



# Esercitazione livello Trasporto

Gaia Maselli

# Esercizio 1

- In una rete con un valore fisso  $m > 1$ , è possibile utilizzare entrambi i meccanismi Go-Back-N e Selective-Repeat.
  1. Indicare i vantaggi e gli svantaggi dell'impiego di ciascuno di essi.
  2. Quali altre considerazioni si devono fare per decidere quale meccanismo utilizzare?

## Vantaggi e svantaggi:

- Il vantaggio di utilizzare il protocollo Go-Back- $N$  è che si può avere una maggiore finestra di invio. Siamo in grado di inviare più pacchetti prima di attendere il loro riscontro. Lo svantaggio di questo protocollo è che la dimensione della finestra di ricezione è solo 1. Il ricevitore non può accettare e memorizzare i pacchetti ricevuti fuori sequenza, che vengono perciò scartati. Scartare tali pacchetti significa che il mittente deve rispediti, con conseguente congestione della rete e riduzione della capacità della condotta (quantità di pacchetti che possono essere inviati senza riscontro). In questo modo il vantaggio di aver una finestra di invio maggiore può essere vanificato riempiendo la rete con pacchetti rispediti.
- Il vantaggio di utilizzare il protocollo Selective-Repeat è che la finestra di ricezione può essere molto più grande di 1. Questo consente di memorizzare i pacchetti ricevuti fuori sequenza ed evita che questi vengano inviati nuovamente, congestionando la rete. D'altra parte, la dimensione della finestra di trasmissione di questo protocollo è la metà del Go-Back- $N$ , il che significa che possiamo inviare meno pacchetti prima di attendere il riscontro.

## Considerazioni:

Se il prodotto banda-ritardo della rete è grande, l'affidabilità è buona e il ritardo è basso, si dovrebbe scegliere il protocollo Go-Back- $N$  per utilizzare più della capacità della rete. D'altra parte, se il prodotto banda-ritardo è piccolo, la rete non è molto affidabile, o la rete introduce lunghi ritardi, abbiamo bisogno di usare il Selective-Repeat.

# Esercizio 2

- Un server TCP ha ricevuto e riscontrato all'interno di una connessione i byte fino al 4000. Dire quale azione esegue il server dopo i seguenti eventi:
  1. Il server riceve un segmento di 1000 byte con numero di sequenza pari a 3001
  2. In seguito all'evento 1 il server riceve un segmento di 1000 byte con numero di sequenza pari a 6001
  3. In seguito all'evento 2 il server riceve un segmento di 1000 byte con numero di sequenza pari a 5001
  4. In seguito all'evento 3 il server riceve un segmento di 1000 byte con numero di sequenza pari a 4001

## Soluzione:

Il server ha riscontrato fino al byte 4000, ovvero ha inviato ack 4001. Il server risponde nel seguente modo:

- **Ack duplicato 4001**
- **Ack duplicato 4001**
- **Ack duplicato 4001**
- **Ack cumulativo 7001**

# Esercizio 3

- Un server TCP ha ricevuto e riscontrato all'interno di una connessione i byte fino al 4000. Dire quale azione esegue il server dopo i seguenti eventi:
  1. Il server riceve un segmento di 1000 byte con numero di sequenza pari a 5001
  2. In seguito all'evento 1 il server riceve un segmento di 1000 byte con numero di sequenza pari a 4001
  3. In seguito all'evento 2 il server riceve un segmento di 1000 byte con numero di sequenza pari a 6001
  4. In seguito all'evento 3 il server riceve un segmento di 1000 byte con numero di sequenza pari a 7001

## Soluzione:

Il server ha riscontrato fino al byte 4000, ovvero ha inviato ack 4001. Il server risponde nel seguente modo:

- **Ack duplicato 4001**
- **Ack cumulativo 6001**
- **Ack posticipato 7001**
- **Se arriva entro i 500 ms allora Ack cumulativo 8001, altrimenti Ack posticipato di 8001**

# Esercizio 4

- Nel protocollo TCP la finestra di invio può essere più piccola, più grande o della stessa dimensione della finestra di ricezione?



## Soluzione:

La finestra di invio nel TCP inizialmente ha la stessa dimensione della finestra di ricezione, ma con il crearsi della congestione in rete, può diventare più piccola della finestra di ricezione. Non può mai diventare più grande della finestra di ricezione. La finestra di invio diventa più piccola della dimensione originale a causa della congestione o perché determinata dal ricevente.

# Esercizio 5

- L'utente A utilizza il proprio browser per aprire due connessioni con il server HTTP in esecuzione sull'host B. Come può il protocollo TCP distinguere queste due connessioni?

## Soluzione:

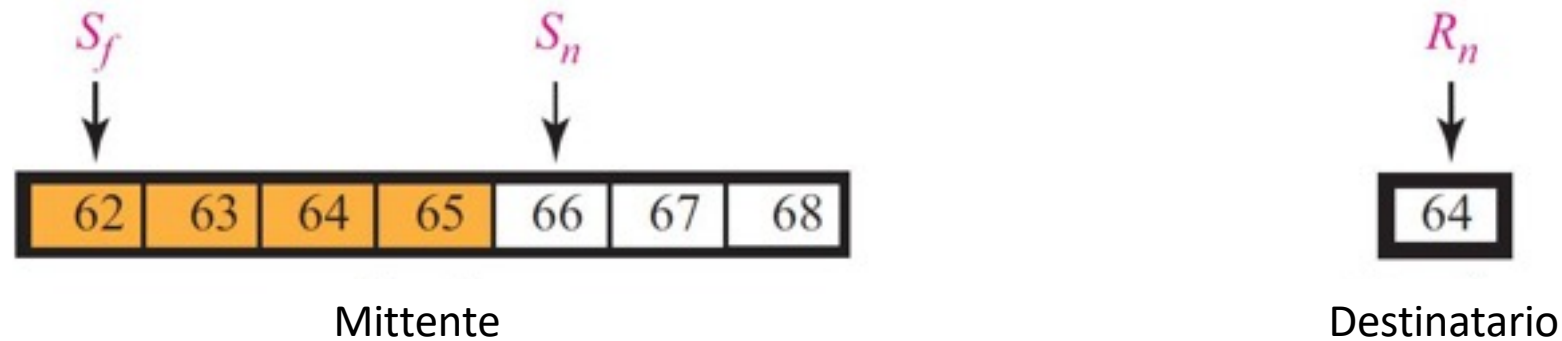
Una connessione è caratterizzata da una coppia di socket address, uno per ciascuna estremità.

Sebbene i socket address dal lato di B siano gli stessi in questo caso, i socket address dal lato di A sono diversi. Ogni socket address dal lato di A ha un diverso numero di porta effimero.

## Esercizio 6 (FATTO)

- In una rete basata su Go-Back-N con  $m=3$  e dimensione della finestra di invio uguale a 7, i valori delle variabili sono:  $S_f=62$ ,  $S_n=66$  e  $R_n=64$ . Si ipotizzi che la rete non duplichi e non alteri l'ordine dei pacchetti.
  1. Qual è il numero di sequenza dei pacchetti in transito?
  2. Qual è il numero di riscontro dei pacchetti ack in transito?

## Soluzione:



1. Se il destinatario si aspetta un pacchetto con numero di sequenza 64 e i pacchetti con numeri di sequenza 62 e 65 sono già stati spediti ma non ancora riscontrati, significa che i due pacchetti con numeri di sequenza 64 e 65 sono in transito dal mittente al destinatario.
2. Se il mittente aspetta un riscontro per il pacchetto 62, ma il valore  $R_n = 64$ , significa che i pacchetti ACK con numeri di riscontro 62 e 63 sono in transito dal destinatario al mittente.

# Esercizio 7

- Si può definire il prodotto rate-ritardo come il numero di pacchetti che possono essere in transito nella rete durante un tempo pari a RTT. Calcolare il prodotto rate-ritardo nel caso in cui:
- Rate = 1Mbps
- RTT = 20ms
- Dimensione dei pacchetti = 1000 bit

## Soluzione:

Prima definiamo il prodotto rate-ritardo (BDP) in bit e poi troviamo il numero di pacchetti:

$$\text{BDP} = 1 \text{ Mbps} * 20 \text{ ms} = 20000 \text{ bit}$$

$$20000 \text{ bit} / 1000 \text{ bit} = 20 \text{ pacchetti}$$

# Esercizio 8

- Si deve progettare un protocollo a finestra scorrevole con meccanismo Selective-Repeat per una rete nella quale il rate=1Gbps e la distanza media fra mittente e destinatario e' 5000km. La dimensione media dei pacchetti e' di 50000 bit e la velocità di propagazione nel mezzo trasmissivo e'  $2 \times 10^8$  m/s.
- Calcolare la dimensione massima delle finestre di invio e di ricezione, il numero di bit (m) nel campo numero di sequenza e un valore appropriato per il timeout del timer.



Calcoliamo il round-trip-time (RTT) medio e il numero di pacchetti nella condotta (pipeline) prima di trovare le dimensioni delle finestre, il valore  $m$ , e il timeout.

1.  $\text{RTT medio} = 2 * (5,000 \text{ Km}) / (2 * 10^8) = 50 \text{ ms}.$
2.  $\text{Prodotto rate-ritardo} = 1\text{Gbps} * 50 \text{ ms} = 50.000.000 \text{ bit}.$
3.  $\text{Prodotto rate-ritardo} = 50.000.000 \text{ bit} / 50.000 \text{ bit} = 1000 \text{ pacchetti}.$
4. La massima dimensione della finestra di invio dovrebbe essere 1000 per non avere più di 1000 pacchetti inviati senza riscontro (nella condotta).
5. La massima dimensione della finestra di ricezione dovrebbe essere di 1000 pacchetti
6. Sappiamo che  $(\text{finestra d'invio}) \leq (2^{m-1})$  ovvero  $1000 \leq (2^{m-1})$ . Questo significa che dobbiamo scegliere  $(m-1)$  almeno uguale a 10, o  $m=11$ . I numeri di sequenza vanno quindi da 0 a 2047.

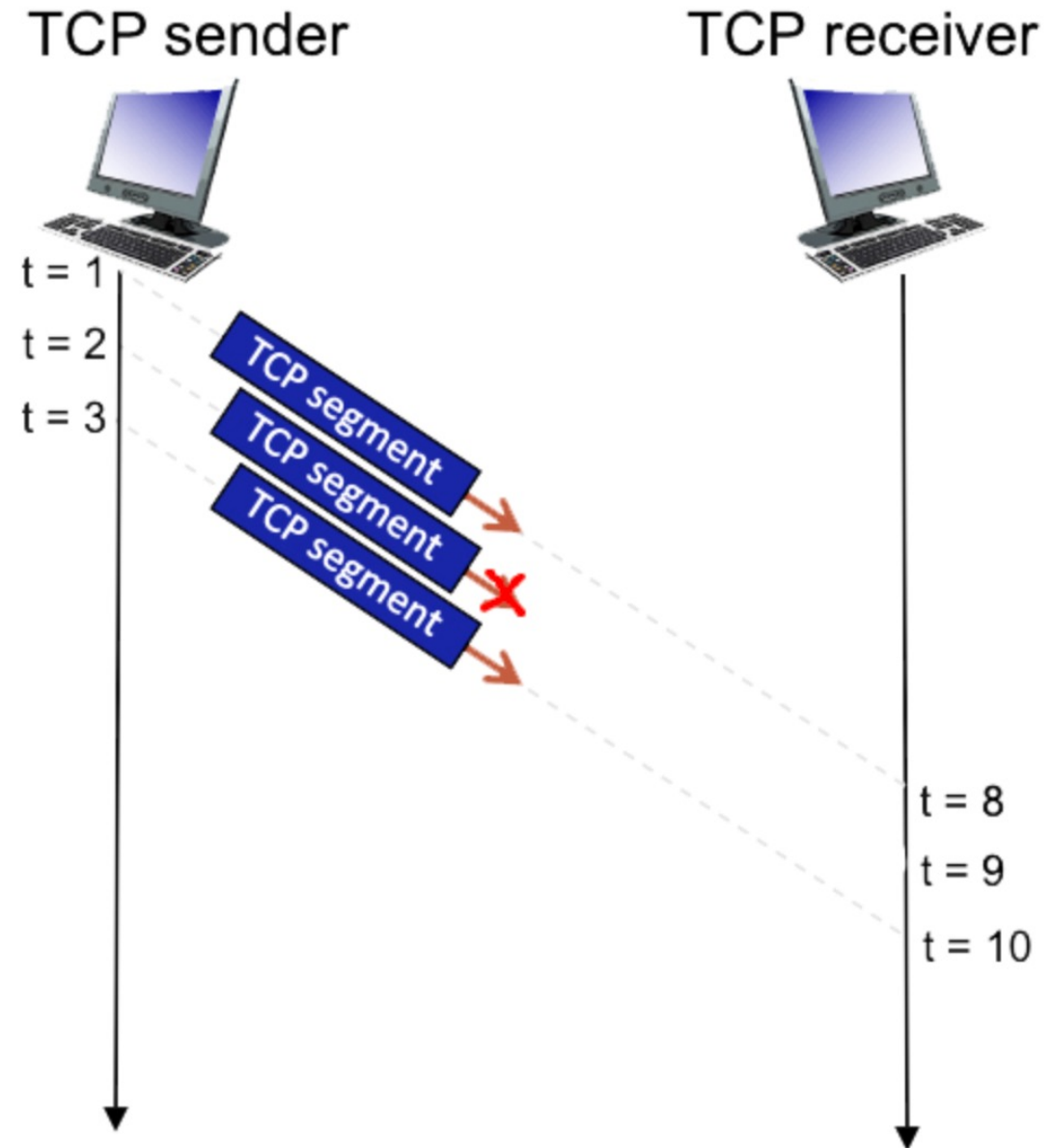
# Esercizio 9

Si consideri la figura, in cui un mittente e un destinatario TCP comunicano su una connessione in cui i segmenti inviati dal mittente al destinatario possono andare persi.

Il mittente TCP invia una finestra iniziale di 3 segmenti. Si supponga che il valore iniziale del numero di sequenza dal mittente al destinatario sia 123 e che ciascuno dei primi 3 segmenti contenga 897 byte. Il ritardo tra il mittente e il destinatario è di 7 unità di tempo, quindi il primo segmento arriva al destinatario al tempo  $t = 8$ .

Come mostrato nella, 1 dei 3 segmenti viene perso tra il mittente e il destinatario.

1. Indica i numeri di sequenza associati a ciascuno dei 3 segmenti inviati dal mittente.
2. Indica i numeri di ACK che il destinatario invia in risposta a ciascuno dei segmenti.



## 1. Numeri di sequenza

Il numero di sequenza del primo segmento è **123**. Ogni segmento contiene 897 byte, quindi:

Segmento 1: 123

Segmento 2:  $123 + 897 = 1020$

Segmento 3:  $1020 + 897 = 1917$

## 2. Numeri di ACK

Il primo segmento (Seq=123) arriva correttamente →  $ACK = 123 + 897 = 1020$

Il secondo segmento non arriva

Il terzo segmento arriva, ma il destinatario non ha ricevuto il secondo → continua a mandare  $ACK = 1020$  (duplicato)

# Esercizio 10

Si supponga che i valori attuali stimati da TCP siano:

- **EstimatedRTT** = 230 millisecondi
- **DevRTT** = 40 millisecondi

I successivi **tre valori misurati** del round-trip time (RTT) sono:

1. 220 ms
2. 220 ms
3. 390 ms

Per ciascuno dei tre RTT misurati, calcolare:

Il nuovo valore di EstimatedRTT

Il nuovo Timeout TCP

considerando  $\alpha=0.125$  e  $\beta=0.25$

# formule

$$\text{EstimatedRTT}_{\text{new}} = (1 - \alpha) * \text{EstimatedRTT}_{\text{old}} + \alpha * \text{SampleRTT}$$

$$\text{DevRTT}_{\text{new}} = (1 - \beta) * \text{DevRTT}_{\text{old}} + \beta * |\text{SampleRTT} - \text{EstimatedRTT}_{\text{old}}|$$

$$\text{Timeout} = \text{EstimatedRTT}_{\text{new}} + 4 * \text{DevRTT}_{\text{new}}$$

# Soluzione

$$\begin{aligned}\text{EstimatedRTT} &= (1 - 0.125) * 230 + 0.125 * 220 \\ &= 0.875 * 230 + 0.125 * 220 \\ &= 201.25 + 27.50 \\ &= 228.75 \text{ ms}\end{aligned}$$

$$\begin{aligned}\text{DevRTT} &= (1 - 0.25) * 40 + 0.25 * |220 - 230| \\ &= 0.75 * 40 + 0.25 * 10 \\ &= 30.00 + 2.50 \\ &= 32.50 \text{ ms}\end{aligned}$$

$$\begin{aligned}\text{Timeout} &= \text{EstimatedRTT} + 4 * \text{DevRTT} \\ &= 228.75 + 4 * 32.50 \\ &= 228.75 + 130.00 \\ &= 358.75 \text{ ms}\end{aligned}$$

$$\begin{aligned}\text{EstimatedRTT} &= (1 - 0.125) * 228.75 + 0.125 * 220 \\ &= 0.875 * 228.75 + 0.125 * 220 \\ &= 200.16 + 27.50 \\ &= 227.66 \text{ ms}\end{aligned}$$

$$\begin{aligned}\text{DevRTT} &= (1 - 0.25) * 32.50 + 0.25 * |220 - 228.75| \\ &= 0.75 * 32.50 + 0.25 * 8.75 \\ &= 24.38 + 2.19 \\ &= 26.56 \text{ ms}\end{aligned}$$

$$\begin{aligned}\text{Timeout} &= \text{EstimatedRTT} + 4 * \text{DevRTT} \\ &= 227.66 + 4 * 26.56 \\ &= 227.66 + 106.25 \\ &= 333.91 \text{ ms}\end{aligned}$$

$$\begin{aligned}\text{EstimatedRTT} &= (1 - 0.125) * 227.66 + 0.125 * 390 \\ &= 0.875 * 227.66 + 0.125 * 390 \\ &= 199.20 + 48.75 \\ &= 247.95 \text{ ms}\end{aligned}$$

$$\begin{aligned}\text{DevRTT} &= (1 - 0.25) * 26.56 + 0.25 * |390 - 227.66| \\ &= 0.75 * 26.56 + 0.25 * 162.34 \\ &= 19.92 + 40.59 \\ &= 60.51 \text{ ms}\end{aligned}$$

$$\begin{aligned}\text{Timeout} &= \text{EstimatedRTT} + 4 * \text{DevRTT} \\ &= 247.95 + 4 * 60.51 \\ &= 247.95 + 242.04 \\ &= 489.99 \text{ ms}\end{aligned}$$



# Esercizio 11

Si consideri la figura che rappresenta l'evoluzione della finestra di congestione di TCP all'inizio di ogni unità di tempo (dove l'unità di tempo è uguale al RTT). TCP invia un «gruppo» di pacchetti di dimensioni  $cwnd$  all'inizio di ogni unità di tempo. Il risultato dell'invio di quel gruppo di pacchetti è che o tutti i pacchetti vengono confermati alla fine dell'unità di tempo, oppure c'è un timeout per il primo pacchetto, oppure c'è un triplo ACK duplicato per il primo pacchetto. In questo problema, viene chiesto di ricostruire la sequenza degli eventi (ACK, perdite) che hanno portato all'evoluzione di  $cwnd$  di TCP mostrata di seguito.

