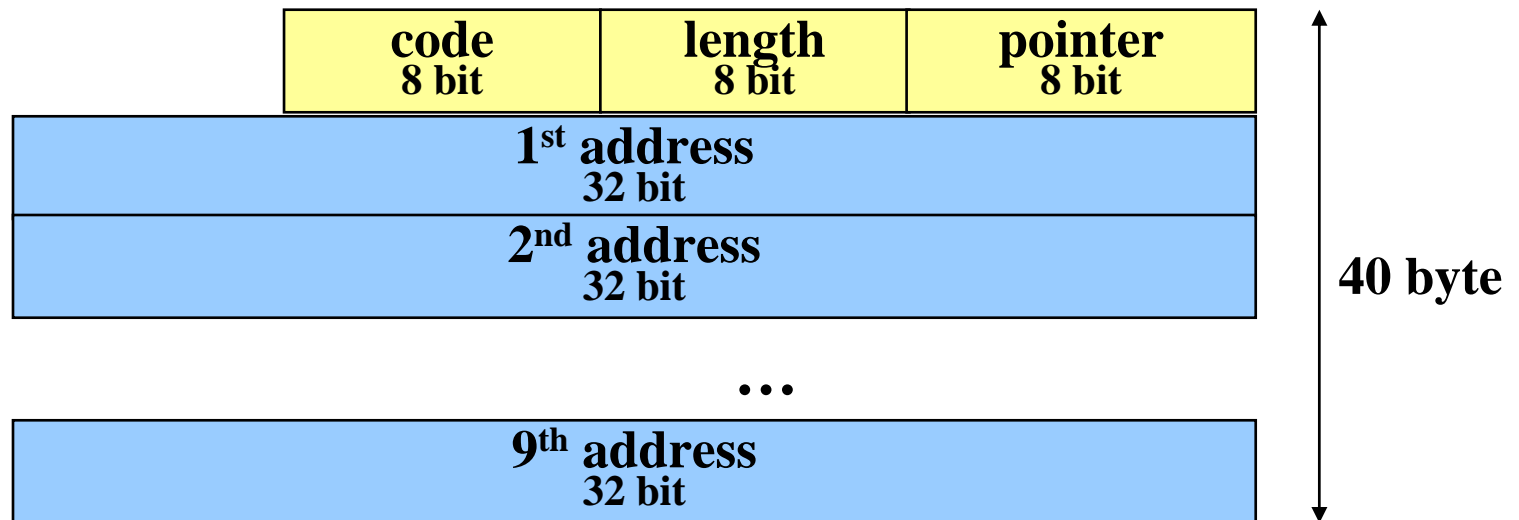
Record Route (2)

□ *Esempio:*



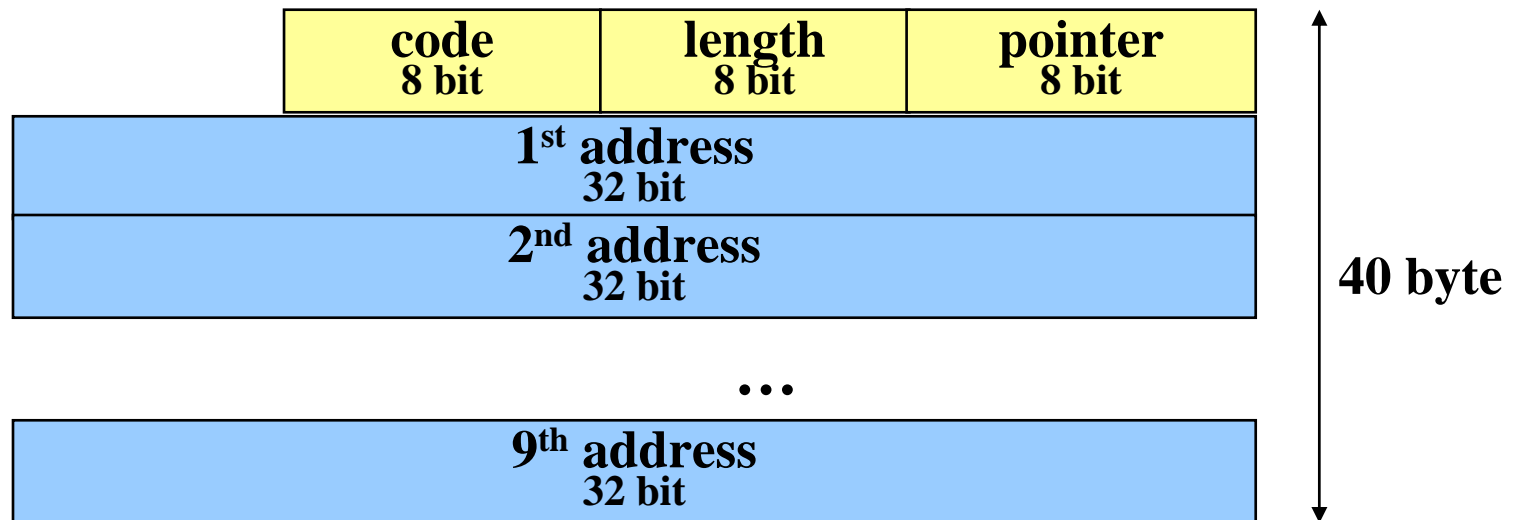
Strict Source Route

- ❑ Implementa un meccanismo di *source routing*
- ❑ Tutti i campi *address* sono inizialmente pieni e indicano i router che si vuole vengano attraversati
- ❑ il puntatore viene incrementato di 4 ad ogni *hop*
- ❑ se viene raggiunto un *router* non previsto il pacchetto viene scartato e viene generato un messaggio di errore
- ❑ (usata molto raramente!!!)



Loose Source Route

- come la precedente, ma è possibile visitare anche altri *router* (il pacchetto non viene scartato)
- (usata molto raramente!!!)



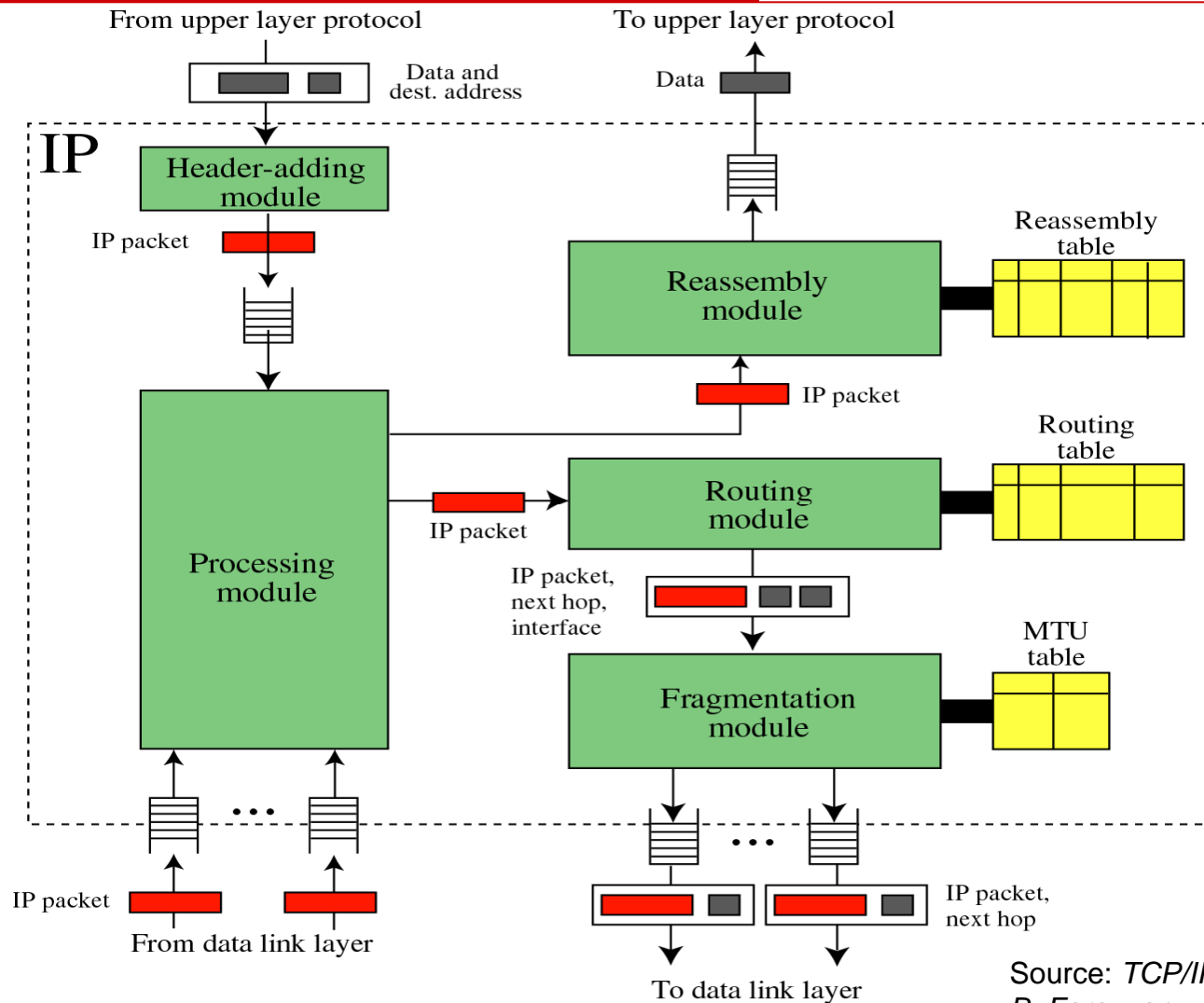
Time Stamp

code 8 bit	length 8 bit	pointer 8 bit	O-Flow 4 bit	Flag 4 bit
1 st address 32 bit				
1 st time stamp 32 bit				
2 nd address 32 bit				
2 nd time stamp 32 bit				

...

- ❑ misura il tempo assoluto di uscita del pacchetto da un router
- ❑ il campo *Over-Flow* indica i *router* sul percorso che non hanno potuto aggiungere il *timestamp*
- ❑ il campo *Flag* indica la modalità operativa stabilita dal mittente

Struttura Implementativa protocollo IP



Source: *TCP/IP Protocol Suite*,
B. Forouzan.



Politecnico di Milano

Advanced **N**etwork **T**echnologies **L**aboratory

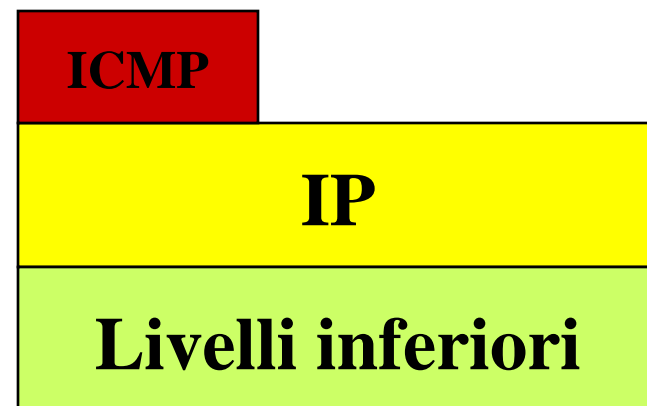
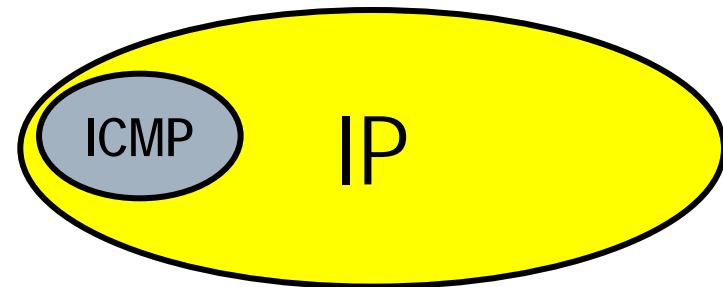


Internet Control Messge Protocol (ICMP)

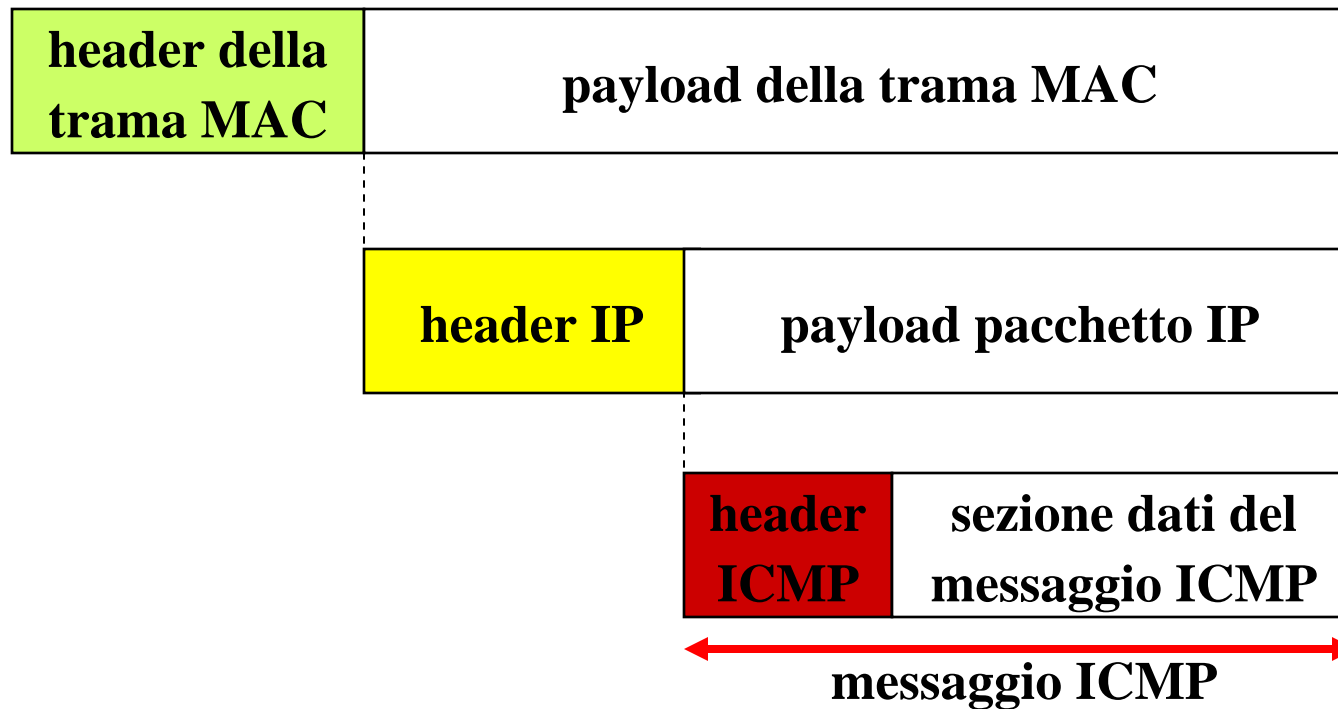
RFC 792

Internet Control Message Protocol (ICMP)

- E' un protocollo per messaggi di servizio fra host e router per informazioni su errori e fasi di attraversamento della rete
- da questo punto di vista può essere considerato come parte di IP
- i messaggi ICMP sono incapsulati e trasportati da IP, e quindi da questo punto di vista può essere considerato un utente di IP



Internet Control Message Protocol (ICMP)



- ❑ Nel pacchetto IP il campo *protocol* indica il codice dell'ICMP
- ❑ il messaggio ICMP viaggia all'interno del pacchetto IP

Formato messaggi ICMP



Type		Type	
0	Echo reply	11	Parameter problem
3	Destination unreachable	13	Timestamp request
4	Source Quench	14	Timestamp reply
5	Redirect (change a route)	17	Address mask request
8	Echo request	18	Address mask reply
11	Time exceeded		

Tipi di messaggi

□ Error Reporting

- *Destination Unreachable* (type 3)
- *Source Quench* (type 4)
- *Time Exceeded* (type 11)
- *Parameter Problem* (type 12)
- *Redirection* (type 5)

□ Query

- *Echo Request/Reply* (type 8,0)
- *Timestamp Request/Reply* (type 13/14)
- *Address Mask Request/Reply* (type 17/18)
- *Router Solicitation/Advertisement* (type 10/9)

Funzionalità di *Error Reporting*

- ❑ ICMP non corregge errori, ma si limita a segnalarli.
- ❑ L'evento errore è notificato alla sorgente del pacchetto IP che lo ha causato
- ❑ Eventi gestiti
 - *Destination Unreachable* (type 3)
 - *Source Quench* (type 4)
 - *Time Exceeded* (type 11)
 - *Parameter Problem* (type 12)
 - *Redirection* (type 5)
- ❑ I messaggi di errore contengono l'header del pacchetto IP che li ha generati e i suoi primi 8 byte di dati.

Destination Unreachable

type (3)	code (0-12)	checksum
non usato (0)		
header + primi 64 bit del pacchetto IP che ha causato il problema		

- ❑ Quando un router scarta un pacchetto per qualche motivo normalmente genera un messaggio di errore che invia alla sorgente del pacchetto
- ❑ nel campo *code* è codificato il motivo che ha causato l'errore
- ❑ ovviamente la generazione del messaggio avviene solo nei casi in cui il router può accorgersi del problema
- ❑ il motivo più comune è il fatto che la destinazione non è presente nella tabella di routing (*code* = 7)

Destination unreachable

type (3)	code (0-12)	checksum
non usato (0)		
header + primi 64 bit del pacchetto IP che ha causato il problema		

Alcuni Code:

- 0 network unreachable
- 1 host unreachable
- 2 protocol unreachable
- 3 port unreachable
- 4 fragmentation needed and DF set
- 5 source route failed
- ...

Time exceeded

type (11)	code (0-1)	checksum
non usato (0)		
header + primi 64 bit del pacchetto IP che ha causato il problema		

- Code 0 (inviato dai router)
 - Il messaggio di *time exceeded* viene usato quando il router decrementando il TTL lo pone a 0
 - il messaggio di *time exceeded* viene inviato alla sorgente del pacchetto
- Code 1 (inviato dalla destinazione)
 - viene usato dalla destinazione quando non tutti i frammenti di un pacchetto arrivano entro un tempo massimo

Parameter problem

type (12)	code (0-1)	checksum
pointer	non usato (0)	
header + primi 64 bit del pacchetto IP che ha causato il problema		

□ Code 0

- se l'header di un pacchetto IP ha una incongruenza in qualcuno dei suoi campi viene inviato il messaggio di *parameter problem*; il campo *pointer* punta al byte del pacchetto che ha causato il problema

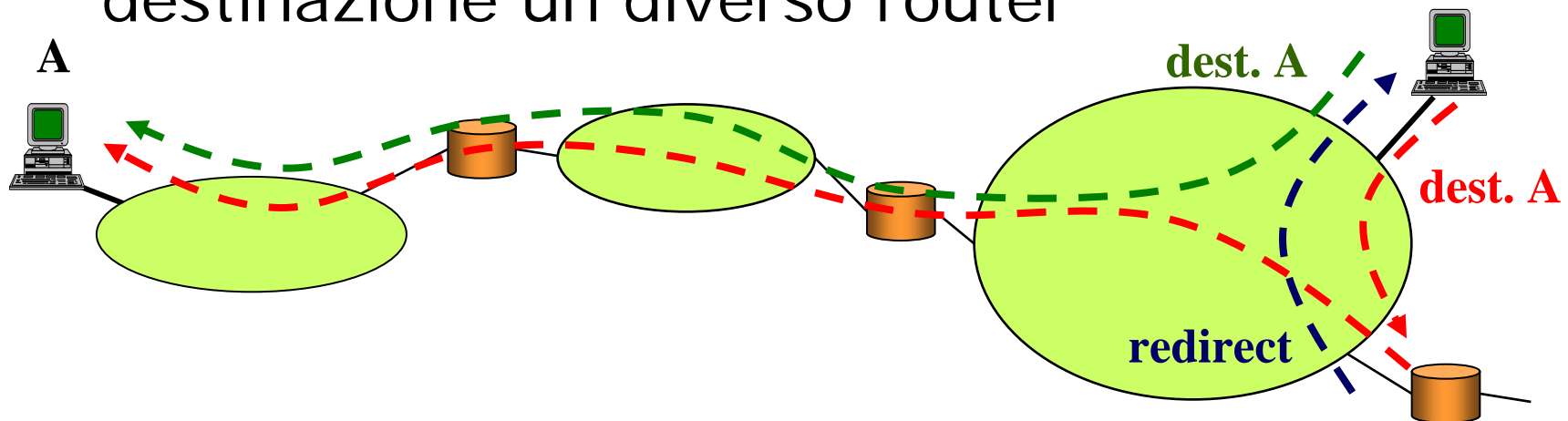
□ Code 1

- viene usato quando un'opzione non è implementata o qualche parte del campo opzioni manca

Redirect

type (5)	code (0-3)	checksum
indirizzo IP del router		
header + primi 64 bit del pacchetto IP		

- Questo messaggio viene usato quando si vuole che la sorgente usi per quella destinazione un diverso router

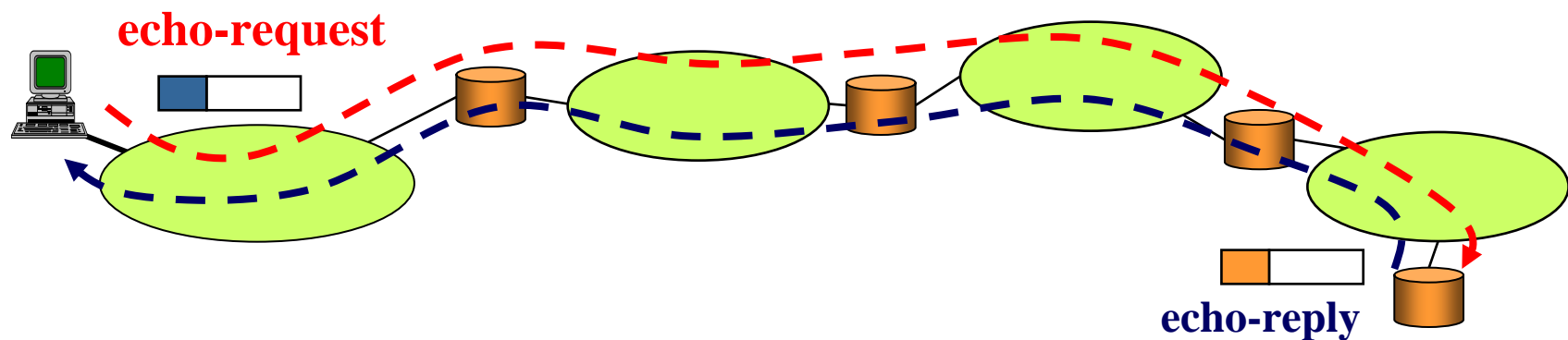


Funzionalità di diagnostica

- Coppie di messaggi secondo il paradigma domanda/risposta
- Tipi di messaggi:
 - *Echo Request/Reply* (type 8,0)
 - *Timestamp Request/Reply* (type 13/14)
 - *Address Mask Request/Reply* (type 17/18)
 - *Router Solicitation/Advertisement* (type 10/9)

Funzionalità di *Echo*

- I messaggi di *Echo-request* e *Echo-reply* sono usati per verificare la raggiungibilità e lo stato di un host o un router
- quando un nodo IP riceve un messaggio di *Echo-request* risponde immediatamente con un messaggio di *Echo reply*

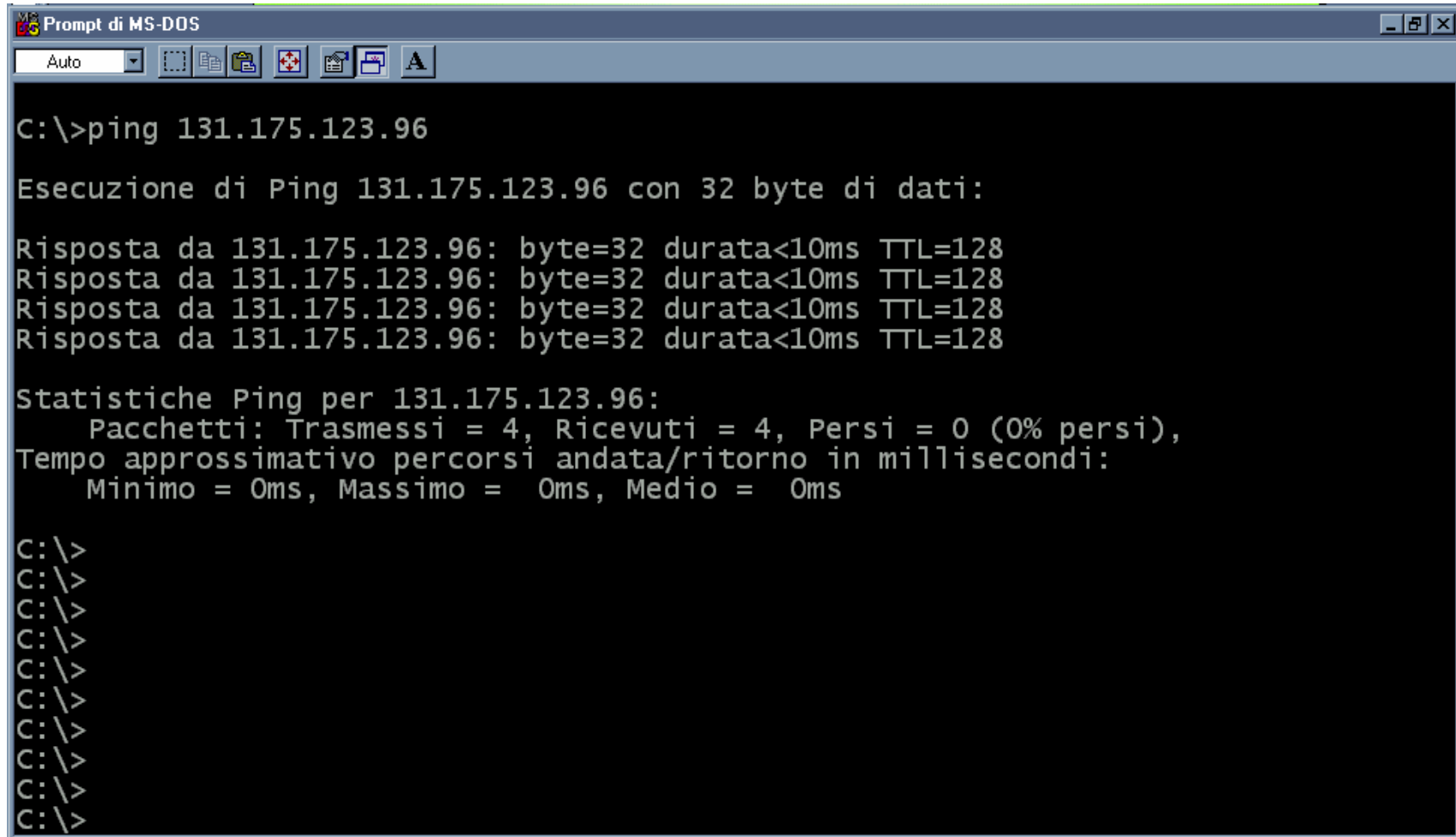


Messaggi *Echo*

type (8 request, 0 reply)	code (0)	checksum
identifier		sequence number
optional data		

- ❑ Il campo *identifier* viene scelto dal mittente della richiesta
- ❑ nella risposta viene ripetuto lo stesso *identifier* della richiesta
- ❑ più richieste consecutive possono avere lo stesso *identifier* e differire per il *sequence number*
- ❑ una sequenza arbitraria può essere aggiunta dal mittente nel campo optional data e deve essere riportata uguale nella risposta

Uso Messaggi di Echo: PING



```
Prompt di MS-DOS
Auto
C:\>ping 131.175.123.96

Esecuzione di Ping 131.175.123.96 con 32 byte di dati:

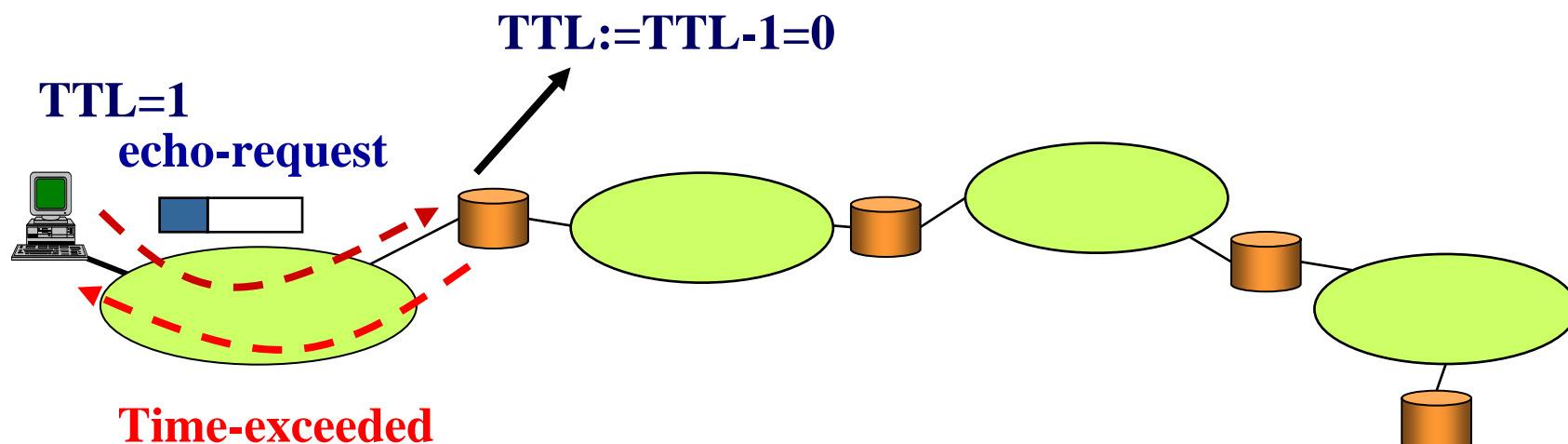
Risposta da 131.175.123.96: byte=32 durata<10ms TTL=128
Risposta da 131.175.123.96: byte=32 durata<10ms TTL=128
Risposta da 131.175.123.96: byte=32 durata<10ms TTL=128
Risposta da 131.175.123.96: byte=32 durata<10ms TTL=128

Statistiche Ping per 131.175.123.96:
  Pacchetti: Trasmessi = 4, Ricevuti = 4, Persi = 0 (0% persi),
Tempo approssimativo percorsi andata/ritorno in millisecondi:
  Minimo = 0ms, Massimo = 0ms, Medio = 0ms

C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
```

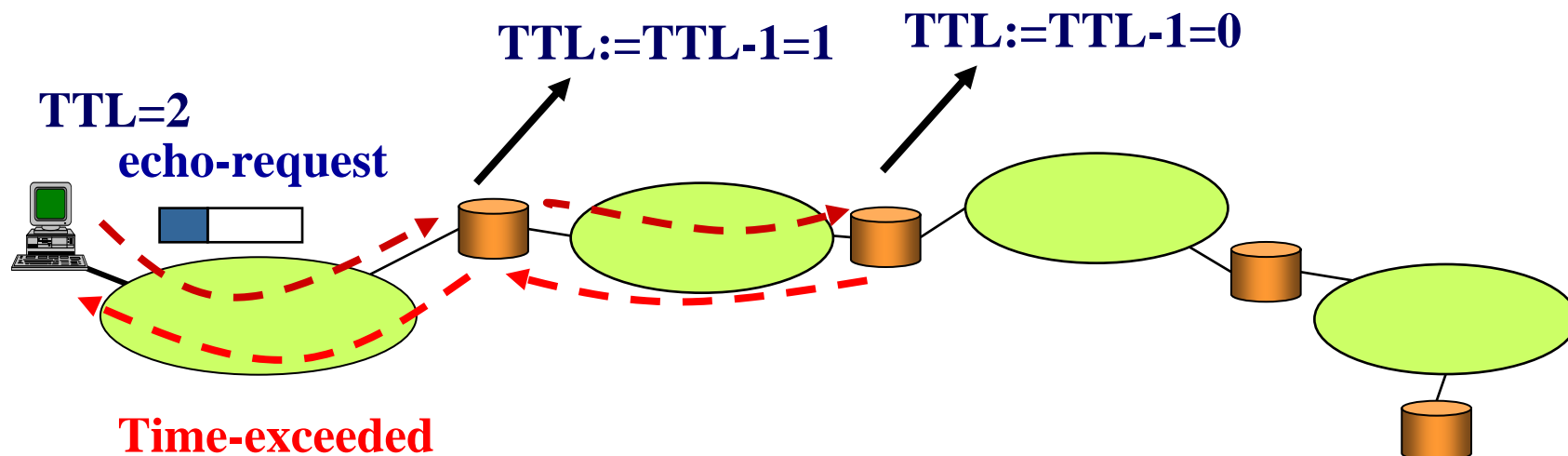

Traceroute: come funziona?

- Il *traceroute* usa (normalmente) messaggi di *Echo-request* verso la destinazione
- Il primo messaggio ha il $TTL=1$



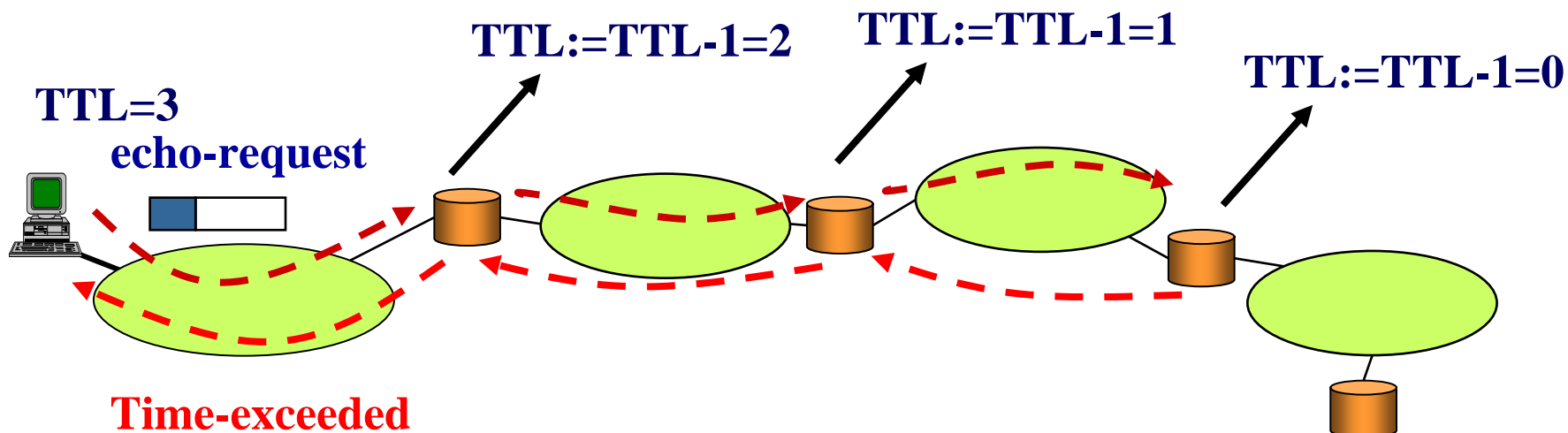
Traceroute: come funziona?

- I secondo messaggio ha il TTL=2



Traceroute: come funziona?

- I terzo messaggio ha il $TTL=3$, e così via ...



Timestamp request e reply

type (13 request, 14 reply)	code (0)	checksum
identifier		sequence number
originate timestamp		
receive timestamp		
transmit timestamp		

- ❑ Questo messaggio viene usato per lo scambio di informazioni sul clock di sorgente e destinazione
- ❑ *originate timestamp*: viene riempito dalla sorgente
- ❑ *receive timestamp*: viene riempito dalla destinazione appena ricevuto il pacchetto
- ❑ *transmit timestamp*: viene riempito dalla destinazione immediatamente prima di inviare il pacchetto di risposta

Address mask request e reply

type (17 request, 18 reply)	code (0)	checksum
identifier		sequence number
address mask		

- ❑ Questo messaggio viene usato per conoscere la netmask di un host/router
- ❑ Il campo *address mask* viene riempito dal destinatario