

# Esercizi: Telecomunicazioni – parte Reti

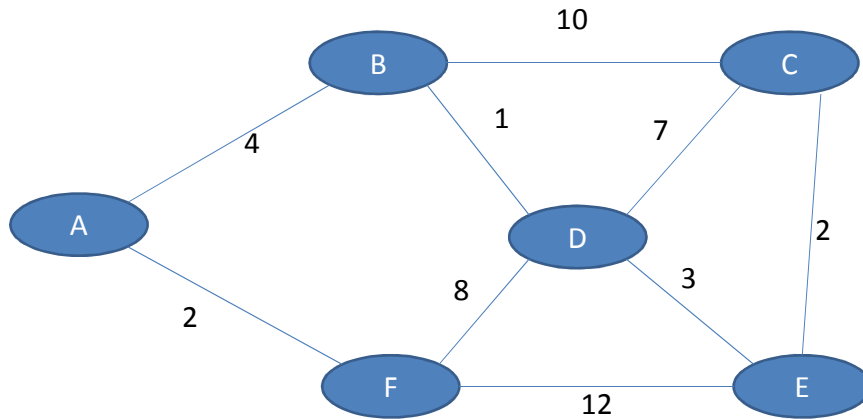
---

## Indice

Indice .....	1
Routing .....	2
Esercizio 1: Link state routing.....	2
Esercizio 2: Distance vector routing.....	4
Esercizio 3: Distance vector routing.....	6
Controllo di errore.....	8
Esercizio 5: go-back N.....	8
Esercizio 6: Selective Repeat .....	10
Controllo di flusso.....	12
Esercizio 7.....	12
Controllo di congestione .....	14
Esercizio 8.....	14

# Routing

## Esercizio 1: Link state routing



Si consideri la rete la cui topologia è rappresentata in figura. Si utilizzi l'algoritmo di routing Dijkstra a stato di collegamento, per calcolare passo passo la tabella di routing nel nodo E.

### Soluzione

Applichiamo l'algoritmo di Dijkstra (riportato di seguito) al nodo E per calcolare il vettore delle distanze  $D$  e dei predecessori  $p$ .

\*\*\*\*\*

1. Inizializza
  - a.  $N' = \{E\}$
  - b. Ripeti per tutti i nodi  $V$ 
    - i. Se  $V$  è adiacente ad  $E$ ,  $D(V) = c(E, V)$ , altrimenti  $D(V) = \infty$
2. Ripeti finchè  $N \neq N'$ 
  - a. Scegli tra i nodi non presenti in  $N'$  il nodo  $W$  con  $D(W)$  minimo ed aggiungilo all'insieme  $N'$
  - b. Ripeti per ogni nodo  $V$  non presente in  $N'$  vicino al nodo  $W$ 
    - i. aggiorna la sua distanza  $D(V)$  con la regola  $D(V) = \min[D(V); c(V, W) + D(W)]$
    - ii. annotando il nodo predecessore  $p(W)$

\*\*\*\*\*

La tabella di seguito riporta i passi dell'algoritmo (passi 1. prima riga, passi 2. righe successive)

N'	D(A),p(A)	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(F),p(F)
{E}	$\infty$	$\infty$	<b>2,E</b>	3,E	12,E
{E,C}	$\infty$	12,C		<b>3,E</b>	12,E
{E,C,D}	$\infty$	<b>4,D</b>			11,D
{E,C,D,B}	<b>8,B</b>				11,D
{E,C,D,B,A}					<b>11,D</b>
{E,C,D,B,A,F}					

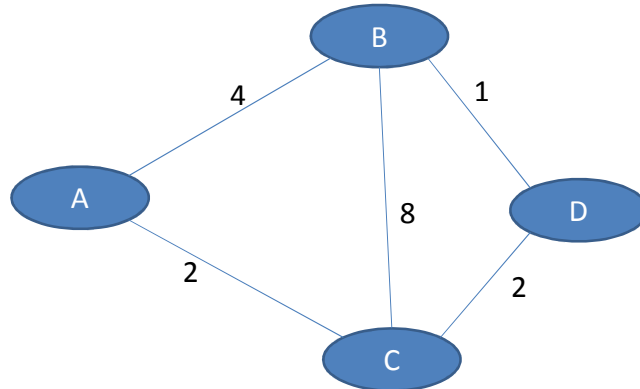
Dalla tabella appena ricavata si ottengono i percorsi ottimi (a costo minimo) riportati in tabella:

Destinazione	Percorso ottimo
A	E->D->B->A
B	E->D->B
C	E->C
D	E->D
F	E->D->F

Pertanto la tabella di routing nel nodo E è la seguente

Destinazione	Prossimo nodo
A	D
B	D
C	C
D	D
F	D

## Esercizio 2: Distance vector routing



Si consideri la rete la cui topologia è rappresentata in figura. Si supponga in ciascun nodo presente una tabella di routing calcolata tramite un algoritmo con vettore distanza minimizzando il costo complessivo dell'attraversamento della rete. Si supponga in un certo istante che si guasti il nodo D.

Applicare l'algoritmo di routing con vettore distanza per ricalcolare le tabelle di routing nei nodi A,B,C dopo tre scambi di messaggi. Si supponga che i nodi eseguano calcoli e si scambino messaggi in modo sincrono. Ovvero, i messaggi di aggiornamento di aggiornamento, quando necessari, sono inviati in istanti  $t_k$  tali che  $t_{k+1} = t_k + T$ .

### Soluzione:

Prima del guasto in ogni nodo sono memorizzati i seguenti vettori distanza (ogni riga è un vettore distanza)

	A	B	C	D
A	0	4	2	4
B	4	0	3	1
C	2	3	0	2

NODO A

	A	B	C	D
A	0	4	2	4
B	4	0	3	1
C	2	3	0	2
D	4	1	2	0

NODI B,C

	A	B	C	D
B	4	0	3	1
C	2	3	0	2
D	4	1	2	0

NODO D

Dopo il guasto il costo dei collegamenti BD e CD diventa  $\infty$ . I nodi B e C avvertono il cambiamento di costo e ricalcolano il loro vettore distanza.

	A	B	C	D
A	0	4	2	4
B	4	0	6	8
C	2	3	0	2

NODO B

	A	B	C	D
A	0	4	2	4
B	4	0	3	1
C	2	6	0	6

NODO C

Si noti che il nodo D, non essendo più raggiungibile da B o C non è più considerato un nodo vicino. Tuttavia, né B, né C possono dedurre che sia caduto e che non sia più raggiungibile in nessun modo.

Il vettore distanza di B e C ricalcolati sono trasmessi ai nodi vicini, (A,C) e (A,B) rispettivamente. I nodi A,B,C ricevendo aggiornamenti dei vettori distanza di nodi vicini ricalcolano il loro vettore distanza come segue:

	A	B	C	D
A	0	4	2	8
B	4	0	3	8
C	2	3	0	6

NODO A

	A	B	C	D
A	0	4	2	4
B	4	0	6	8
C	2	6	0	6

NODO B

	A	B	C	D
A	0	4	2	4
B	4	0	6	8
C	2	6	0	6

NODO C

Da questo momento in poi le distanze tra i nodi A,B,C sono calcolate correttamente e rimangono stabili, le distanze dal nodo D, invece, crescono lentamente verso l'infinito (count to infinity problem) continuando i nodi A,B,C a scambiarsi messaggi di aggiornamento dei loro vettori distanza.

Le tabelle di routing in A,B,C sono corrette per le destinazioni A,B,C. Tuttavia, includono ancora una destinazione irraggiungibile D. Infatti i nodi B e C non raggiungendo il nodo D, possono imputare l'irraggiungibilità di D alla sola caduta del collegamento che li collega direttamente a D e non alla caduta dell'intero nodo D.

destinazione	Prossimo nodo
B	B
C	C
D	C

NODO A

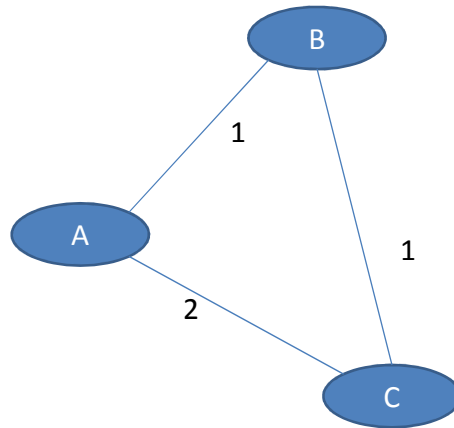
destinazione	Prossimo nodo
A	A
C	A
D	A

NODO B

destinazione	Prossimo nodo
A	A
B	A
D	A

NODO C

### Esercizio 3: Distance vector routing



Si consideri la rete la cui topologia è rappresentata in figura. Si supponga in ciascun nodo presente una tabella di routing calcolata tramite un algoritmo con vettore distanza minimizzando il costo complessivo dell'attraversamento della rete. Si supponga in un certo istante che si guasti il collegamento BC.

Applicare l'algoritmo di routing con vettore distanza per ricalcolare le tabelle di routing nei nodi A,B,C. Si supponga che i nodi eseguano calcoli e si scambino messaggi in modo sincrono. Ovvero, i messaggi di aggiornamento, quando necessari, sono inviati in istanti  $t_k$  tali che  $t_{k+1} = t_k + T$ .

#### Soluzione:

Prima del guasto in ogni nodo sono memorizzati i seguenti vettori distanza (ogni riga è un vettore distanza)

	A	B	C
A	0	1	2
B	1	0	1
C	2	1	0

Dopo il guasto il collegamento diretto tra B e C assume costo infinito. I nodi B e C avvertono il cambiamento di costo e ricalcolano il loro vettore distanza come segue.

	A	B	C
A	0	1	2
B	1	0	3

NODO B

	A	B	C
A	0	1	2
C	2	3	0

NODO C

I vettori distanza di B e C ricalcolati sono trasmessi al nodo vicino A. Il nodo A ricevendo aggiornamenti dei vettori distanza dei nodi vicini B e C ricalcola il vettore distanza come segue:

	A	B	C
A	0	1	2
B	1	0	3
C	2	3	0

NODO A

Come si può notare, il nodo A non esegue nessuna modifica sul suo vettore delle distanze in quanto la caduta del link tra B e C non influisce sul cammino ottimo di istradamento del nodo A. Pertanto, l'algoritmo si arresta con queste tre tabelle delle distanze

	A	B	C
A	0	1	2
B	1	0	3
C	2	3	0

NODO A

	A	B	C
A	0	1	2
B	1	0	3

NODO B

	A	B	C
A	0	1	2
C	2	3	0

NODO C

Le tabelle di routing in A,B,C dedotte dai vettori distanza sono le seguenti

destinazione	Prossimo nodo
B	B
C	C

NODO A

destinazione	Prossimo nodo
A	A
C	A

NODO B

destinazione	Prossimo nodo
A	A
B	A

NODO C

## Controllo di errore

### Esercizio 5: go-back N

Si consideri un protocollo di trasporto affidabile (con recupero di errore) operante su un canale tra due nodi A e B e con le seguenti caratteristiche:

- Recupero di errore con ritrasmissione non-selettiva
- Dimensione dei segmenti dati costante e pari a L
- Dimensione dei riscontri costante e pari ad L/3
- Finestra di trasmissione  $W_T=3$
- Finestra in ricezione  $W_R=1$

Il canale tra A e B abbia le seguenti caratteristiche:

- Capacità C
- Ritardo di propagazione pari a  $2/3$  del tempo di trasmissione di un segmento

Si supponga di dover trasferire 7 segmenti da A a B.

*Domande:*

1. Calcolare la durata del trasferimento nell'ipotesi di assenza di errori (durata nominale del trasferimento)
2. Disegnare il diagramma spazio-tempo nell'ipotesi che il trasferimento avvenga con un errore al primo tentativo di trasferimento del quinto segmento.
3. Calcolare la durata del trasferimento nell'ipotesi di errore sul quinto segmento di cui al punto 2 (considerare il tempo che intercorre tra l'inizio dell'emissione del primo segmento ed il termine della ricezione corretta dell'ultimo segmento)

### Soluzione:

Domanda 1.

La durata nominale è pari al tempo di trasmissione di sette segmenti,

$$7*(L/C)$$

più il ritardo di propagazione

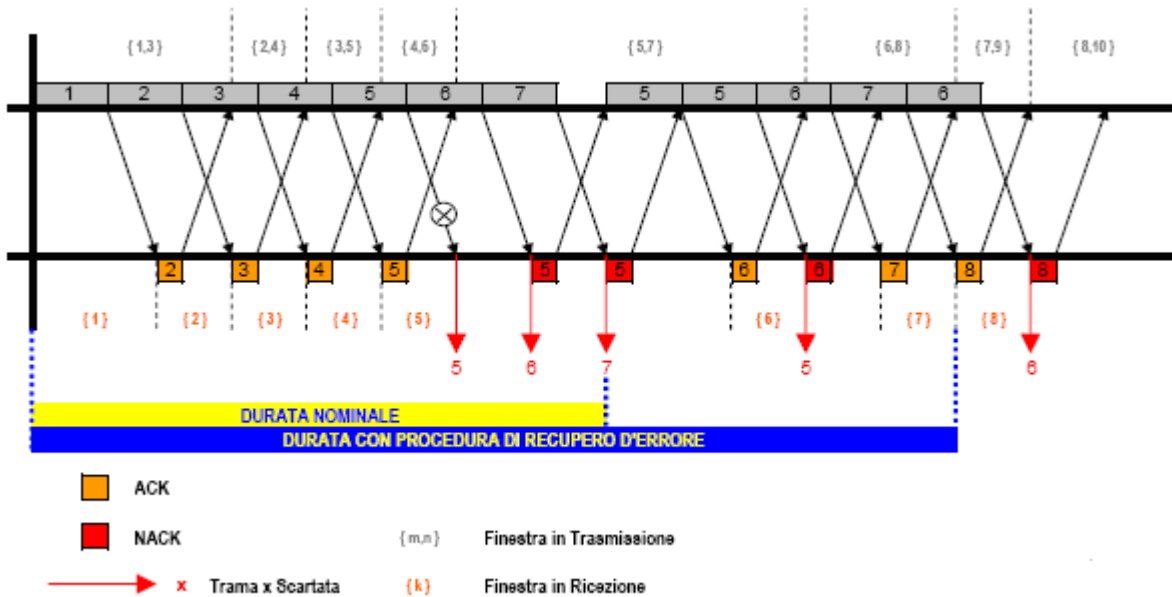
$$2/3*(L/C)$$

ovvero

$$23/3*(L/C)$$



Domanda 2.



Domanda 3.

Come evidenziato nella figura di cui al punto 2,

1. il trasmettitore invia i 7 segmenti senza fermarsi, ricevendo riscontri di corrette ricezioni fino al segmento 4 e portando quindi nella finestra in trasmissione i segmenti 5,6,7;
2. successivamente alla perdita del segmento 5, il trasmettitore riceve quindi due richieste di ritrasmissione duplicate del segmento 5 in conseguenza della ricezione dei segmenti 6 e 7 senza l'arrivo del segmento 5; la prima richiesta di ritrasmissione arriva dopo  $2/3 * L/C$  dal primo invio del pacchetto 7;
3. le due richieste di ritrasmissione duplicate di 5, determinano due invii del segmento 5;
4. si reinvia quindi i segmenti 6 e 7;
5. la procedura di errore per i segmenti 1-7 termina con l'arrivo del segmento 7 al ricevitore.

Si può facilmente dimostrare che il tempo totale di trasmissione è pari alla somma dei seguenti termini:

1.  $7 * (L/C)$
2.  $2/3 * (L/C)$
3.  $2 * (L/C)$
4.  $2 * (L/C)$
5.  $2/3 * L/C$

Ovvero  $37 * L/C$

## Esercizio 6: Selective Repeat

Si consideri un protocollo di trasporto affidabile (con recupero di errore) operante su canale tra i nodi A e B, con le seguenti caratteristiche:

- Recupero di errore con ritrasmissione selettiva
- Dimensione dei segmenti dati costante e pari a L
- Dimensione dei riscontrri costante e pari ad L/3
- Finestra di trasmissione  $WT=3$
- Finestra in ricezione  $WR=2$

Si supponga che il canale tra A e B abbia le seguenti caratteristiche:

- Capacità C
- Ritardo di propagazione pari a due terzi del tempo di trasmissione di un segmento

Si supponga di dover trasferire 7 segmenti da A a B.

*Domande:*

1. Calcolare la durata del trasferimento nell'ipotesi di assenza di errori (durata nominale del trasferimento)
2. Disegnare il diagramma spazio-tempo nell'ipotesi che il trasferimento avvenga con un errore al primo tentativo di trasferimento del quinto segmento.
3. Calcolare la durata del trasferimento nell'ipotesi di errore sul quinto segmento di cui al punto 2 (considerare il tempo che intercorre tra l'inizio dell'emissione del primo segmento ed il termine della ricezione corretta dell'ultimo segmento)

**Soluzione:**

Domanda 1.

La durata nominale è pari al tempo di trasmissione di sette segmenti,

$$7*(L/C)$$

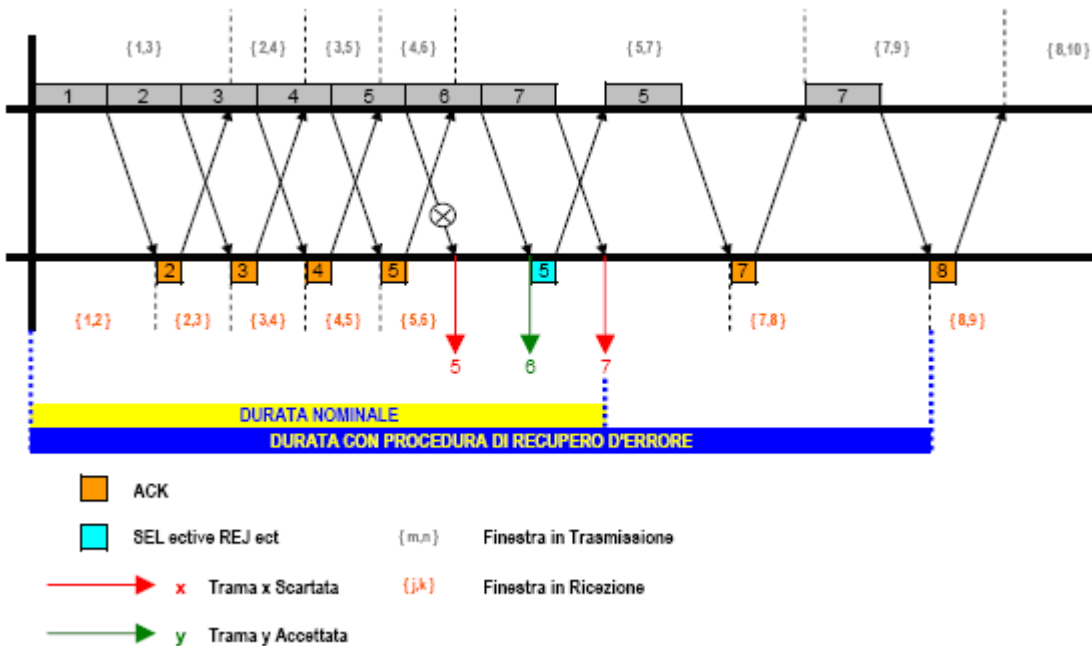
più il ritardo di propagazione

$$2/3*(L/C)$$

ovvero

$$23/3*(L/C)$$

Domanda 2.



Domanda 3.

Come evidenziato nella figura di cui al punto 2,

1. il trasmettitore invia i 7 segmenti senza fermarsi, ricevendo riscontri di corrette ricezioni fino al segmento 4 e portando quindi nella finestra in trasmissione i segmenti 5,6,7;
2. successivamente alla perdita del segmento 5, il trasmettitore riceve quindi una richiesta di ritrasmissione del segmento 5 in conseguenza della ricezione dei segmenti 6 senza l'arrivo del segmento 5; la richiesta di ritrasmissione arriva dopo  $2/3 * L/C$  dal primo invio del pacchetto 7; si noti che il ricevitore non memorizza la ricezione del segmento 7, potendo mantenere nella finestra di ricezione solo due segmenti, il 5 (ancora da ricevere) ed il 6 (ricevuto correttamente)
3. la richiesta di ritrasmissione di 5, determina il rinvio del segmento 5;
4. il trasmettitore riceve una richiesta di ritrasmissione del segmento 7, avendo ricevuto correttamente i segmenti 5 e 6;
5. Il trasmettitore invia il segmento 7;
6. la procedura di errore per i segmenti 1-7 termina con l'arrivo del segmento 7 al ricevitore.

Si può facilmente dimostrare che il tempo totale di trasmissione è pari alla somma dei seguenti termini:

1.  $7 * (L/C)$
2.  $2/3 * (L/C)$
3.  $L/C$
4.  $5/3 * (L/C)$
5.  $L/C$
6.  $2/3 * (L/C)$

Ovvero  $12 * L/C$ .

Si fa notare che avendo avuto una finestra in ricezione di 3, non si sarebbe persa la ricezione del segmento 7 al primo invio e la procedura di errore si sarebbe accorciata considerevolmente.

## Controllo di flusso

### Esercizio 7

Si consideri una connessione TCP tra un client ed un server per il trasferimento affidabile di un file di dimensione  $O=256\text{KB}$ . Si supponga che

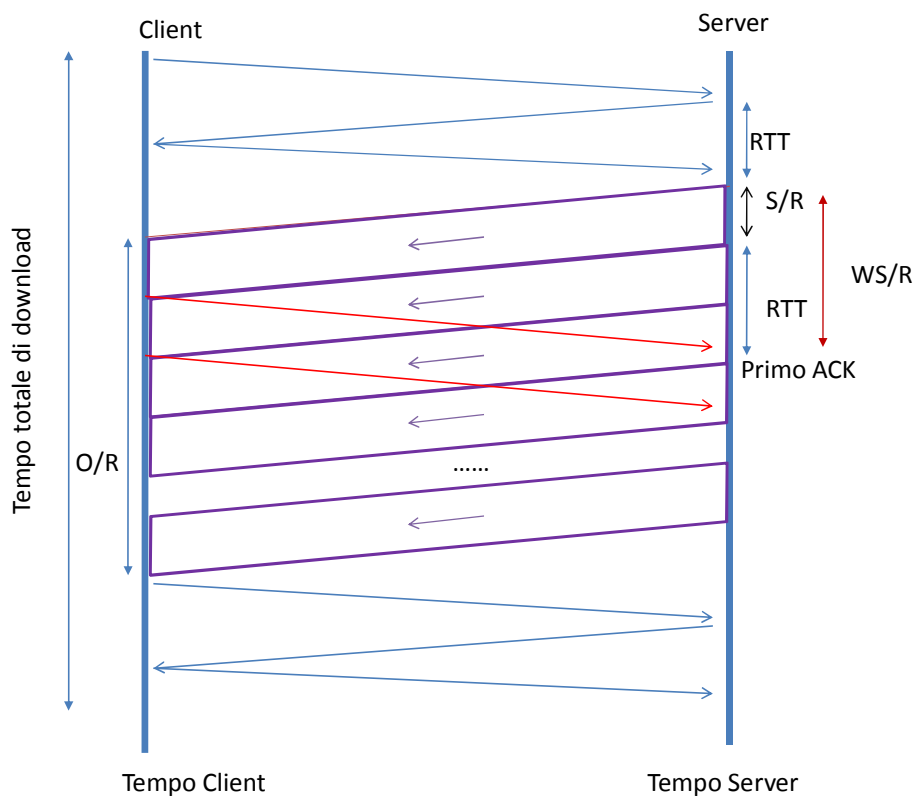
- il trasferimento avvenga tramite segmenti di dimensione costante e pari a  $S=512\text{ B}$
- non si verifichino errori di trasferimento o perdite di pacchetti e che la finestra di congestione del TCP rimanga costante
- il tasso di trasferimento rimanga costante e pari a  $R=32\text{Kb/s}$
- il round trip time tra client e server sia pari a  $\text{RTT}=200\text{ ms}$

Domande:

Assumendo i tempi di trasmissione dei segmenti di segnalazione TCP trascurabili

1. Determinare la dimensione minima della finestra di congestione  $W$  in byte per la quale si riesce a trasmettere dal server al client segmenti TCP senza interruzioni.
2. Determinare il tempo totale di download lato client nelle condizioni di cui al punto 1 per il trasferimento completo del file dal server al client (inclusi i tempi di instaurazione e abbattimento della connessione TCP, supposti entrambi con modalità three-way handshaking).

Soluzione:



Domanda 1.

Per avere la minima latenza di trasferimento, il server deve ricevere il primo ACK prima di completare l'invio dei segmenti validi della prima finestra di congestione. Ciò avviene quando

$$W \cdot S/R > RTT + S/R$$

da cui risulta che la minima finestra di congestione  $W$  è

$$W = \lceil (RTT \cdot R/S) + 1 \rceil = 3$$

Domanda 2.

La minima latenza è di:  $3,5 \cdot RTT + O/R = 64,7$  s

# Controllo di congestione

## Esercizio 8

Si consideri l'instaurazione di una connessione TCP Reno per il trasferimento di un file corrispondente a 13 segmenti, su un canale con  $RTT=400$  ms, con i seguenti parametri:

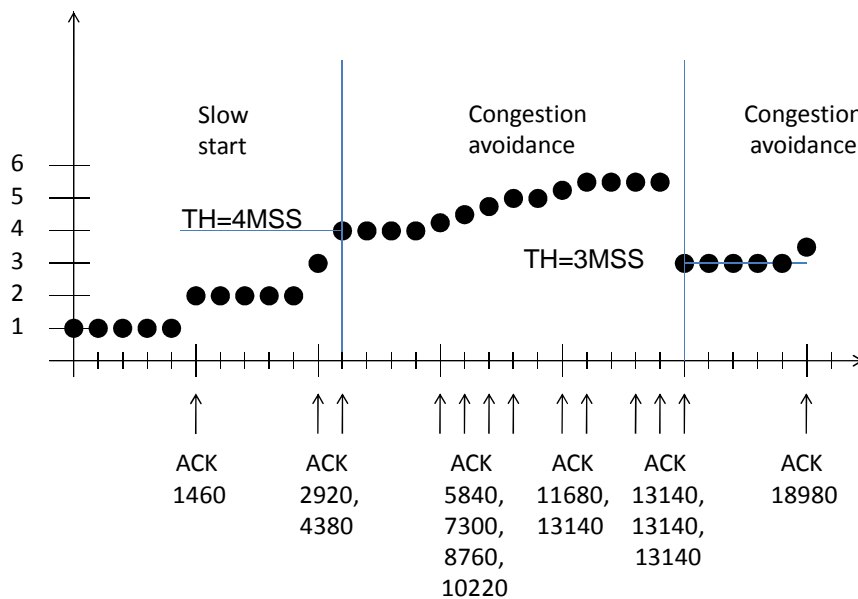
- $MSS= 1460$  byte (primo byte ad essere trasmesso 0)
- Tempo di trasmissione dei segmenti dati pari a  $T=100$  ms
- Tempo di trasmissione dei riscontri trascurabile rispetto ad  $RTT$  e tempo di trasmissione dei segmenti dati
- Finestra di ricezione  $W_R=14600$  byte
- Soglia iniziale di slow start  $TH=4*MSS$ , soglie successive  $TH=k*MSS$  con  $k$  pari all'estremo superiore del rapporto  $W_{cong}/MSS$ , dove  $W_{cong}$  è la finestra di congestione

Si supponga che durante il trasferimento si perda il decimo segmento e che non si verifichino ulteriori perdite.

Domande:

1. Disegnare l'evoluzione della finestra di congestione durante il trasferimento del file
2. Si disegni il diagramma temporale del trasferimento del file

Domanda 1.



Domanda 2.

