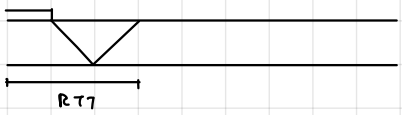
ESERCIZIO

$d = 3900\,Km$
$C = 2\,Mb/s$
$L = 1500\,B$
ACK trascurabili.

a) T per $F = 600\,KB$ + efficienza (S-a-W)
b) T per $F = 600\,KB$ + efficienza (G-B-N, $N = 30$)
c) valore minimo della finestra.
d) T nel caso che n 31 sia errato ($T_0 = 1\,s$, $N = 30$)
e) T nel caso che n [31; 40] siano errati ($T_0 = 1\,s$, $N = 30$)

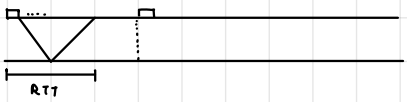
$t = \frac{d}{v} = \dots = 195\,ms$
$T = \frac{L}{C} = \dots = 6\,ms/s$
$RTT = T + 2\tau = 396\,ms$
$n = \frac{600 \cdot 10^3 B}{1500\,B} = 400$

a)



$T_{TOT} = n \cdot RTT = \dots = 158,4\,s$
$\eta = \frac{T}{RTT} \simeq 0,015$

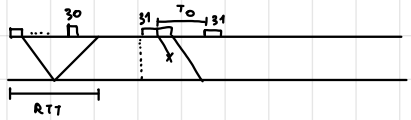
b)



? $N \cdot T < RTT \rightarrow$ sì
$m = \lceil \frac{n}{N} \rceil = \lceil 13,33 \rceil = 14$
$T_{TOT} = (m-1)RTT + 10T = 5,208\,s$

c) $N \geq \lceil \frac{RTT}{T} \rceil = W_{min} = 66 \rightarrow T_{TOT} = 400T + 2t = 2,79\,s$

d)



$T_{TOT} = (m-1)\,RTT + 10T + 2t + T_0 = \dots$

e) uguale a d)

4.2.2 CONTROLLO DI FLUSSO

Lo scopo del controllo di flusso è quello di non sovraccaricare il buffer di ricezione ed evitare la perdita di pacchetti. Nei protocolli tipo GBN la velocità di trasmissione dipende dalla dimensione N della finestra. Nel flow control rimane lo stesso concetto.

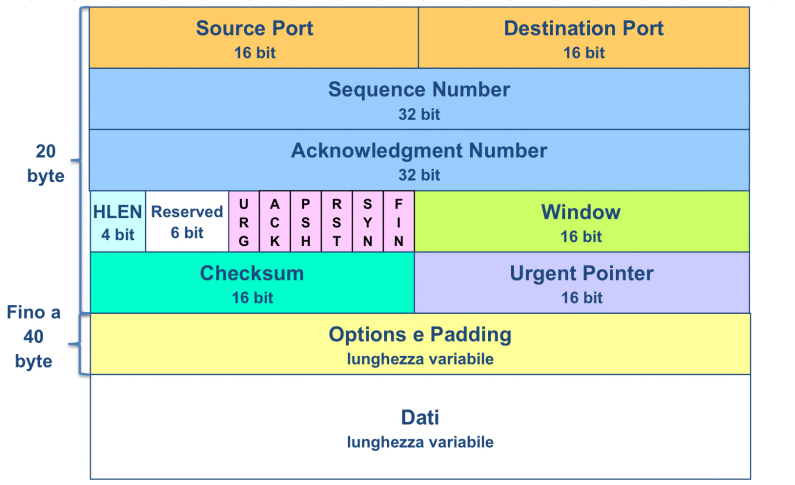
La sliding window viene spostata solo a riscontro inviato, ossia quando il pacchetto viene preso dalla coda dal livello superiore. Se il livello superiore è troppo lento, il timeout potrebbe scadere, causando inefficienza

Per risolvere i problemi della sliding window si usa il campo 'W' dell'header; il ricevitore lo usa nei riscontri per indicare quanti pacchetti può inviare prima della saturazione. Non è necessario dire la verità in W, in modo da mantenere un margine o altri motivi.

4.2.3 TCP

Il TCP è un protocollo a connessione, quindi necessita hand-shake, e full-duplex. Il TCP trasmette stream di byte che divide in segmenti per il trasporto. La lunghezza dei segmenti viene decisa dal protocollo per ragioni di performance.

Il controllo d'errore è in stile GBN. Il TCP numera i byte, quindi RN sarà il numero del primo byte del segmento e, di conseguenza, RN è il prossimo byte da ricevere. Le dimensioni delle finestre sono espresse in byte.



- **Source port, Destination port:** indirizzi di porta sorgente e porta destinazione di 16 bit
- **Sequence Number**: il numero di sequenza del primo byte nel payload
- **Acknowledge Number:** numero di sequenza del prossimo byte che si intende ricevere (numero valido solo se flag ACK valido)
- **HLEN (4 byte words):** contiene la lunghezza complessiva dell'header TCP, che DEVE essere un multiplo intero di 32 bit
- **Window:** contiene il valore della finestra di ricezione come comunicato dal ricevitore al trasmettitore
- **Checksum:** il medesimo di UDP, calcolato in maniera uguale

- Flags:
  - **URG**: vale 1 se vi sono dati urgenti e quindi il TCP deve passare in modalità urgente; in questo caso *urgent pointer* punta all'ultimo byte dei dati all'interno del flusso oltre il quale TCP può tornare in modalità normale
  - **ACK**: vale 1 se il pacchetto è un ACK valido; in questo caso *l'acknowledge number* contiene un numero valido
  - **PSH:** vale 1 quando il trasmettitore intende usare il comando di PUSH; il ricevitore può anche ignorare il comando (dipende dalle implementazioni)
  - **RST:** reset, resetta la connessione senza un *tear down* esplicito
  - **SYN:** *synchronize*; usato durante il *setup* per comunicare i numeri di sequenza iniziale
  - **FIN:** usato per la chiusura esplicita di una connessione
- **Options and Padding:** riempimento (fino a multipli di 32 bit) e campi opzionali, es., durante il setup per comunicare il MSS (il valore di default è 536 byte)

Il TCP offre opzioni. Alcune non fanno nulla, sono lunghe 1B e servono da padding per avere un header con lunghezza multipla di 32. Le opzioni che vedremo sono MSS e il fattore di scala della finestra.

1

- Definisce la dimensione massima del segmento che verrà usata nella connessione TCP
- La dimensione è decisa dal mittente durante la fase di *setup*
- Valore di *default* è 536 byte, il valore massimo 65535 byte

Code (00000010)	Length (00000100)	MSS 16 bit

2

- Definisce il fattore di scala della finestra
- Il valore di *default* è 1
- L'opzione fa sì che venga moltiplicato il valore del campo *Window* di un fattore pari a 2 elevato al valore contenuto nel campo *fattore di scala*

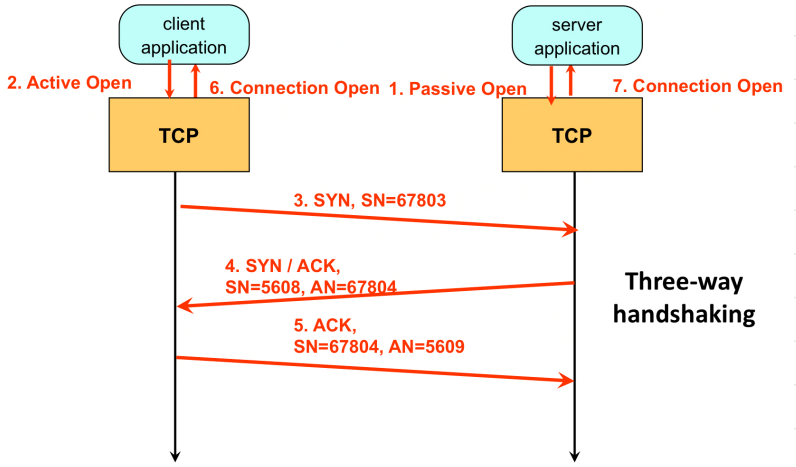
Code (00000010)	Length (00000011)	Fattore di scala 8 bit

4.2.3.1 SETUP

Prima del 'call setup' il lato client e quello server si preparano:

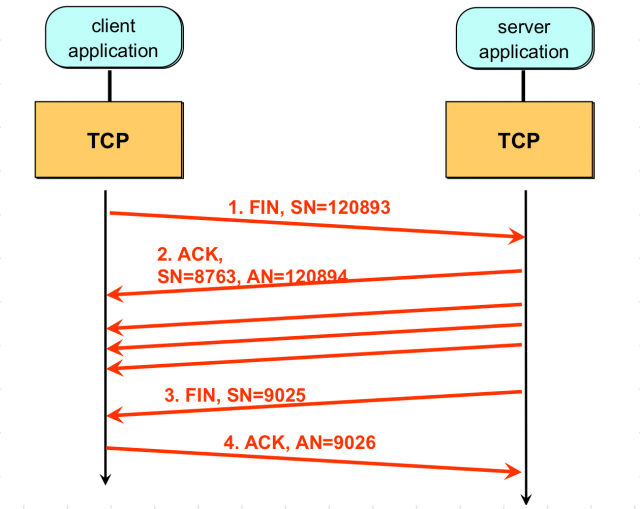
1. PASSIVE OPEN: il server comunica alla TCP stack che è in ascolto
2. ACTIVE OPEN: il client comunica alla TCP stack che è pronto a creare una connessione

Il client inizia la comunicazione con un messaggio di SYN dove comunica l'inizio della sequenza (e eventuali opzioni). Il server risponde con un SYN/ACK dove ha come SN = 67804 per ricordo. Il client risponde con un ultimo ACK che ha SN = 67804 e AN la prima sequenza. Le rispettive stack notificano le applicazioni del fatto che la connessione è stata stabilita. Questo è detto 3-way handshake.



4.2.3.2 TEAR-DOWN

Il modo corretto per chiudere la connessione è l'invio di un segmento con flag FIN, seguito da un ACK di conferma dall'altro lato. La connessione rimane aperta dall'altra parte per permettere l'invio degli ultimi segmenti. Quando anche l'altra parte manda un segmento FIN e l'altra risponde con ACK la connessione è del tutto chiusa.



Un altro modo è quello di utilizzare il flag RST dell'header.