



ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

# Architettura di Internet

## a.a. 2022/2023

### Esercitazione 4

Ulderico Vagnoni  
[ulderico.vagnoni2@unibo.it](mailto:ulderico.vagnoni2@unibo.it)

[www.unibo.it](http://www.unibo.it)

## TCP - Congestion avoidance

Una congestione di rete si verifica quando la quantità di dati trasmessi attraverso una rete supera la capacità della rete stessa di elaborarli e di trasmetterli.

Una volta che la capacità della rete è superata, la rete può diventare congestionata, causando una diminuzione delle prestazioni, un aumento del tempo di ritardo di trasmissione (latenza) e un aumento del tasso di perdita di pacchetti.

TCP Congestion Avoidance è un meccanismo utilizzato da TCP per evitare la congestione di rete durante le trasmissioni di dati. La congestione di rete si verifica quando la quantità di dati che viene trasmessa attraverso la rete supera la capacità della rete stessa, causando perdite di pacchetti, ritardi e degradazione delle prestazioni.



## Finestra di congestione

La finestra di congestione rappresenta la quantità di dati che un mittente può inviare attraverso la rete in un certo momento, tenendo conto della congestione presente.

La finestra di congestione viene gestita dinamicamente durante la trasmissione dei dati, in modo da evitare il congestionamento della rete e quindi la perdita di pacchetti

Quando la rete è congestionata, la finestra di congestione viene ridotta per evitare che la congestione peggiori, quando la rete è poco utilizzata, la finestra di congestione viene aumentata per sfruttare al massimo la banda disponibile.



## Slow Start

Per prevenire la congestione, TCP utilizza un meccanismo di controllo di congestione che regola la quantità di dati che viene trasmessa attraverso la rete. Durante la fase di Slow Start, TCP aumenta esponenzialmente la finestra di congestione.



## Slow Start Threshold

Il Slow Start Threshold (SSThresh) è il valore di soglia che viene utilizzato durante la fase di slow start per passare alla fase di congestion avoidance.

Il valore di SSThresh viene calcolato come la metà della finestra di congestione attuale nel momento in cui si verifica la perdita del pacchetto (timeout o triple duplicate ACK).

Quindi, per calcolare il valore di SSThresh, è necessario conoscere il valore della finestra di congestione attuale nel momento in cui si verifica la perdita del pacchetto.

Ad esempio, se la finestra di congestione attuale è di 64 pacchetti e si verifica una perdita di pacchetti, il valore di SSThresh sarà calcolato come  $64/2 = 32$ .

Una volta calcolato il valore di SSThresh, il mittente passa dalla fase di slow start alla fase di congestion avoidance e inizia ad aumentare la finestra di congestione in modo più graduale.



## Congestion avoidance

Una volta raggiunto il valore massimo della finestra di congestione, TCP passa alla fase di Congestion Avoidance. Durante questa fase, TCP utilizza un algoritmo che regola la velocità di trasmissione in modo da evitare la congestione di rete. In particolare, TCP riduce la velocità di trasmissione aumentando la finestra di congestione in modo più lento e rilevando la presenza di congestione sulla rete attraverso l'osservazione dei ritardi dei pacchetti e la ricezione di notifiche di congestione dal destinatario.



# Timeout

Il timeout è un evento che si verifica quando il mittente non riceve una conferma di ricezione (ACK) da parte del destinatario entro un determinato intervallo di tempo.

Durante la trasmissione di dati, il mittente invia pacchetti al destinatario e attende una conferma di ricezione (ACK). Se il mittente non riceve l'ACK entro un certo intervallo di tempo, il pacchetto viene considerato perso.

Nel caso di un timeout, la finestra di congestione viene fissata a 1 e la slow start threshold assume il valore pari alla finestra di congestione durante il timeout diviso 2.



## Triplo acknowledgment duplicato

Un triplo acknowledgment duplicato è un evento che si verifica nel caso di un pacchetto perso.

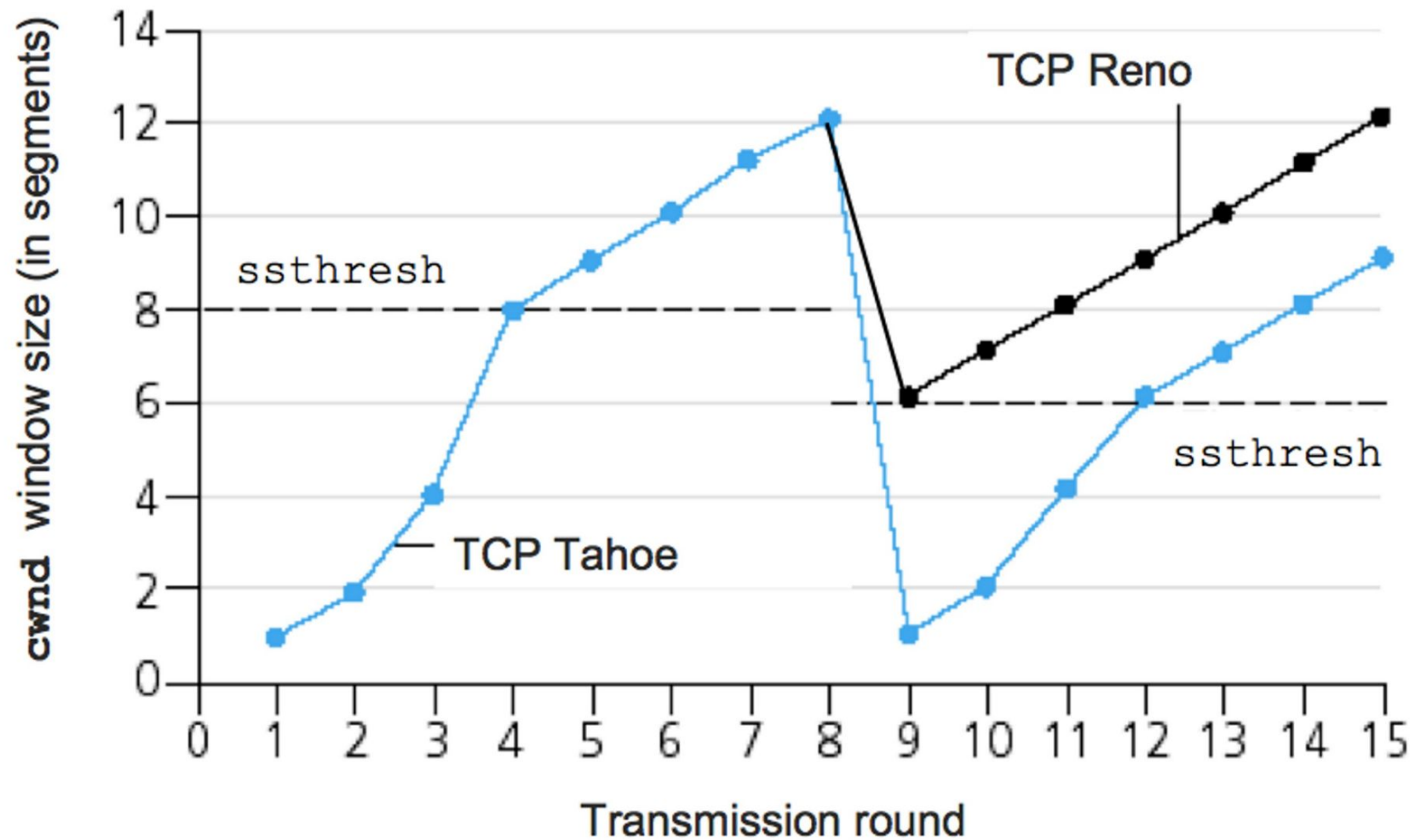
Quando un pacchetto viene perso, il destinatario continua a ricevere i pacchetti successivi. Non avendo ricevuto un pacchetto però, invia gli acknowledgment dei pacchetti successivi, all'interno dei quali richiede anche la ritrasmissione del pacchetto perso.

Nel momento in cui si hanno tre acknowledgment di questo tipo, allora la rete TCP capisce di trovarsi in una situazione di congestionamento di rete.





## TCP - Tahoe / Reno



## AIMD - Additive Increase Multiplicative Decrease

AIMD sta per "Additive Increase Multiplicative Decrease", ed è un algoritmo di controllo della congestione utilizzato da TCP per regolare la velocità di trasmissione dei dati sulla rete.

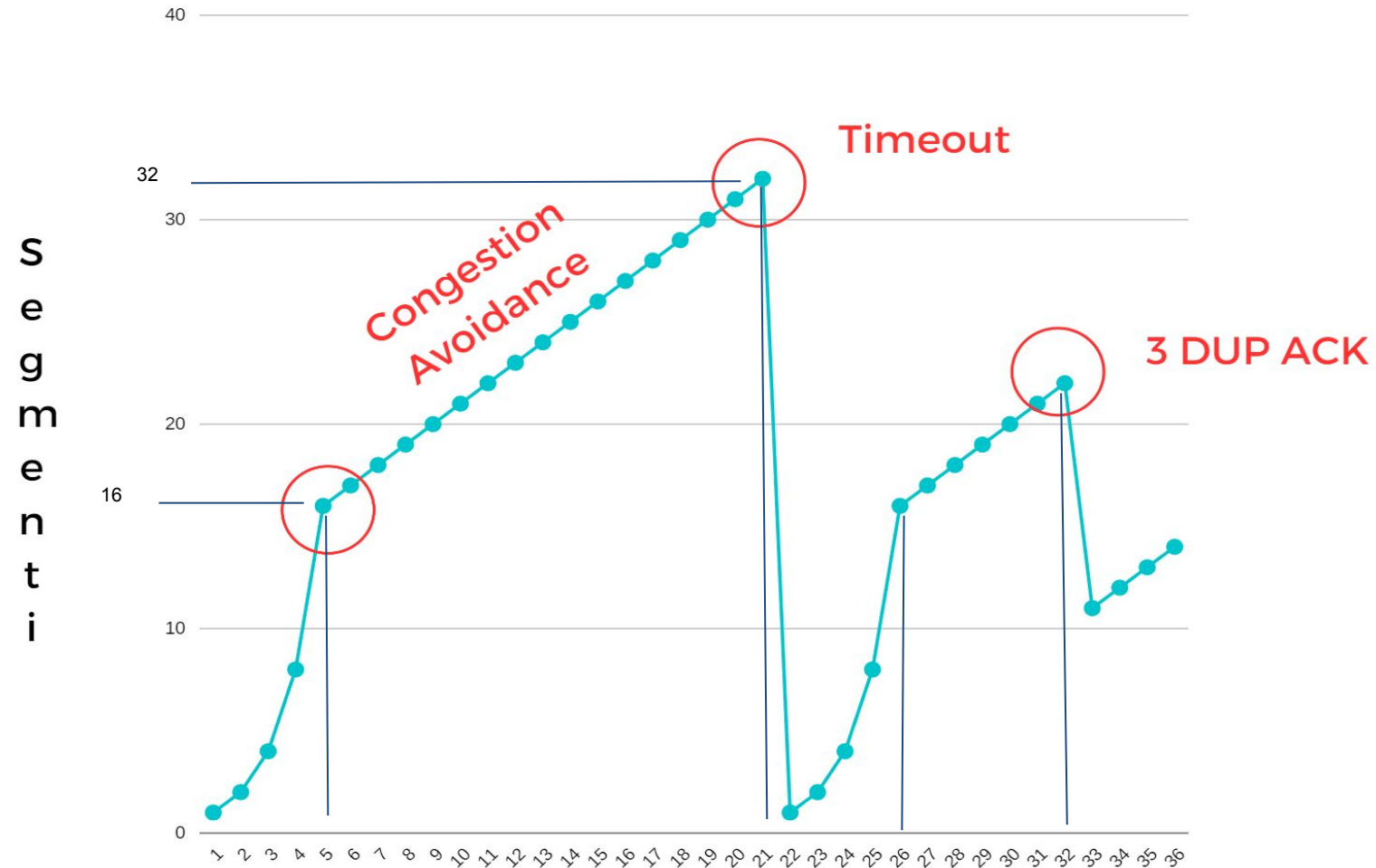
Durante la fase di congestion avoidance, AIMD aumenta la finestra di congestione in modo additivo quando non ci sono perdite di pacchetti e la diminuisce in modo moltiplicativo quando ci sono perdite di pacchetti.

L'incremento additivo avviene di solito con un incremento di uno alla finestra di congestione ad ogni round-trip-time, mentre la diminuzione moltiplicativa di solito riduce la finestra di congestione a metà quando si verifica una perdita di pacchetto.

La multiplicative decrease avviene unicamente nel TCP Reno.



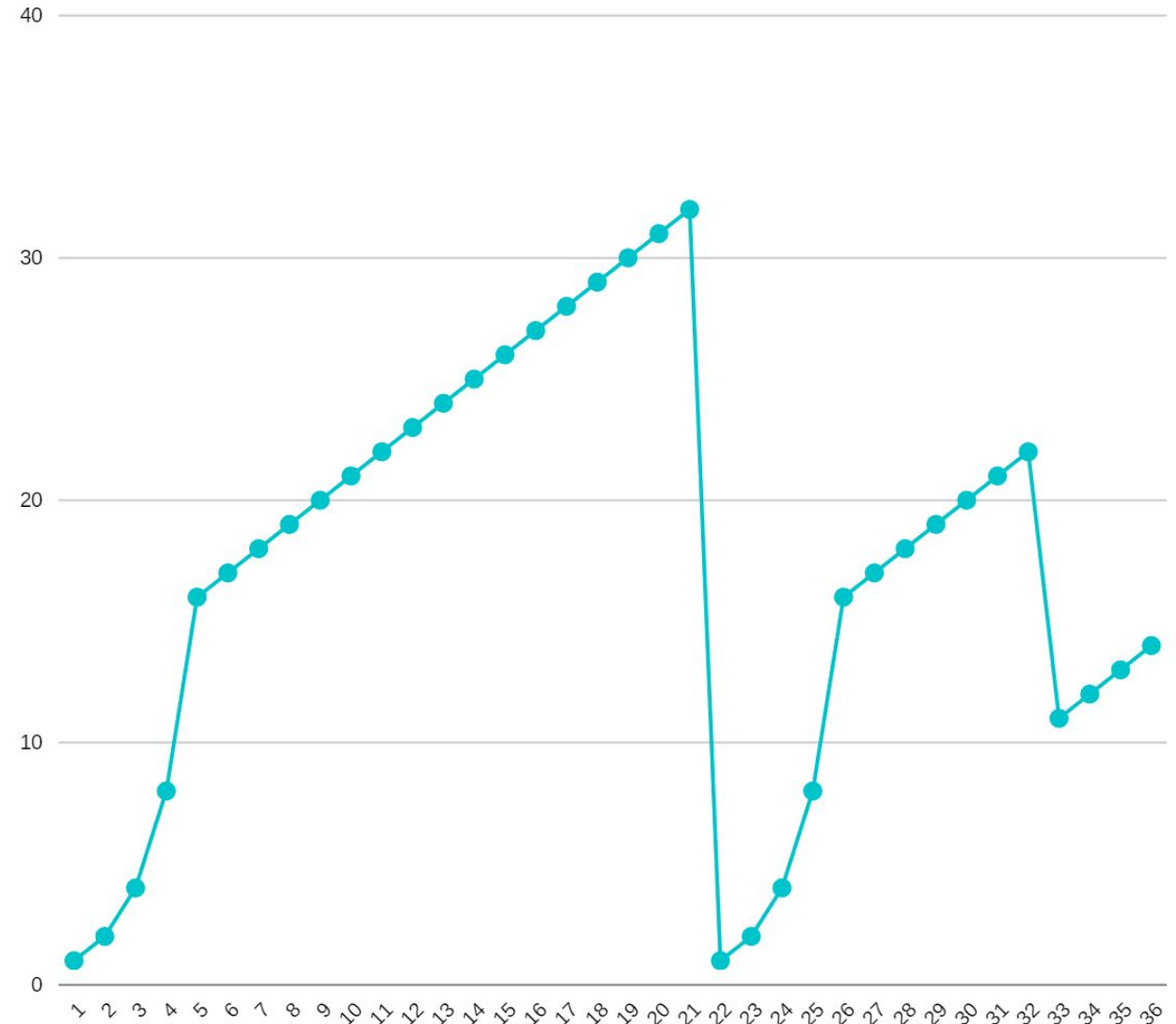
# Scenario



# Esercizio 1

Considerare la figura seguente che mostra l'andamento della finestra di congestione del protocollo TCP Reno durante l'invio di un file. Assumere che il RTT (round trip time) dei segmenti tra i due host sia fisso e uguale a 15 ms e che i segmenti inviati abbiano lunghezza fissa di 2400 bit. Rispondere alle seguenti domande:

- A. In quali intervalli opera lo slow start?
- B. In quali intervalli opera la congestion avoidance?
- C. Cosa accade al turno 21?
- D. Cosa accade al turno 32?
- E. Qual è la SST al primo turno di trasmissione?
- F. Qual è la SST al turno 22?
- G. Qual è la SST al turno 33?
- H. Calcolare la velocità massima e media del trasferimento



# Esercizio 1 - Soluzioni

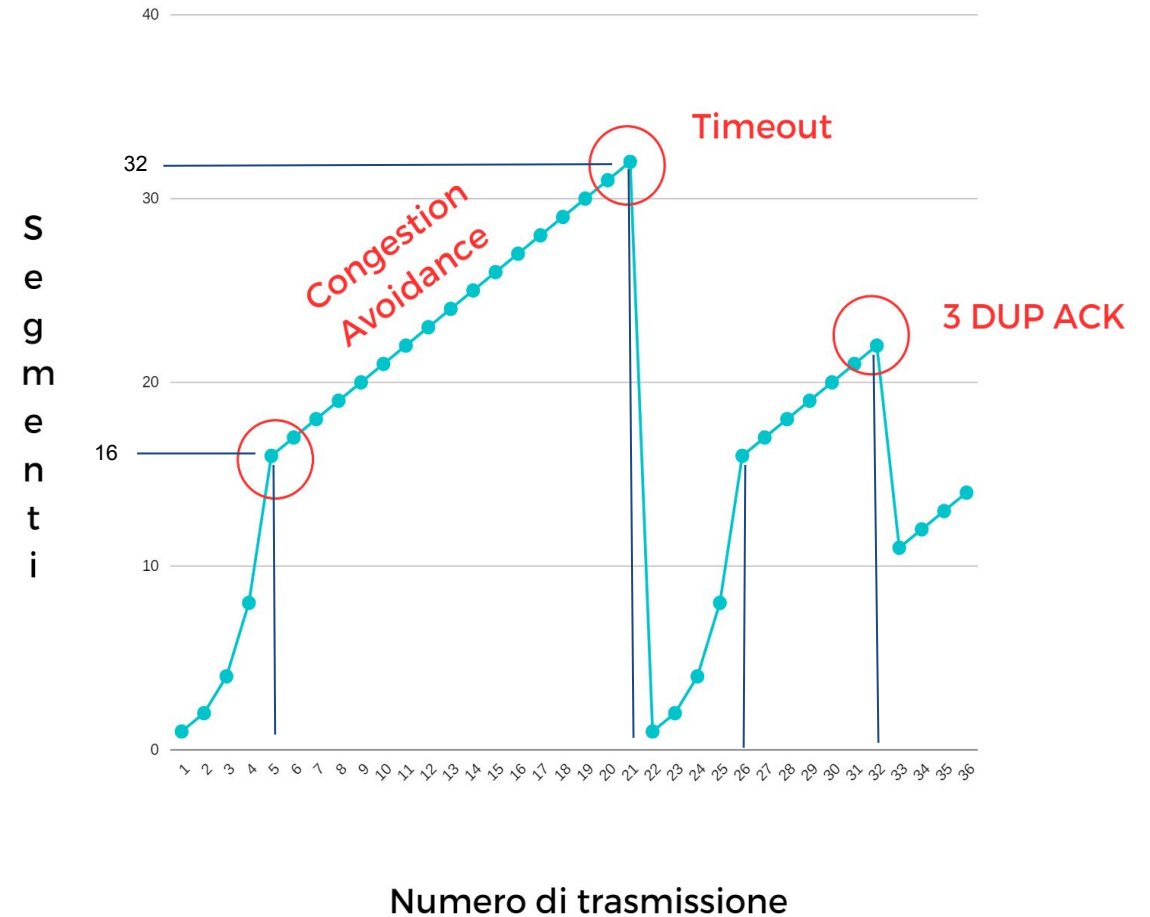
- A. [1,5], [22,26]
- B. [6,21], [27,32], [33,36]
- C. Timeout
- D. Triplo ack duplicato
- E. 16
- F. 16
- G. 11
- H. La velocità massima raggiunta viene calcolata come il numero massimo di segmenti trasmessi (trasmissione 21) in un solo RTT:

$$V_{max} = \frac{32 \times 2400}{15 \times 10^{-3}} = 5.12 \times 10^6 bps = 5.12 Mbps$$

La velocità media invece, la otteniamo facendo la somma dei dati inviati e dividendo per il tempo totale di trasmissione:

$$N = 1 + 2 + 4 + 8 + 16 + 17 + 18 + 19 + 20 + 21 + 22 + 23 + 24 + 25 + 26 + 27 + \dots + 14 = 621$$

$$V_{media} = \frac{N \times L}{RTT \times 36} = \frac{621 \times 2400}{15 \times 10^{-3} \times 36} = 4.14 \times 10^6 bps = 4.14 Mbps$$



## Esercizio 2

Assumere che in una sessione TCP Reno il RTT sia fisso a 15 ms, che i segmenti abbiano una lunghezza fissa di 2000 bit e che la banda massima disponibile tra i due host sia di 2 MBps.

- Calcolare la finestra di congestione massima
- Di quanti segmenti diminuisce la finestra di congestione se, una volta raggiunto il massimo, vengono ricevuti 3 ack duplicati?
- E nel caso di un timeout?

Assumere che il file inviato dal server sia composto da 50 segmenti e che la SST iniziale sia di 10 segmenti e che, la prima volta che vengono trasmessi, i pacchetti 4, 8, 22 e 48 vengono persi.

- Disegnare l'evoluzione della finestra di congestione e della SST
- Calcolare il tempo totale impiegato per trasferire il file



## Esercizio 2 - Soluzioni

A. Finestra di congestione massima (in bit)  $CW = RTT \cdot R = 15 \cdot 10^{-3} \cdot 2 \cdot 8 \cdot 10^6 \text{ bit}$   
 $= 240 \cdot 10^3 \text{ bit} = 240Kbit$

Finestra di congestione in segmenti  $= \frac{CW}{L} = \frac{240 \cdot 10^3}{2000} = 120 \text{ segmenti}$

B. 60 segmenti perché dimezzata

C. 119 segmenti perché torna a 1



## Esercizio 2 - Soluzioni

D.	Round	CW	SSTRESH	Pacchetti Trasmessi	Evento
	1	1	10	1	
	2	2	10	2, 3	
	3	4	10	<del>4</del> , 5, 6, 7	3DUPACK
	4	2	2	4, 5	
	5	3	2	6, 7, <del>8</del>	
	6	4	2	9, 10, 11, 12	3DUPACK
	7	2	2	8, 9	
	8	3	2	10, 11, 12	
	9	4	2	13, 14, 15, 16	
	10	5	2	17, 18, 19, 20, 21	
	11	6	2	<del>22</del> , 23, 24, 25, 26, 28	3DUPACK
	12	3	3	22, 23, 24	
	13	4	3	25 – 28	
	14	5	3	29 – 33	
	15	6	3	34 – 39	
	16	7	3	40 – 46	
	17	8	3	47, <del>48</del> , 49, 50	3DUPACK
	18	4	4	48, 49, 50	

se perdetes l'ultimo pacchetto di una serie:

- al turno successivo 3DUPACK
- due turni dopo rimandate

→ Timeout?





## Esercizio 3

Assumere che in una sessione TCP Reno il RTT sia fisso a 25 ms, che i segmenti abbiano una lunghezza fissa di 2500 bit e che la banda massima disponibile tra i due host sia di 10 MBps.

- Calcolare la finestra di congestione massima
- Di quanti segmenti diminuisce la finestra di congestione se, una volta raggiunto il massimo, vengono ricevuti 3 ack duplicati?
- E nel caso di un timeout?
- Calcolare l'ampiezza media della finestra di congestione e la velocità media di trasmissione
- Calcolare il tempo impiegato dalla connessione per raggiungere di nuovo il valore massimo di finestra dopo la perdita di un pacchetto

Assumere che il file inviato dal server sia composto da 50 segmenti, che la SST iniziale sia di 16 segmenti e che non vi siano perdite.

- Disegnare l'evoluzione della finestra di congestione e della SST
- Calcolare il tempo totale impiegato per trasferire il file



## Esercizio 3 - Soluzioni

A. Il valore della finestra di congestione massima in segmenti si calcola con la formula:

$$CW_{max} = \frac{R \cdot RTT}{L} = \frac{10 \cdot 8 \cdot 10^6 \cdot 25 \cdot 10^{-3}}{2500} = \frac{200 \cdot 10^4}{25 \cdot 10^2} = 8 \cdot 10^2 = 800 \text{ segmenti}$$

B. La finestra di congestione si dimezza, diminuisce quindi di 400 segmenti

C. La finestra di congestione torna a 1, diminuisce quindi di 799 segmenti

D. L'evoluzione della finestra è la seguente:

Round	CW	SST	Totale Pacchetti Trasmessi	
1	1	16	1	
2	2	16	2, 3	Slow Start
3	4	16	4, 5, 6, 7	
4	8	16	8 – 15	
5	16	16	16 – 31	
6	17	16	32 – 48	Congestion avoidance
7	18	16	49, 50	

E. L'invio del file richiede 7 round di trasmissione, ogni round richiede un tempo pari a un RTT, quindi il tempo totale sarà:

$$T = N \cdot RTT = 7 \cdot 25 \text{ ms} = 175 \text{ ms}$$



## Esercizio 4

Considerare due connessioni TCP, C1 e C2, che condividono un collegamento con banda pari a 560 kbps. Assumere che entrambe le connessioni abbiano un RTT di 100 ms e dimensione del segmento pari a 4 kbit. Supporre che C1 e C2 siano nella fase di congestion avoidance e che quando il tasso di invio dei dati eccede la finestra di congestione massima tutte le connessioni soffrono di perdita dei pacchetti.

- Disegnare l'andamento temporale delle finestre di congestione dei due collegamenti supponendo che al tempo iniziale entrambi abbiano una finestra di congestione pari a 5.
- A lungo termine, come sarà divisa la banda del collegamento?
- Disegnare l'andamento temporale delle finestre di congestione supponendo che la finestra di congestione iniziale di C1 sia 8 segmenti, mentre quella di C2 sia di 3 segmenti.
- A lungo termine, come sarà divisa la banda del collegamento nel secondo caso?



## Esercizio 4 - Soluzioni

Poiché entrambe le connessioni hanno lo stesso RTT, la finestra di congestione massima del collegamento è pari a:

$$CW_{max} = \frac{R \cdot RTT}{L} = \frac{560 \cdot 10^3 \cdot 100 \cdot 10^{-3}}{4000} = 14 \text{ segmenti}$$

Quando la somma delle finestre di C1 e C2 è minore o uguale alla finestra di congestione massima allora non c'è congestione, quando invece è superiore allora c'è congestione ed entrambe le connessioni perdono segmenti.

a) Per risolvere questo esercizio si può usare una tabella simile a questa:

Round	C.W. C <sub>1</sub>	C.W. C <sub>2</sub>	C.W. C <sub>1</sub> + C <sub>2</sub>	Evento
1	5	5	10	
2	6	6	12	
3	7	7	14	
4	8	8	16	3DUPACK
5	4	4	8	
6	5	5	10	
7	6	6	12	
8	7	7	14	
9	8	8	16	3DUPACK
10	4	4	8	

da qui in poi si ripete all'infinito



## Esercizio 4 - Soluzioni

b) A lungo termine, la banda viene suddivisa equamente (metà e metà) tra le due connessioni

c) Per risolvere questo esercizio si può usare una tabella simile a questa:

Round	C.W. $C_1$	C.W. $C_2$	C.W. $C_1 + C_2$	Evento
1	8	3	11	3DUPACK
2	9	4	13	
3	10	5	15	
4	5	2	7	
5	6	3	9	
6	7	4	11	3DUPACK
7	8	5	13	
8	9	6	15	
9	4	3	7	
10	5	4	9	
11	6	5	11	3DUPACK
12	7	6	13	
13	8	7	15	
14	4	3	7	

da qui in poi si ripete all'infinito

d) Anche qui, a lungo termine, la banda viene suddivisa equamente (metà e metà) tra le due connessioni



## Esercizio 5

Considerare una modifica all'algoritmo di controllo di congestione TCP in cui, al posto di un incremento additivo viene utilizzato un incremento moltiplicativo: la finestra di congestione viene moltiplicata per due. Assumendo le stesse ipotesi dell'esercizio precedente:

- Disegnare l'andamento temporale delle finestre di congestione dei due collegamenti supponendo che al tempo iniziale C1 ha una finestra di 8 e C2 di 3.
- A lungo termine, come viene suddivisa la banda del collegamento?
- Cosa succede se sia l'incremento che il decremento sono additivi?
- A lungo termine, come viene suddivisa la banda del collegamento?



## Esercizio 5 - Soluzioni

- a) Per risolvere questo esercizio si può usare una tabella simile a questa:

Round	C.W. $C_1$	C.W. $C_2$	C.W. $C_1 + C_2$	Evento
1	8	3	11	3 DUPACK
2	16	6	24	
3	8	3	15	
4	16	6	24	3 DUPACK

da qui in poi si ripete all'infinito

- b) A lungo termine la banda è suddivisa in modo *non equo* tra le due connessioni:  $C_1$  riceve sempre più banda di  $C_2$

- c) Per risolvere questo esercizio si può usare una tabella simile a questa:

Round	C.W. $C_1$	C.W. $C_2$	C.W. $C_1 + C_2$	Evento
1	8	3	11	3DUPACK
2	9	4	13	
3	10	5	15	
4	9	4	13	
5	10	5	15	3DUPACK

da qui in poi si ripete all'infinito

- d) Anche in questo caso la suddivisione è non equa:  $C_1$  riceve sempre più banda di  $C_2$





## Riassumendo

In caso di più connessioni parallele, se l'algoritmo di incremento/decremento della finestra di congestione è:

- AIMD allora la divisione della banda è equa → due connessioni hanno una finestra di congestione iniziale diversa e subiscono una perdita di pacchetto, la connessione con la finestra di congestione più grande vedrà una diminuzione maggiore nella sua finestra di congestione rispetto all'altra connessione
- Con incremento e decremento moltiplicativo, allora la divisione non è equa → la connessione con finestra di congestione più grande viene aumentata di più
- Con incremento e decremento additivo, allora la suddivisione della banda non è equa → la connessione con finestra di congestione più grande viene decrementata in maniera lineare





## Esercizio 6

Assumere che in una sessione TCP Reno il RTT sia fisso a 150 ms, che i segmenti abbiano una lunghezza fissa di 1875 byte e che la banda massima disponibile tra i due host sia di 6 Mbps.

- Calcolare la finestra di congestione massima
- Calcolare il tempo impiegato dalla connessione per raggiungere di nuovo il valore massimo della finestra dopo un 3 ack duplicati

Assumere che il file inviato dal server sia composto da 30 segmenti e che la SST iniziale sia di 4 segmenti e che vi siano perdite nel caso dei pacchetti 18 e 23.

- Disegnare l'evoluzione della finestra di congestione e della SST



## Esercizio 6 - Svolgimento

A)

$$CW_{max} = \frac{RTT \times R}{L} = \frac{6 \times 10^6 \times 1.5 * 10^{-1}}{1875 \times 8} = 60segmenti$$

B)

$$30 \times RTT = 4500ms = 4.5s$$



## Esercizio 6 - Svolgimento

C)

Round	CW	SST	Pacchetti
1	1	4	1
2	2	4	2,3
3	4	4	4,5,6,7
4	5	4	8,9,10,11,12
5	6	4	13,14,15,16,17,18
6	7	4	19,20,21,22,23,24,25
7	3	3	18,19,20
8	4	3	21,22,23,24
9	5	3	25,26,27,28,29
10	6	3	30

→ 3 DUP ACK





ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

Campus di Bologna  
Corso di Laurea in Informatica per il management

E-mail [ulderico.vagnoni2@unibo.it](mailto:ulderico.vagnoni2@unibo.it)

[www.unibo.it](http://www.unibo.it)