# Esercizi – Transport Layer

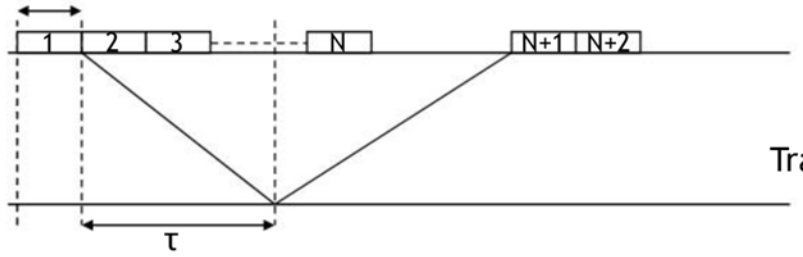
16/11/2022

# Ex. n. 1

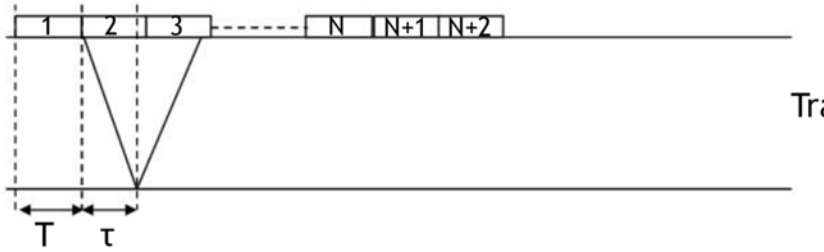
Si consideri un canale via satellite della capacità di 1 Mb/s. Considerando che il tempo di propagazione attraverso un satellite geostazionario richiede 250 ms, si chiede di dimensionare la minima finestra di trasmissione di un protocollo Go-BACK-N (con time-out) in modo che sia consentita la massima efficienza temporale del canale quando vengano trasmesse dei frame di 2000 bit. Si suppongano gli ACK trascurabili. Si calcoli poi la massima efficienza trasmissiva che si avrebbe nel caso in cui il meccanismo fosse di tipo STOP and WAIT semplice.

# Soluzione ex 1

L'efficienza del meccanismo Go-BACK-N dipende dal rapporto tra RTT e lunghezza della finestra.




Trasmissione Inefficiente
RTT>NT




Trasmissione Continua
RTT<NT

Per avere la massima efficienza il numero di pacchetti N nella finestra deve essere tale che il loro tempo di trasmissione copra il tempo di andata e ritorno del primo pacchetto.
Indicati con:

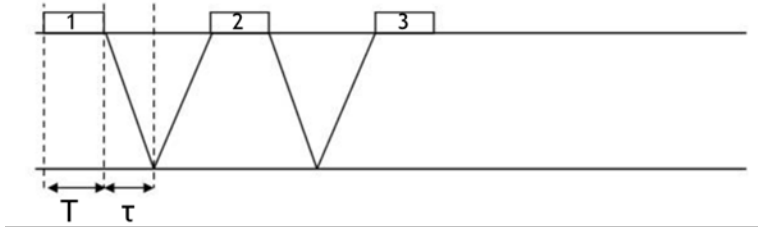
- $T = 2\,[ms] = 2000\,[bit] / 1\,[Mb/s]$, il tempo di trasmissione di un pacchetto
- $\tau = 250\,[ms]$, il tempo di propagazione

allora deve essere:

$$NT \geq T + 2\tau \rightarrow N \geq 1 + 2\frac{\tau}{T} = 1 + 2 \cdot 250/2 = 251$$

Per finestre $N \geq 251$, la trasmissione risulta continua, dunque l'efficienza del meccanismo è 1.

Nel caso di meccanismo STOP&WAIT abbiamo:




L'efficienza del meccanismo STOP&WAIT si calcola considerando che il meccanismo trasmette 1 pacchetto (durata $T = 2\,ms$) ogni $T + 2\tau$, dunque l'efficienza: $\eta = T/(T + 2\tau) = 1/251$.

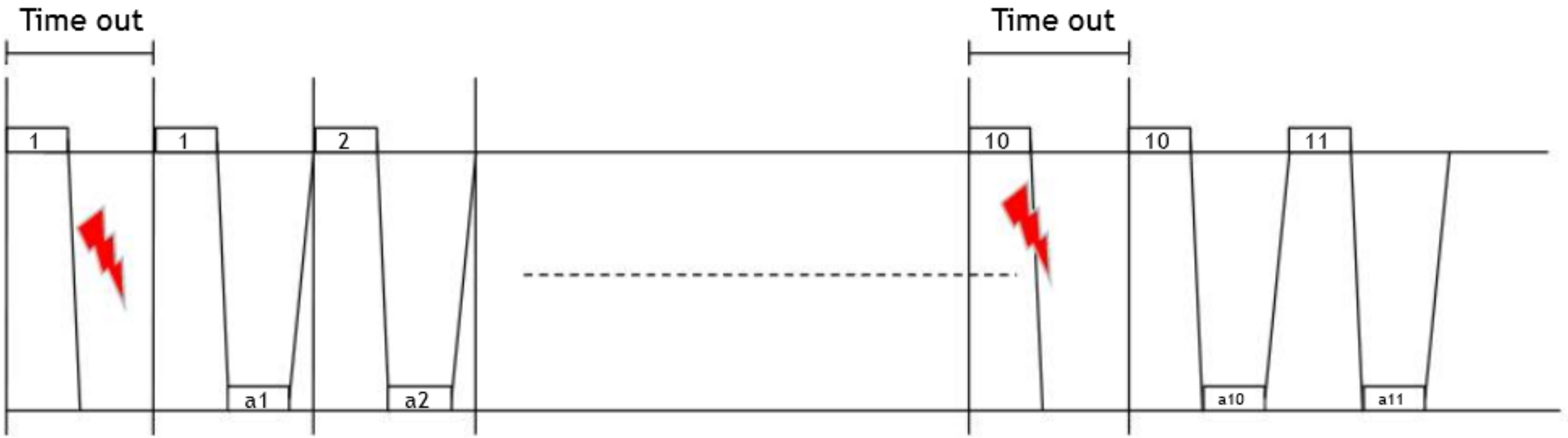
# Ex. n. 2

Un canale genera errori nella trasmissione di pacchetti in ragione di *1* ogni *10,* mentre non introduce mai negli ACK di ritorno. Si calcoli l'efficienza del meccanismo in termini di *numero di pacchetti corretti / numero totale pacchetti trasmessi* nel caso in cui si usi STOP and WAIT con time-out minimo.

Si calcoli poi l'efficienza trasmissiva temporale totale (*tempo usato per trasmettere pacchetti corretti/tempo totale*) nel caso in cui il tempo di propagazione sia pari a *n* volte il tempo di trasmissione di un pacchetto *T* e il tempo di trasmissione dell'ACK sia pari a *T*.

# Soluzione ex 2

Con lo STOP and WAIT, ogni 10 pacchetti, 1 è errato. Dunque l'efficienza è $\eta_{pkt} = 9/10 = 0.9$

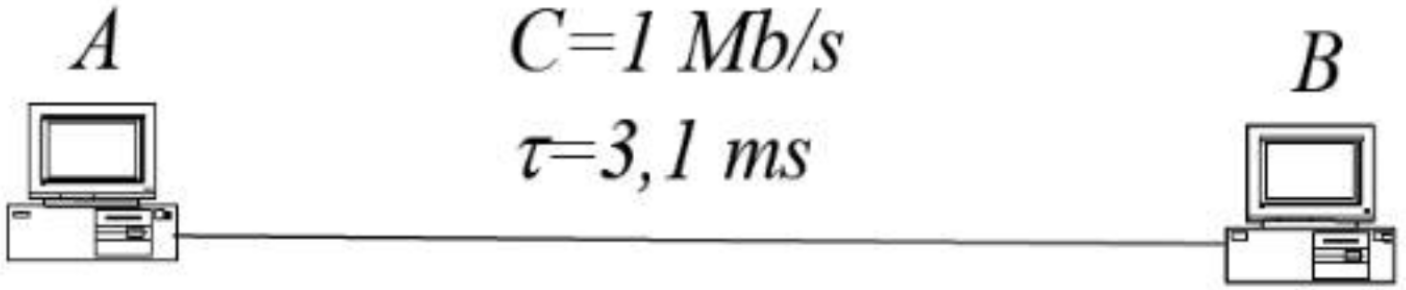



Per quanto visto prima si trasmettono 9 pacchetti corretti (durata 9T) ogni 10 round trip time. Un round trip time è pari a $RTT = T + T_{ack} + 2\tau = T + T + 2nT$. Dunque: $\eta_{tot} = 9T / 10T(2 + 2n) = 9 / 20(1 + 1n)$

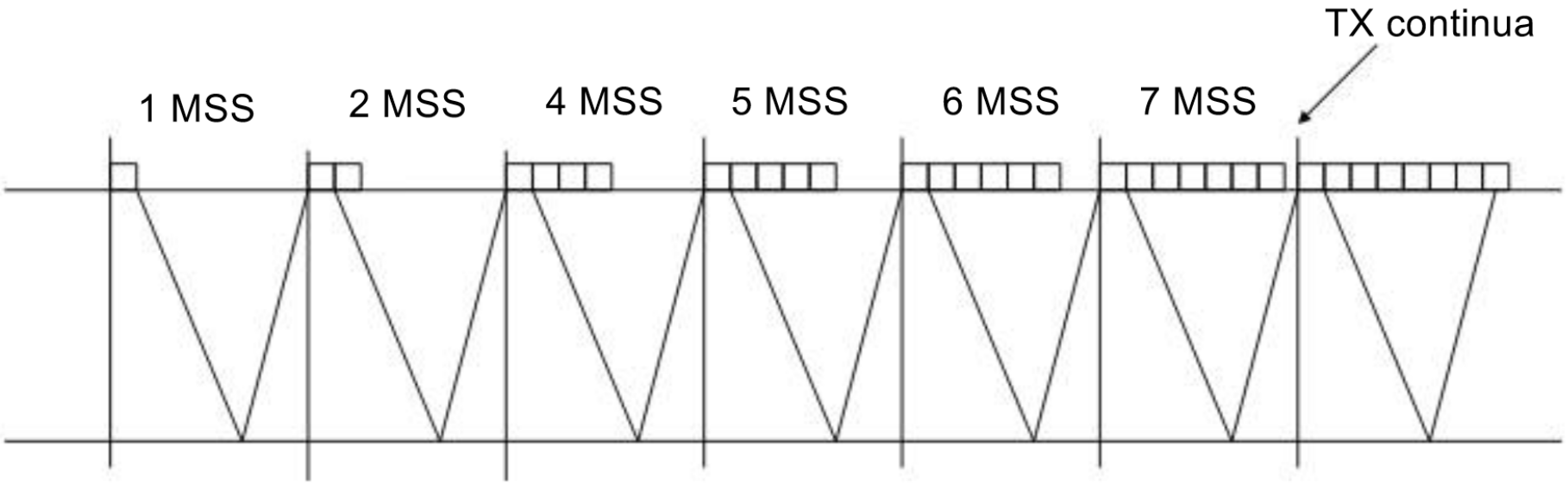
# Ex. n. 3

Si consideri il collegamento in figura tra i due host A e B. A deve trasferire una sequenza di *100* segmenti di lunghezza massima usando TCP. Si calcoli il tempo necessario supponendo:

- MSS = *1000 [bit]*
- lunghezza degli header di tutti i livelli trascurabile
- la connessione venga aperta da A e la lunghezza dei segmenti di apertura della connessione sia trascurabile
- la lunghezza degli ACK sia trascurabile
- SSTHRESH sia pari a *5* MSS



# Soluzione ex 3




Il tempo di $trasmissione\ T = 1000\ [bit]\ /\ 1\ [Mb/s]\ =\ 1\ [ms]$, mentre $RTT\ =\ 6.2\ [ms] + \ T\ =\ 7.2\ [ms]$

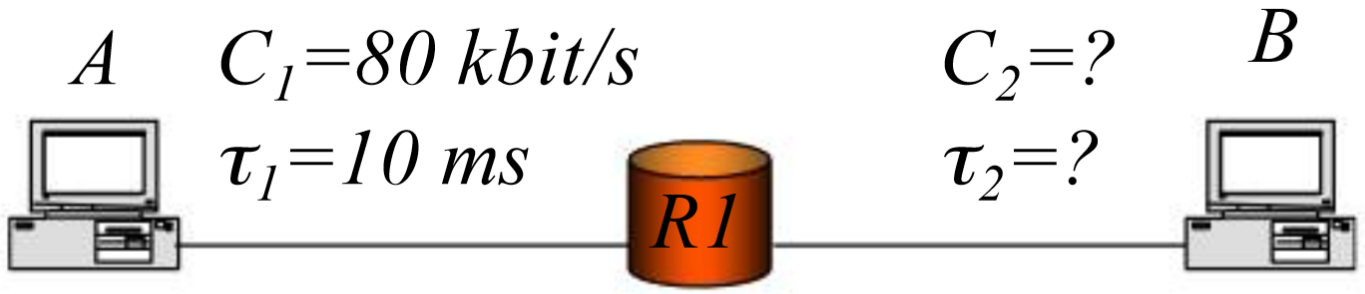
La trasmissione è dunque discontinua fino a che WT < RTT, cioè fino a che W = 8.

Il tempo totale di trasferimento è pari a:

$$2\ \tau\ (setup\ connessione) +\ 6\ RTT\ (Primi\ 25\ MSS)$$
$$+\ 75\ T\ (75\ MSS\ in\ trasmissione\ continua)$$
$$+\ 2\ \tau\ (ritorno\ ACK\ dell'ultimo\ MSS) =\ 130.6\ [ms]$$

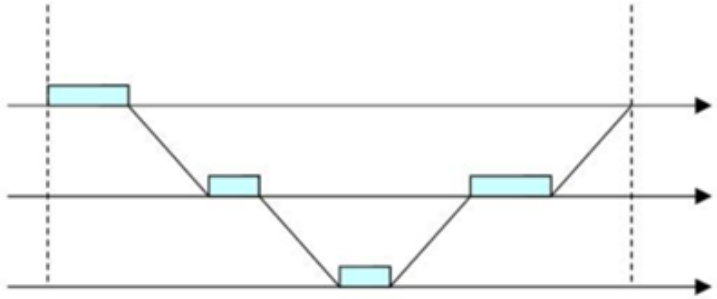
# Ex. n. 4

Si consideri il collegamento in figura




A vuole conoscere la capacità e il ritardo di propagazione del link 2 e allo scopo invia a B due messaggi di echo: $M_1$ di lunghezza $l_1 = 1000$ [*byte*], ed $M_2$ di lunghezza $l_2 = 1500$ [*byte*]; per ognuno di essi misura un Round-Trip-Time (RTT) pari a *780 [ms]* e *1130 [m]s*, rispettivamente. Nella risposta, B utilizza messaggi con le stesse lunghezze. Calcolare $C_2$ e $\tau_2$ nell'ipotesi che le lunghezze degli header siano trascurabili.

# Soluzione ex 4



Secondo lo scambio in figura, impostiamo un sistema di due equazioni (una per pacchetto) in due incognite ($C_2$ e $\tau_2$)

$$RTT_1 = 2\left(\frac{m_1}{C_1} + \tau_1 + \frac{m_1}{C_2} + \tau_2\right)$$

$$RTT_2 = 2\left(\frac{m_2}{C_1} + \tau_1 + \frac{m_2}{C_2} + \tau_2\right)$$

Inserendo i valori numerici abbiamo

$$780 = 2\left(\frac{8000}{80} + 10 + \frac{8000}{C_2} + \tau_2\right) = 220 + \frac{16000}{C_2} + 2\tau_2$$

$$1130 = 2\left(\frac{12000}{80} + 10 + \frac{12000}{C_2} + \tau_2\right) = 320 + \frac{24000}{C_2} + 2\tau_2$$

E risolvendo

$$\tau_2 = 280 - \frac{8000}{C_2}$$

$$810 = \frac{24000}{C_2} + 2\left(280 - \frac{8000}{C_2}\right) = \frac{8000}{C_2} + 560;$$

$$C_2 = \frac{8000}{250} = 32\ kbit/s$$

$$\tau_2 = 280 - \frac{8000}{32} = 30\ ms$$

# Ex. n. 5

Una connessione TCP è usata per trasmettere un file da *39.5 [kbyte]* utilizzando i seguenti parametri:

- MSS = *500 [byte]*
- RTT = *500 [ms]*
- timeout pari a 2 RTT.

Si assuma che le condizioni iniziali delle finestre siano:

- RCWND = *12 [kbyte]*
- SSTHRESH = *8 [kbyte]*
- CWND = *500 [byte]*

E che inoltre:

- si verifichi un errore sulla connessione all'istante *3 s* (tutti i segmenti in trasmissione vengano persi)
- al tempo *4.5 [s]* il ricevitore segnali RCWND = *2 [kbyte]*

Si tracci l'andamento nel tempo di:

- CWND
- SSTHRESH
- RCWND

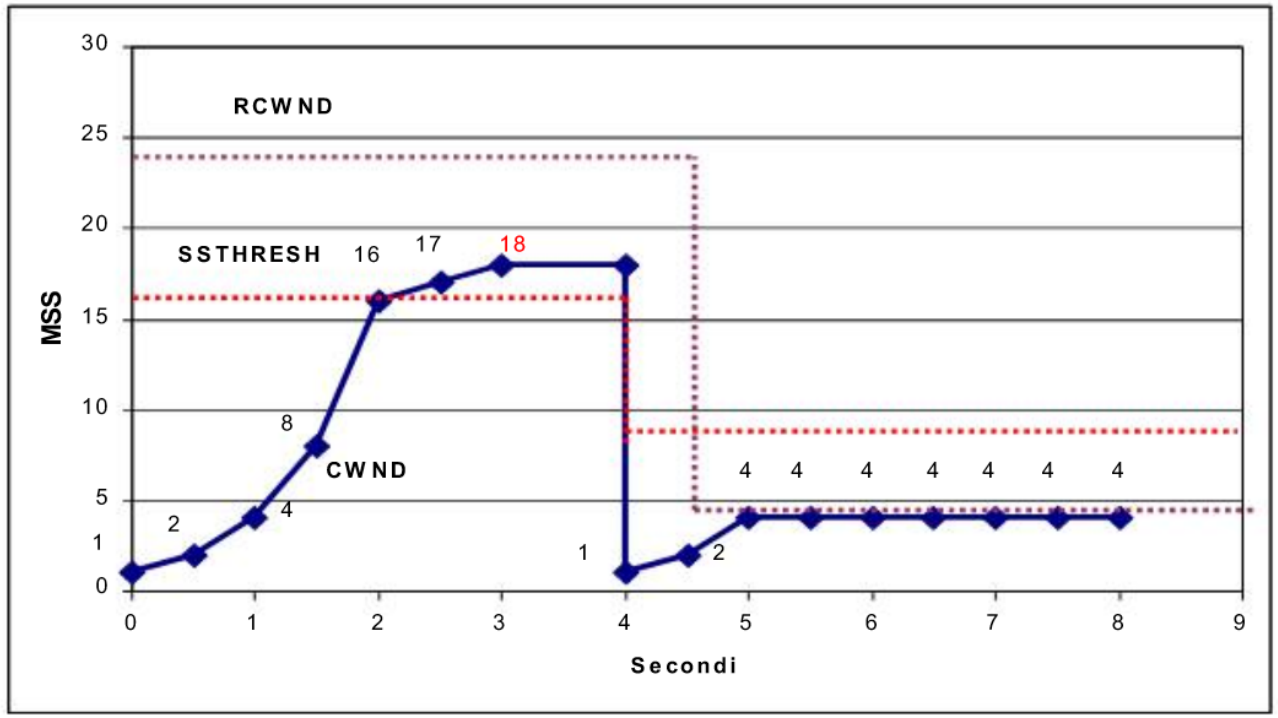
Si calcoli il tempo di trasmissione del file utilizzando multipli di RTT come base temporale

# Soluzione ex 5

Conviene ragionare in numero di segmenti trasmessi
Dimensione File (in MSS) = 39.5 $[Kbyte]$ / 500 $[byte]$ = 79 $MSS$

Dobbiamo trovare il tempo necessario per trasferire 79 MSS, possiamo farlo utilizzando il seguente grafico

$RCWND = 12\ [Kbyte] / 500\ [byte] = 24\ MSS$
$SSTHRESH = 8\ [Kbyte] / 500\ [byte] = 16\ MSS$
$Timeout = 1\ [s]$




Il tempo di trasferimento del file è T = 8.5 [s], alle fine dell'RTT che inizia a 8 [s].

# Ex. n. 6

Si calcoli il checksum secondo la modalità del protocollo UDP della seguente sequenza di bit:

**1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 1 1 0 1 0 1 0 1 0 1 0 1 0 1 1 0 0 0 0 1 0 0 0**
**1 0 0 0 1 0 1**

# Soluzione ex 6

Il primo passo consiste nello spezzare la sequenza in blocchi da 16 bit

**1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0**
**1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1**
**1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 1**

Sommiamo i primi due blocchi

**1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0** +
**1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1** =
**1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1**

Il riporto viene aggiunto al risultato come bit meno significativo

**1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1** +
**1** =
**1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 0**

Sommiamo il terzo blocco

**1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 0** +
**1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 1** =
**1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1**

Il riporto viene aggiunto al risultato come bit meno significativo

**0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1** +
**1** =
**0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0**

Eseguiamo il complemento a 1 e troviamo il valore del checksum
1 0 1 1 1 1 1 1 1 1 1 1 0 1

# Ex. n. 7

Si consideri il trasferimento di dati tra due stazioni, A e B, attraverso una connessione TCP con le seguenti ipotesi:

(i) solo la stazione A riceve dalla propria applicazione dati da inviare, e in particolare un primo blocco di 4500 byte al tempo t = 0, e un secondo blocco di 4000 byte al tempo $t\_1 = 3{,}5 * T_s$, dove il tempo di trasmissione $T_s$ di un segmento di dimensione massima MSS = 1500 byte è dato da $T_s = 1{,}5 * \tau$, dove $\tau$=10 *ms* è il ritardo di propagazione tra le due stazioni;

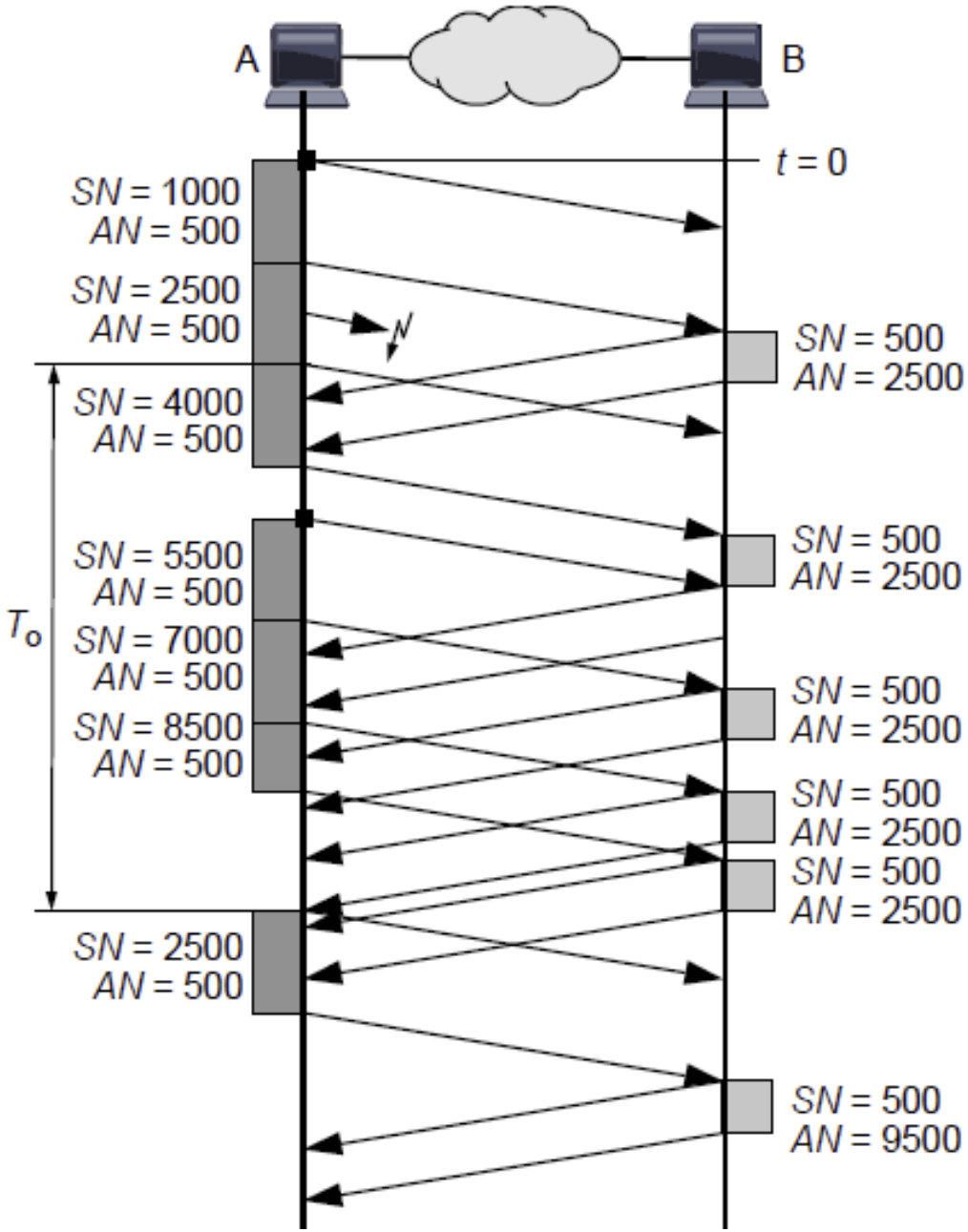
(ii) il secondo segmento inviato dalla stazione A non viene ricevuto dalla stazione B;

(iii) il tempo di trasmissione di un riscontro è $T_a = T_s /2$; la numerazione iniziale $ISN_A$= 1000 e $ISN_B$ = 500.

Si chiede di determinare per questo caso il valore del time-out $T_0$ oltre il quale la procedura fast retransmit consente di ritrasmettere più velocemente il segmento perduto

Determinare il valore richiesto per il time-out richiede di calcolare il tempo richiesto alla procedura fast retransmit per recuperare il segmento perduto. Come mostrato in figura, la stazione A, dopo il riscontro del primo segmento, riceve 3 ACK ripetuti con stessa numerazione AN, cosa che indica perdita di segmento secondo la procedura fast retransmit.

Alla ricezione del terzo riscontro ripetuto la stazione A procede a ritrasmettere il segmento che si assume sia andato perso. Il tempo che intercorre tra la fine della prima trasmissione del segmento e l'inizio della sua ritrasmissione determina il valore del parametro $T_0$ richiesto.




$$T_o = t_1 - 2T_s + 2T_s + T_a + 2\tau = 4T_s + 2\tau = 8\tau = 80:$$

# Ex. n. 8

Si calcoli il nuovo valore del round trip time se la stima corrente è pari a RTT = 15 *ms* e si ricevono 5 riscontri che risultano nei ritardi misurati di RTT uguali a 18, 23, 24, 32 *ms*.

Si assuma che α=1/8.

Soluzione ex. N. 8

L'aggiornamento del parametro RTT si basa sull'equazione $RTT_{av}=(1-1/8)RTT_{av}+(1/8)RTT_{last}$ dove $RTT_{av}$ rappresenta il valore corrente e $RTT_{last}$ indica l'ultima rilevazione ricevuta (e α=1/8). Quindi si ricavano i nuovi valori di RTT

$$RTT_{av1} = \left(1 - \frac{1}{8}\right) \cdot 15 + \frac{1}{8} \cdot 18 = 15,375\text{ms}$$

$$RTT_{av2} = \left(1 - \frac{1}{8}\right) \cdot 15,375 + \frac{1}{8} \cdot 23 = 16,328\text{ms}$$

$$RTT_{av3} = \left(1 - \frac{1}{8}\right) \cdot 16,328 + \frac{1}{8} \cdot 24 = 17,287\text{ms}$$

$$RTT_{av4} = \left(1 - \frac{1}{8}\right) \cdot 17,287 + \frac{1}{8} \cdot 32 = 19,126\text{ms}$$

Si osservi inoltre come il meccanismo di aumento con media mobile pesata determini una variazione più lenta della variazione effettivamente rilevata nel parametro RTT.

# Ex. n. 9

Si chiede di rappresentare l'andamento dell'apertura della congestion window per i protocolli TCP Tahoe e TCP Reno per i primi 40 intervalli di RTT nell'ipotesi che:

(i) uno dei segmenti inviati al tempo 13 RTT venga perso e vengano ricevuti solo ACK duplicati;

(ii) che nell'intervallo (14-15) RTT la rete sia guasta e nessun segmento venga trasmesso;

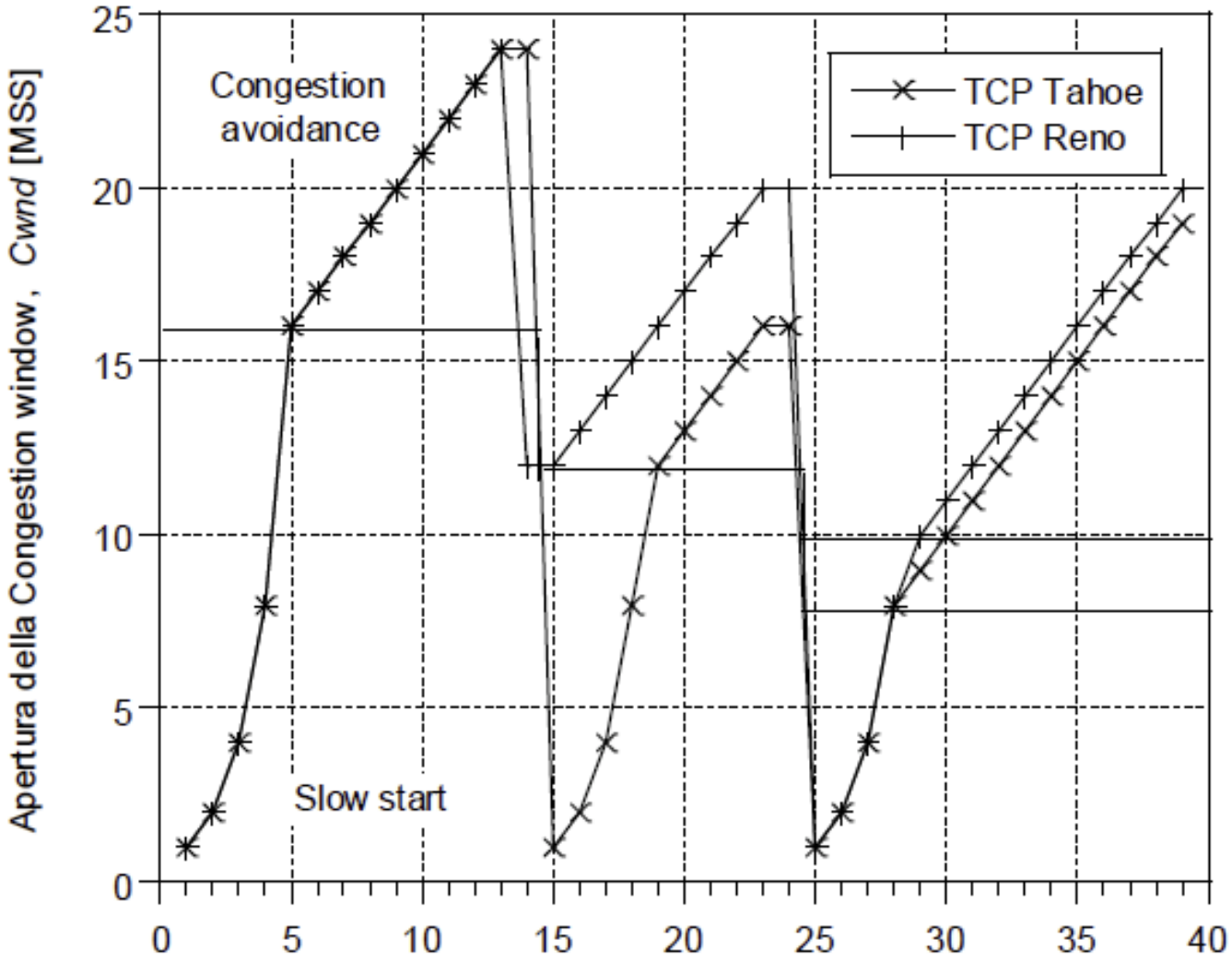
(iii) la rete vada fuori servizio nell'intervallo (23-25) RTT (e la perdita viene rilevata mediante timeout expiration).

I valori iniziali sono: soglia tra le due regioni Ssthresh = 16 MSS, congestion window Cwnd = 1 MSS, time-out $T_0$ = 2 RTT. La trasmissione ha inizio al tempo 1 RTT.

# Soluzione ex 9

L'andamento del parametro Cwnd per i due protocolli è riportato in figura.

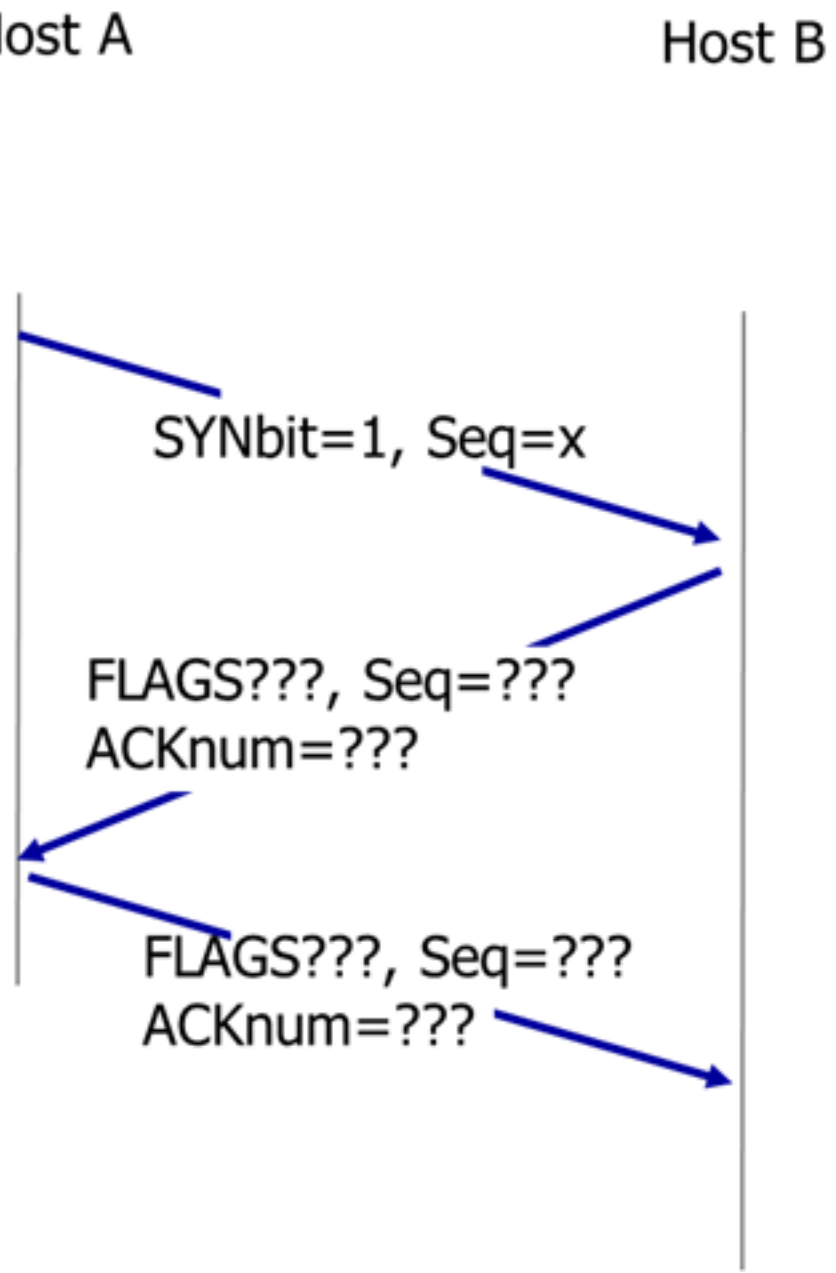
- Per i primi 5 cicli di RTT si opera nella regione slow start con aumento esponenziale di Cwnd,
- per poi entrare nella regione di congestion avoidance caratterizzata da aumento lineare.
- La perdita di segmento trasmesso al tempo 13 RTT viene rilevata al tempo 14 RTT per mezzo di 3 ACK duplicati con il protocollo TCP Reno, al tempo 15 RTT per mezzo della scadenza del time-out con il protocollo TCP Tahoe.
  Il nuovo valore della soglia diviene Ssthresh = 12 MSS, dato che il numero di segmenti trasmessi privi di riscontro è Flightsize = 24. Nel caso del protocollo TCP Tahoe, la congestion window riparte dal valore unitario e aumenta in modo esponenziale fino al nuovo valore di soglia per poi aumentare in modo lineare. Nel caso del protocollo TCP Reno la congestion window riparte dal valore Cwnd = Ssthresh = 12 MSS e l'aumento continua in modo lineare dato che ci si trova nella regione congestion avoidance.
- Al verificarsi di perdita rivelata della scadenza del time-out al tempo 25 RTT la soglia assume il nuovo valore Ssthresh = 8 MSS con il protocollo TCP Tahoe e Ssthresh = 10 MSS con il protocollo TCP Reno, cioè metà dell'ultimo valore di Cwnd, che indica anche il numero di segmenti inviati privi di riscontro. In entrambi i casi la nuova apertura della congestion window è quella minima Cwnd = 1 MSS.
- All'aumentare di RTT l'apertura aumenta in modo esponenziale in entrambi i protocolli fino al raggiungimento delle rispettive soglie, dopo cui l'aumento torna a essere lineare.



# Ex. n. 10

Un host A inizia una connessione TCP verso un host B. Nella figura viene evidenziato che il FLAG SYN del primo segmento della connessione è attivato (posto uguale ad 1) e il sequence number assume valore x.
Rispondere alle domande seguenti

1. Quali (o quale flag) sono attivi nel secondo segmento (quello dall'host B verso l'host A)? Ed i valori del sequence number e dell'ACK number?
2. Quali (o quale flag) sono attivi nel terzo segmento (quello dall'host A verso l'host B)? Ed i valori del sequence number e dell'ACK number?

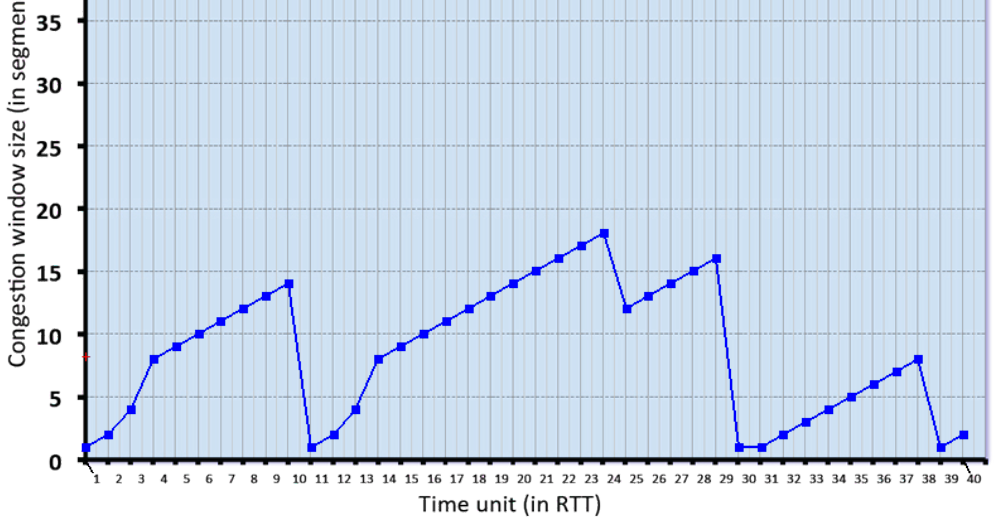


# Ex. n. 11

Considerare la figura alla destra, che mostra l'evoluzione della finestra di congestione del TCP all'inizio di ogni unità di tempo (dove l'unità di tempo è uguale all'RTT).
Nel modello astratto per questo problema, TCP invia una sequenza di segmenti di dimensioni cwnd all'inizio di ogni unità di tempo. Come risultato dell'invio di quella sequenza di segmenti si può verificare uno dei seguenti eventi:

- tutti i segmenti sono riscontrati alla fine del round; l'unità di tempo,
- c'è un timeout per il primo pacchetto,
- c'è un triplo ACK duplicato per il primo pacchetto.

In questo quesito, viene richiesto di ricostruire la sequenza di eventi (ACK, perdite) che hanno portato all'evoluzione del cwnd di TCP come viene evidenata dalla figura seguente.




Rispondere alle domande seguenti:

1. Indicare gli intervalli durante i quali TCP è nello stato di slow start.
2. Indicare gli intervalli durante i quali TCP è nello stato di congestion avoidance.
3. Indicare i round in cui TCP è in fast recovery.
4. Indicare i round in cui i pacchetti vengono persi per timeout.
5. Indicare i round in cui le perdite di pacchetti vengono rilevate mediante triple duplicate ACK.
6. Indicare i round in cui il valore di ssthresh cambia.