



Corso di
SISTEMI TELEMATICI
a.a. 2009-2010

Frammentazione e aggregazione nelle reti
INTERNET



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 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)

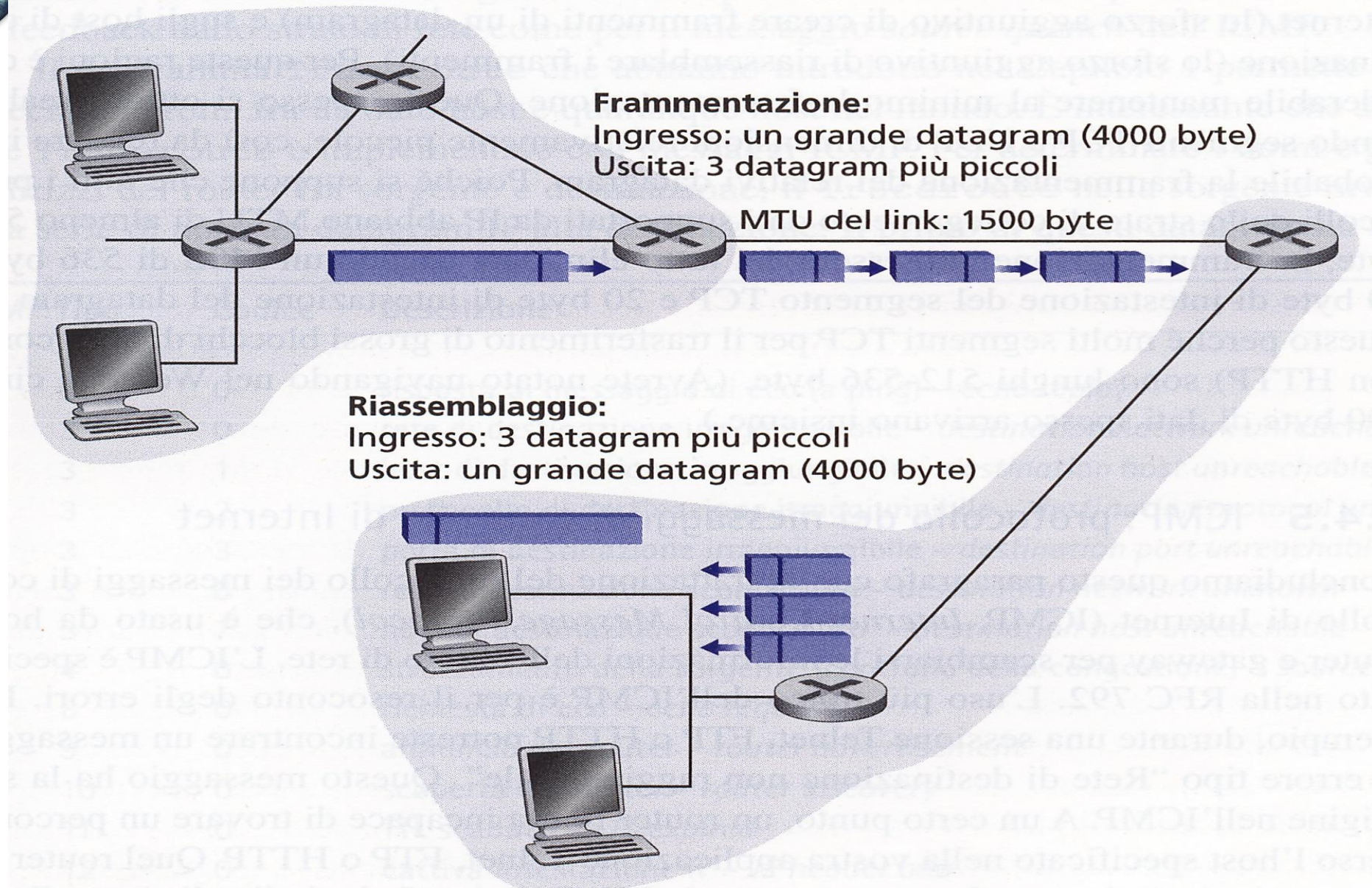
- ✗ i frammenti non devono necessariamente essere tutti delle stesse dimensioni; almeno l'ultimo frammento ha dimensione minore degli altri
- Anche il frammento può essere frammentato a sua volta se incontra sottoreti con MTU ancora più piccole



Frammentazione e aggregazione

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)



Esempio di frammentazione e riassemblaggio



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
- La segmentazione/aggregazione serve 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



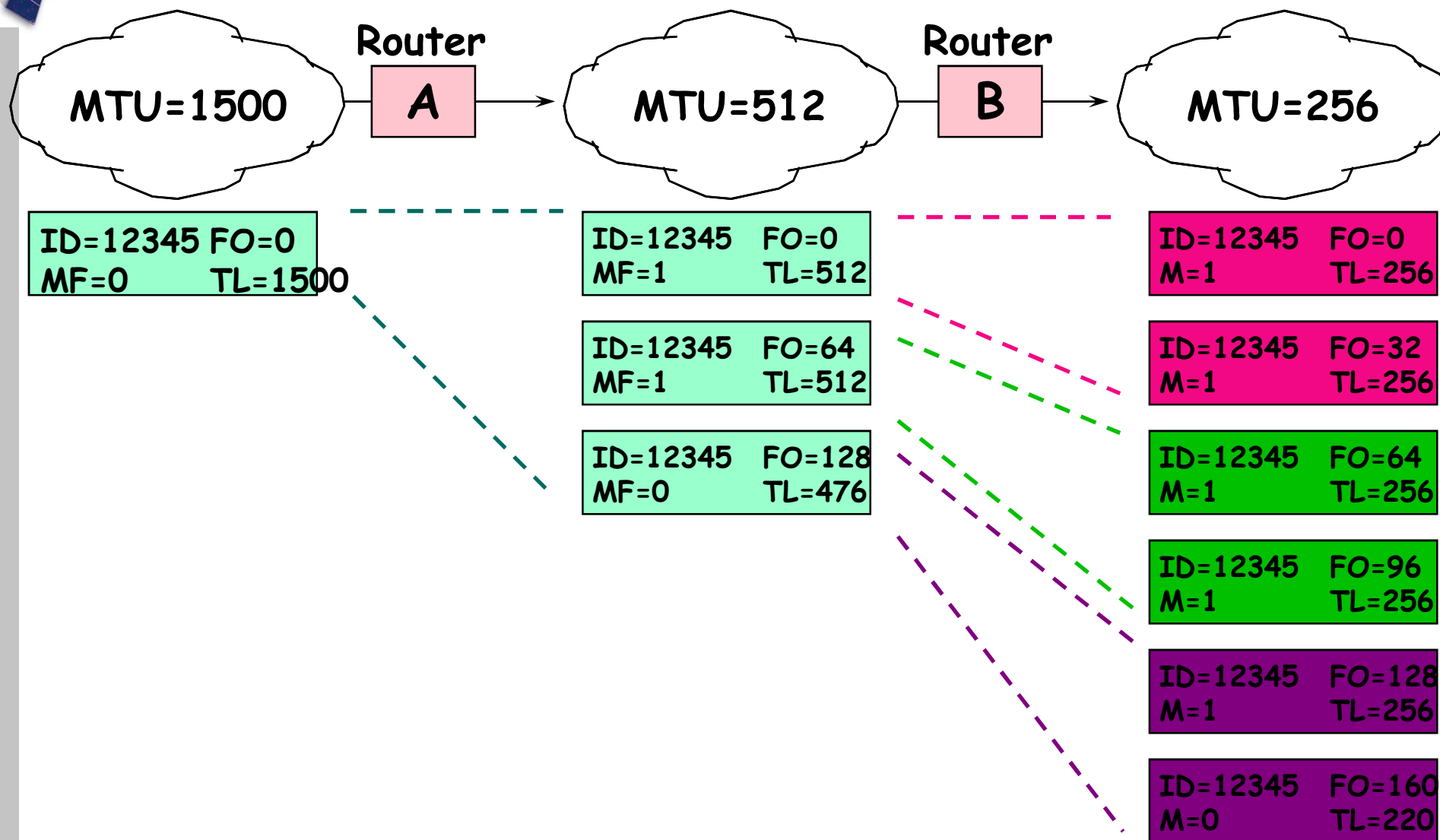
Frammentazione e aggregazione

• 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", " \leftarrow " 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 Total Length \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) OldIHL \leftarrow IHL; OldTL \leftarrow TL; OldFO \leftarrow FO;
OldMF \leftarrow MF;
- (3) Number of Fragment Blocks \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;



Procedura di frammentazione

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:
$$\text{IHL} \leftarrow (((\text{OIHL} * 4) - (\text{length of options not copied})) + 3) / 4;$$
$$\text{TL} \leftarrow \text{OTL} - \text{NFB} * 8 - (\text{OIHL} - \text{IHL}) * 4;$$
$$\text{FO} \leftarrow \text{OFO} + \text{NFB}; \text{MF} \leftarrow \text{OMF};$$

Recompute Checksum;
- (10) Submit this fragment to the fragmentation test;

DONE.



Procedura di aggregazione

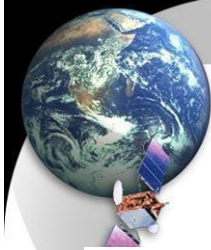
Notazione

- 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;



Procedura di aggregazione

- (9) set RCVBT bits from FO to $FO + ((TL - (IHL * 4) + 7) / 8)$;
- (10) IF MF = 0 THEN $TDL \leftarrow TL - (IHL * 4) + (FO * 8)$
- (11) IF FO = 0 THEN put header in header buffer
- (12) IF TDL \neq 0
- (13) AND all RCVBT bits from 0 to $(TDL + 7) / 8$ are set
- (14) THEN $TL \leftarrow TDL + (IHL * 4)$
- (15) Submit datagram to next step;
- (16) free all reassembly resources for this BUFID;
- DONE.
- (17) $TIMER \leftarrow \text{MAX}(TIMER, TTL)$;
- (18) give up until next fragment or timer expires;
- (19) timer expires: flush all reassembly with this BUFID;
- DONE.