

### 0.0.1 Appello del 15/07/2008

ESERCIZIO 1: Consideriamo due host  $H_1$  ed  $H_2$  collegati mediante una linea punto-punto esente da errori ( $BER = 0$ ) e di rate  $R$  bps. Supponiamo che i due host utilizzino il TCP/IP e che  $H_1$  voglia trasferire un file di dimensione infinita ad  $H_2$ . Assumiamo inoltre che il meccanismo di controllo della congestione sia lo *slow start* e che il trasferimento del file inizi subito dopo l'apertura della connessione di trasporto. Il TCP che consideriamo ha però le seguenti caratteristiche che lo rendono difforme dal TCP standard: a) il valore della *Congestion Threshold* ( $L$ ) subito dopo l'apertura della connessione vale  $2 MSS$  anziché essere infinita; b) solo quando la *Congestion Window* vale  $4 MSS$  l'ultimo segmento in essa compreso viene alterato durante la trasmissione ad  $H_2$ ; c) la ritrasmissione di tale segmento va a buon fine.

Al fine di semplificare i calcoli richiesti supponiamo che:

- (i) tutti i segmenti dati abbiano la stessa dimensione  $m = MSS$  bits, tempo di trasmissione  $T = m/R$  msec e *timeout* pari a  $2T$ ;
- (ii) la latenza dei segmenti dati sia  $T$  msec;
- (iii) i segmenti di *ACK* abbiano tempi di trasmissione trascurabili;
- (iv) la latenza degli *ACK* sia nulla e che gli *ACK* non si alterino mai;
- (v)  $H_1$  riceve un *ACK* per ogni segmento ricevuto correttamente da  $H_2$ ;
- (vi) la finestra di *Flow Control* del TCP sia infinita.

Il candidato:

1. illustri la dinamica del sistema;
  2. calcoli il throughput  $\gamma$  dopo aver specificato in una tabella la corrispondenza tra la  $k$ .ma  $\{k = 1, 2, 3, \dots\}$  *Congestion Window*, il numero di segmenti compresi nella *Congestion Window* ( $CW(k)$ ) e l'intervallo di tempo ( $t(k)$ ) che intercorre tra la trasmissione del primo bit del primo segmento della  $k$ .ma *Congestion Window* e l'istante in cui viene ricevuto l'*ACK* relativo all'ultimo segmento della medesima;
  3. generalizzi il calcolo del throughput  $\gamma$  al caso generale in cui la *Congestion Threshold* valga  $L$ , con  $L$  tale che:  $2^{k-1} = L$ , e supponendo che quando la *Congestion Window* vale  $2L MSS$  (solo in questo caso) l'ultimo segmento in essa compreso venga alterato durante la trasmissione ad  $H_2$ .
  4. dica se il *Fast Retransmit* può essere mai attivato nello scenario esaminato
- Supponiamo infine che sia attivo l'algoritmo di *AIMD* per il controllo

della congestione e che la trasmissione del file considerato precedentemente inizi quando la  $(CW(k) = 2)$ . Per semplificare la scrittura delle formule supponiamo che sia  $(CW(1) = 2)$ . Il candidato:

5. risponda alle precedenti tre domande.

NOTA. Nel calcolare il throughput il candidato ignori il sovraccarico (overhead) introdotto dalle intestazioni dei vari protocolli (data-link layer, IP, TCP)

RISOLUZIONE

1. La Figura 1.1 illustra la dinamica del sistema. Come si evince dalla stessa

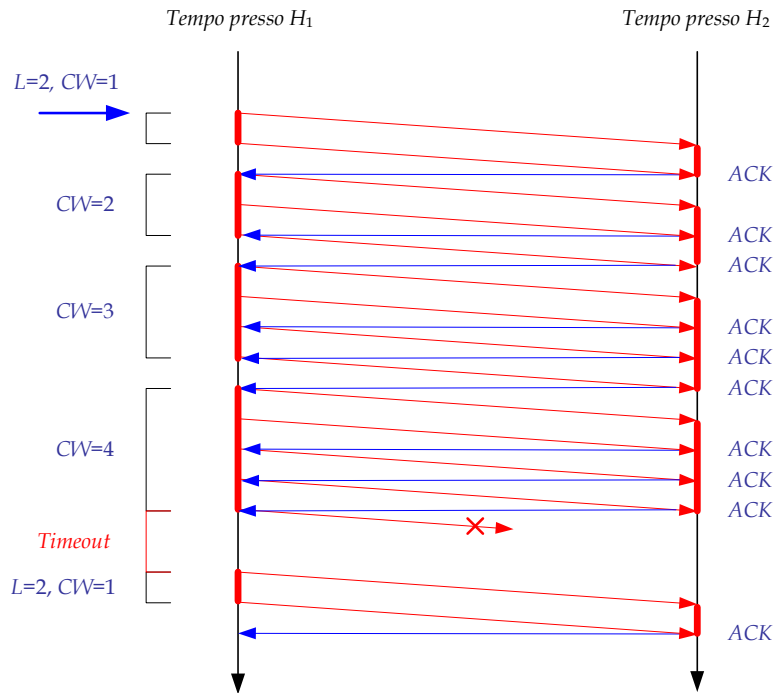


Figura 1.1: Dinamica del sistema nel caso dello *Slow Start*

figura, la connessione di trasporto è nella fase di *Slow Start* nelle prime due *Congestion Window* per poi passare nella fase lineare (*Additive Increase*) nelle altre due *Congestion Window*. Il quarto segmento della quarta *Congestion Window* è affetto da errore. Dopo un tempo pari a  $2T$  dalla fine della trasmissione del segmento, viene rilevata la congestione per cui il processo

precedente si ripete con  $CW = 1$  e  $L = 2$ . In altri termini il comportamento della *Congestion Window* è deterministico tra i due istanti (ciclo) che separano l'occorrenza di due congestioni consecutive. La Figura 1.2 illustra

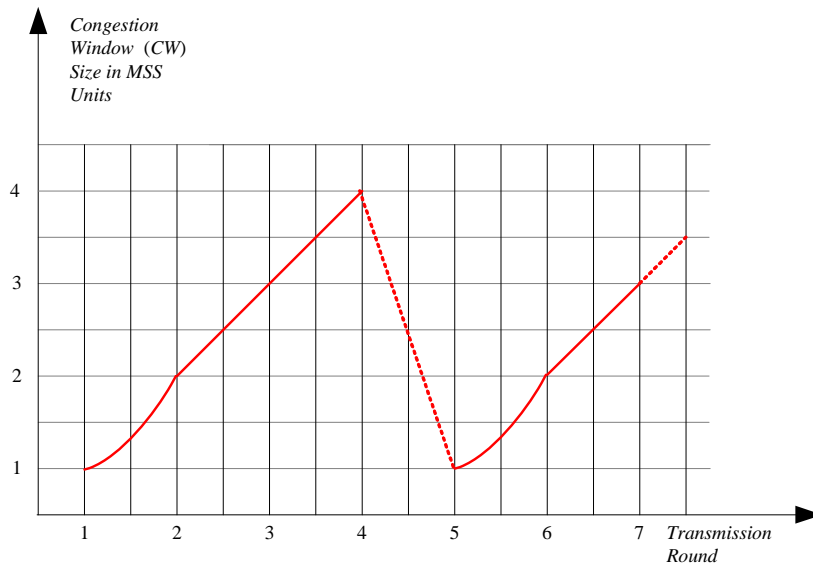


Figura 1.2:  $CW$  in funzione del numero di *Transmission Rounds*

l'andamento della *Congestion Window* in funzione del numero di *transmission round*. Di conseguenza, per determinare il throughput della connessione di trasporto, basta analizzare cosa accade nell'ambito di un generico ciclo.

2. La Figura 1.3 illustra la corrispondenza richiesta. Dalla essa si deduce che

$k$	$CW(k)$	$t(k)$
$k=1$	$CW(1)=2^0$	$t(1)=T+T=2T$
$k=2$	$CW(2)=2^1$	$t(2)=2T+T=3T$
$k=3$	$CW(3)=2^1+1$	$t(3)=3T+T=4T$
$k=4$	$CW(4)=2^1+2$	$t(4)=4T+2T=6T$

Figura 1.3: Relazione tra  $k$ ,  $CW(k)$  e  $t(k)$

ogni ciclo consta di quattro *Congestion Window* che contengono complessi-

vamente 10 *MSS*. Di questi 10 segmenti solo 9 (per complessivi  $9m$  bits) arrivano a destinazione. Per trasmettere correttamente questi  $9m$  bits sono necessari  $15 T$  msec. Di conseguenza il throughput  $\gamma$  risulta:

$$\gamma = \frac{9m}{15T} = 0,6R \tag{1.1}$$

3. L'andamento della *Congestion Window* in funzione del numero di *transmission round* è riportata in Figura 1.4. Tale andamento può essere rappre-

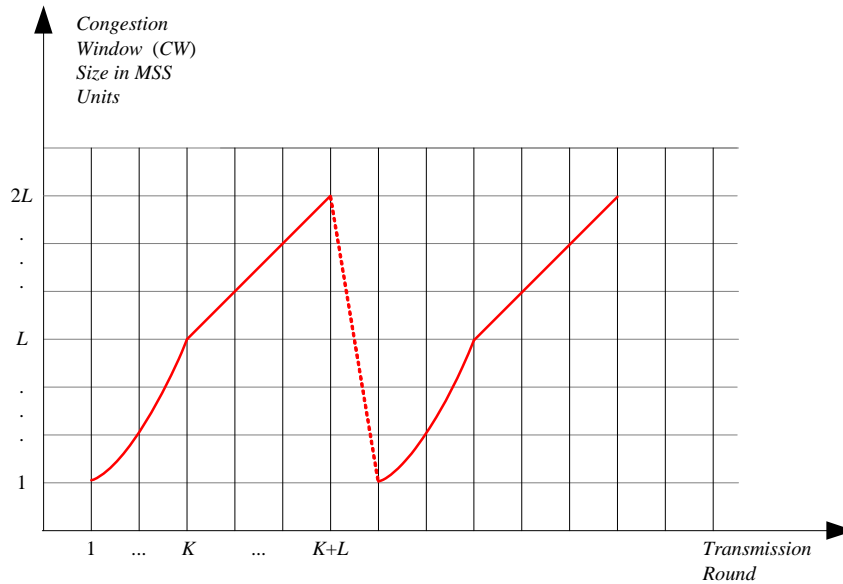


Figura 1.4: Andamento della *Congestion Window* nel caso generale

sentato analiticamente mediante l'espressione:

$$CW(k) = \begin{cases} 2^{k-1} & 1 \leq k \leq K, 2^{K-1} = L \\ 2^{K-1} + (k - K) & k > K \end{cases} \tag{1.2}$$

La congestione si verifica perciò in corrispondenza del *transmission round* che indicheremo con  $\bar{K}$ , espresso dalla seguente condizione:

$$2^{K-1} + (\bar{K} - K) = 2L \tag{1.3}$$

Sostituendo  $2^{K-1} = L$  nella (1.3) si ottiene

$$\bar{K} = L + K \quad (1.4)$$

Tale risultato era prevedibile in quanto a partire dal  $K$ .mo *transmission round* (in corrispondenza del quale  $CW$  vale  $L$ ) la  $CW$  viene incrementata di un  $MSS$  per *transmission round*. Di conseguenza la  $CW$  passa da  $L$  a  $2L$  dopo  $L$  *transmission rounds*.

Per calcolare il numero di segmenti trasmessi correttamente ( $M$ ) in un ciclo basta sommare il numero di segmenti compresi nelle  $CW$  fino alla  $CW$  di ampiezza  $2L$ . Bisogna però tener presente che l'ultimo segmento della  $CW$  che comprende  $2L$  segmenti viene alterato. Dunque

$$M = \sum_{k=1}^{\bar{K}} 2^{k-1} + \sum_{k=K+1}^{\bar{K}} [2^{K-1} + (k-K)] - 1 \quad (1.5)$$

Nella seconda sommatoria conviene cambiare indice, ossia poniamo  $j = k - K$ . Con tale cambiamento la (1.5) può essere riscritta

$$M = \frac{1-2^{\bar{K}}}{1-2} + \sum_{j=1}^L [2^{K-1} + j] - 1$$

Tenendo presente che  $2^{K-1} = L$ , dopo alcuni passaggi algebrici si ottiene

$$M = \frac{3L^2 + 5L - 4}{2} \quad (1.6)$$

Se istanziamo la (1.6) con il valore  $L = 2$  (v. punto precedente) otteniamo (ovviamente)  $M = 9$ .

Per calcolare la lunghezza  $\Lambda$  del ciclo bisogna tener presente che per trasmettere i segmenti compresi nella *Congestion Window* relativa al  $k$ .mo *transmission round*, ossia  $CW(k)$ , con  $k = \{1, 2, \dots, \bar{K} = K + L - 1\}$ , sono necessari  $T[CW(k)] + T$  secondi dove il primo addendo comprende il tempo di trasmissione dei segmenti compresi nella  $CW(k)$  mentre il secondo addendo tiene conto della latenza  $T$  del mezzo. Per quanto riguarda l'ultimo segmento della  $CW(\bar{K} = K + L) = 2L$  bisogna tener conto del *time-out* di ampiezza pari a  $2T$ . Quindi

$$\Lambda = \sum_{k=1}^{\bar{K}} [(2^{k-1})T + T] + \sum_{k=K+1}^{\bar{K}} \{[2^{K-1} + (k-K)]T + T\} + 2LT + 2T$$

Dopo vari passaggi algebrici si perviene al seguente risultato

$$\Lambda = \frac{LT}{2}(7 + 3L) + KT \tag{1.7}$$

Se istanziamo la (1.7) con i valori  $L = 2$  e  $K = 2$  (v. punto precedente) otteniamo (ovviamente)  $\Lambda = 15$ .

Dalle (1.6) e (1.7) si ottiene

$$\gamma = \frac{Mm}{\Lambda} = \frac{(3L^2 + 5L - 4)m}{LT(7 + 3L) + KT} \tag{1.8}$$

4. Il *Fast Retransmit* non può mai essere attivato. Infatti  $H_1$  non riceve nessun *DUPACK* in quanto si altera l'ultimo segmento compreso nella *CW*.

5. La dinamica del sistema è illustrata in Figura 1.5. La relazione tra

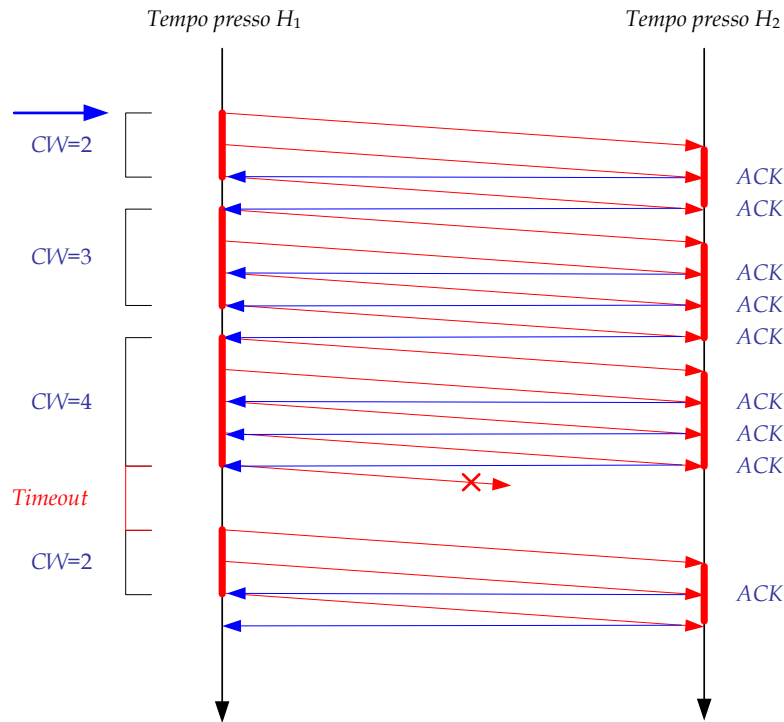


Figura 1.5: Dinamica del sistema nel caso di AIMD

$CW(k)$  il numero di segmenti compresi nella  $CW(k)$  e  $t(k)$  è riportata in

$k$	$CW(k)$	$t(k)$
$k=1$	$CW(2)=2$	$t(2)=2T+T=3T$
$k=2$	$CW(3)=2+1=3$	$t(3)=3T+T=4T$
$k=3$	$CW(4)=3+1=4$	$t(4)=4T+2T=6T$

Figura 1.6: Relazione tra  $k$ ,  $CW(k)$  e  $t(k)$  nel caso di *AIMD*

Figura 1.6. Dalle considerazioni fatte precedentemente per lo *Slow Start* si ricava facilmente che:

$$\gamma = \frac{8m}{13T} \cong 0,61R \quad (1.9)$$

Nel caso particolare descritto nell'esercizio il throughput di *AIMD* risulta perciò leggermente superiore a quello dello *Slow Start*.

