
La rete Internet e l'architettura di protocolli TCP/IP

Salvatore Marano

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
- **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

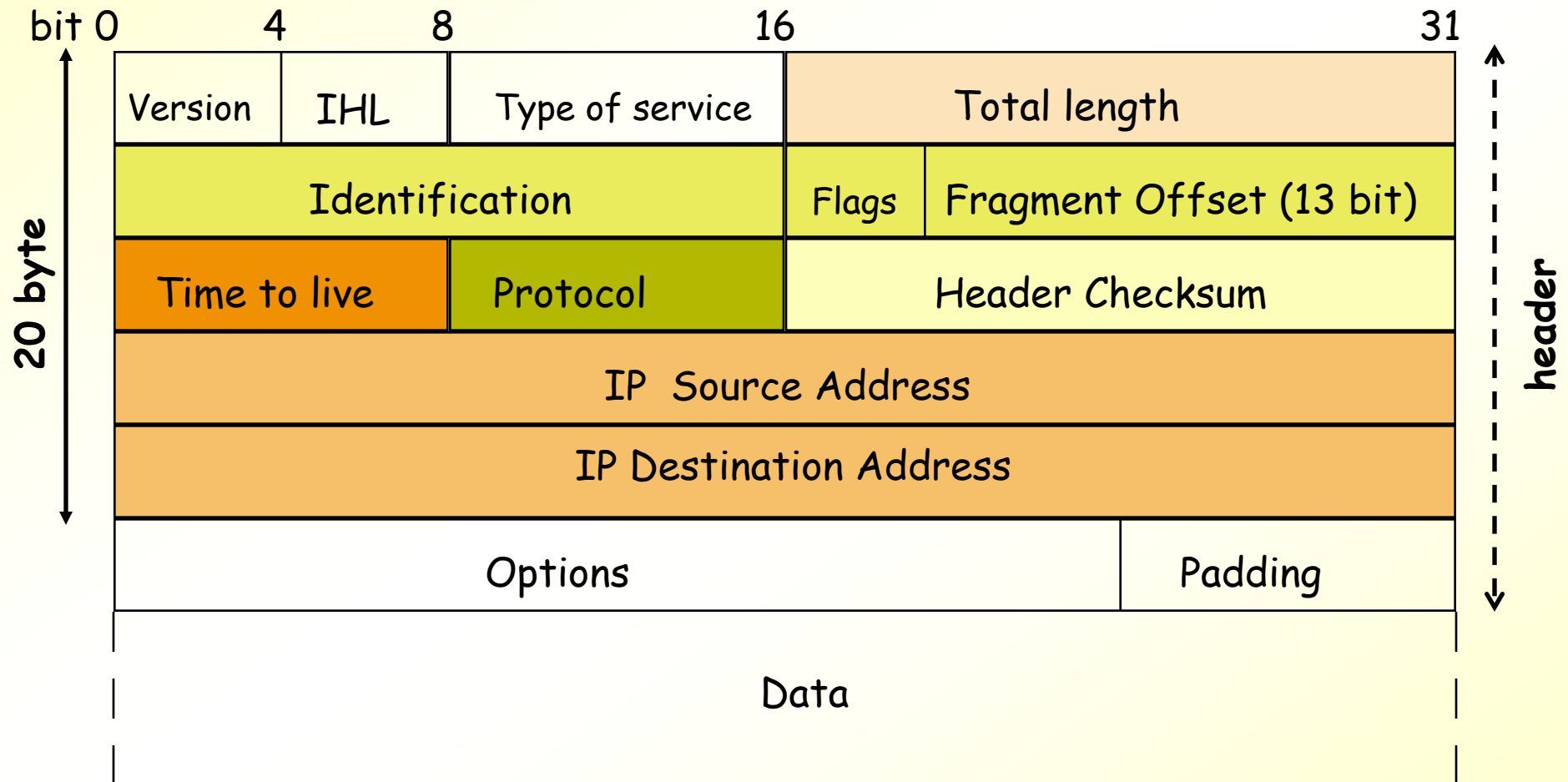
Formato dell'unità dati

- 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 di IHL è nel caso senza opzioni, cioè la lunghezza dell'header è 20 byte e il campo IHL è uguale a 5
- × **Type of Service (TOS) (8 bit):** specifica parametri della qualità di servizio richiesti dall'utente
 - nella versione originale di IP (RFC791), il campo TOS era formato da 2 sottocampi: un sub-campo di 3 bit (valori da 0 a 7) per specificare l'importanza del datagramma; il sub-campo successivo di 3 bit usato per richiedere particolari caratteristiche (se posto a 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 reputato 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
- × **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

Header IP

- × **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
- × **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

Frammentazione e aggregazione

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

Frammentazione e aggregazione

- Se il datagramma incontra una sottorete con una MTU di dimensione inferiore a quella scelta per il datagramma, quest'ultimo viene segmentato in più parti (frammenti)
 - × i frammenti non devono necessariamente essere tutti delle stesse dimensioni; almeno l'ultimo frammento ha dimensione minore degli altri
- Anche il frammento può essere frammentato a sua volta se incontra sottoreti con MTU ancora più piccole
- I soli vincoli imposti da IP sono
 - × i router devono accettare datagrammi di dimensioni pari a quelli delle MTU delle sottoreti interconnesse
 - × tutti i sistemi (host e router) devono comunque accettare datagrammi di dimensioni almeno pari a 576 byte (frammentati o non)
 - × I sistemi devono poter inoltrare datagrammi di 68 byte senza ulteriore frammentazione (60 byte di header al massimo + 8 byte di frammento, che è il minimo)

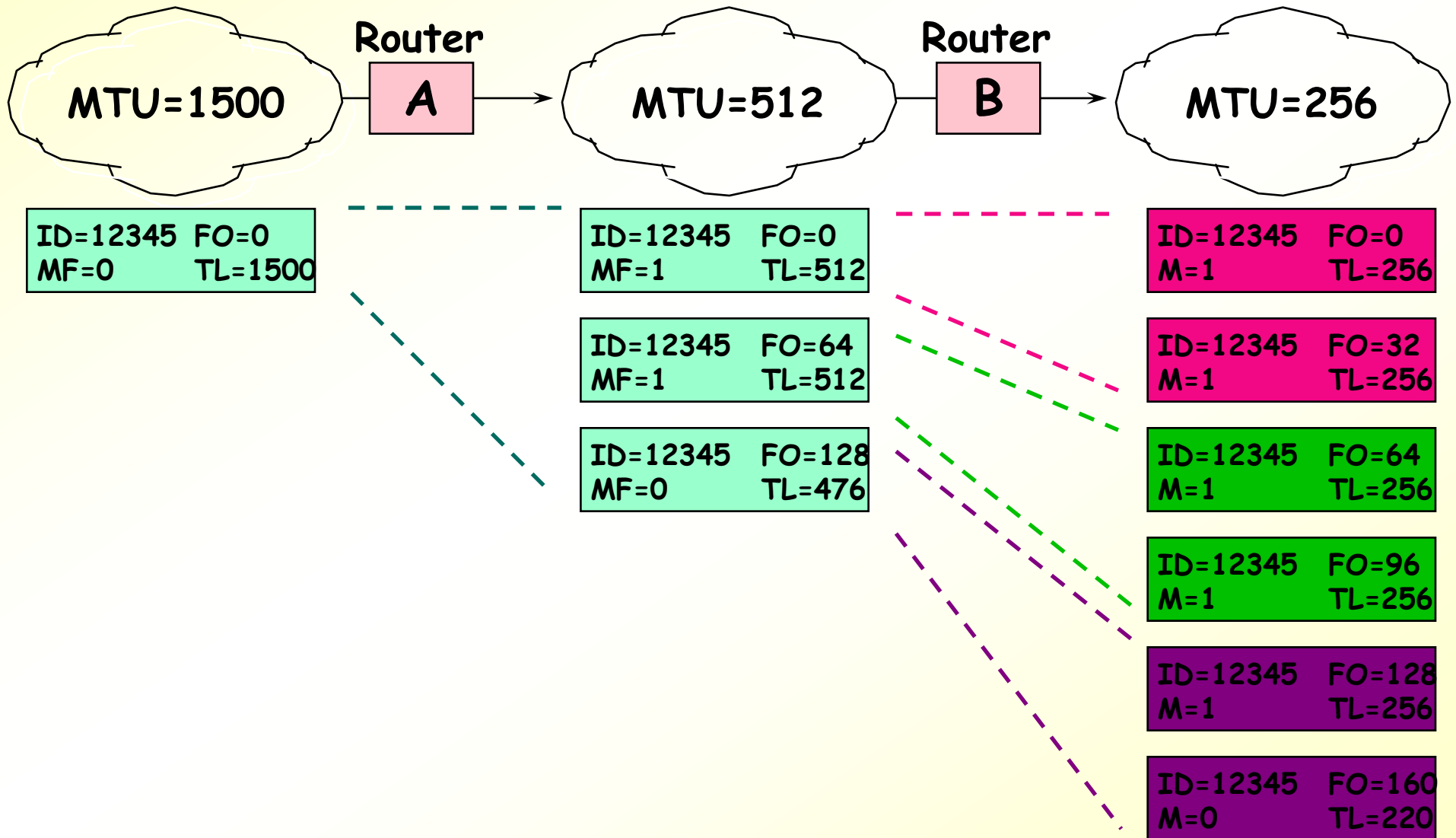
Frammentazione e aggregazione

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

Frammentazione e aggregazione

- L'informazione presente nell'intestazione del datagramma viene "copiata" nell'intestazione di ogni frammento; tranne il campo Total Length che viene modificato con la lunghezza del frammento in questione
 - × I campi Options dell'header possono venir copiati o non, o possono venir copiati solo nel primo frammento (es. RRO)
 - × ogni frammento diventa a sua volta un datagramma e può a sua volta essere ulteriormente frammentato
- Il datagramma originale viene ricostruito solo a destinazione
 - × la destinazione combina i frammenti con gli stessi valori dei campi Identification, Source e Destination Address e Protocol; la parte dati di ogni frammento è inserita nel datagramma nella posizione indicata dal campo Fragment Offset
 - × se uno o più frammenti vengono persi, i restanti che arrivano a destinazione vengono anch'essi scartati (dopo un time-out)
 - × se si utilizza TCP come protocollo di trasporto, l'intero datagramma viene ritrasmesso dalla sorgente

Frammentazione di datagrammi IP



Procedura di frammentazione

- Notazione del pseudo-codice: " \leq " significa "minore o uguale", "#" significa "diverso", "=" significa "uguale", " \neq " significa "assegnazione di un valore". Inoltre, "x to y" include x ed esclude y; per es. "4 to 7" include 4, 5, e 6 (non 7)
- - FO - Fragment Offset
 - IHL - Internet Header Length
 - DF - Don't Fragment flag
 - MF - More Fragments flag
 - TL - Total Length
 - OFO - Old Fragment Offset
 - OIHL - Old Internet Header Length
 - OMF - Old More Fragments flag
 - OTL - Old Total Length
 - NFB - Number of Fragment Blocks
 - MTU - Maximum Transmission Unit

Procedura di frammentazione

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

ELSE IF $DF = 1$ THEN discard the datagram ELSE

To produce the first fragment:

(1) Copy the original internet header;

(2) $OIHL \leftarrow IHL$; $OTL \leftarrow TL$; $OFO \leftarrow FO$; $OMF \leftarrow MF$;

(3) $NFB \leftarrow (MTU - IHL * 4) / 8$;

(4) Attach the first $NFB * 8$ data octets;

(5) Correct the header:

$MF \leftarrow 1$; $TL \leftarrow (IHL * 4) + (NFB * 8)$;

Recompute Checksum;

(6) Submit this fragment to the next step in datagram processing;

To produce the second fragment:

(7) Selectively copy the internet header (some options are not copied);

(8) Append the remaining data;

(9) Correct the header:

$IHL \leftarrow (((OIHL * 4) - (\text{length of options not copied})) + 3) / 4$;

$TL \leftarrow OTL - NFB * 8 - (OIHL - IHL) * 4$;

$FO \leftarrow OFO + NFB$; $MF \leftarrow OMF$; Recompute Checksum;

(10) Submit this fragment to the fragmentation test; DONE.

Procedura di aggregazione

- Notazione

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

Procedura di aggregazione

- ```

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

```

# Procedura di aggregazione

---

```
(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.
```