

# **Programmazione di Reti**

## *Formulario*

Luca Casadei - Francesco Pazzaglia - Martin Tomassi

21 maggio 2024

# Indice

<b>1</b>	<b>Ritardi di trasferimento</b>	<b>2</b>
1.1	Tempo di trasmissione . . . . .	2
1.2	Tempo di propagazione . . . . .	2
1.3	Tempo totale . . . . .	2
1.4	Quantità di bit presenti sul canale . . . . .	3
1.5	Scenari Cut-Through e Store & Forward . . . . .	3
1.5.1	Esempio con Store & Forward . . . . .	3
1.5.2	Esempio con Cut-Through . . . . .	3
<b>2</b>	<b>Round-Trip-Time (RTT)</b>	<b>4</b>
2.1	Capacità di propagazione . . . . .	4
2.2	RTT . . . . .	4
<b>3</b>	<b>Efficienza protocolli "ARQ"</b>	<b>5</b>
3.1	Go-back-N . . . . .	5
3.2	Stop & Wait . . . . .	5
<b>4</b>	<b>Livello Applicazione</b>	<b>6</b>
4.1	Differenza di SAP per i protocolli TCP-UDP . . . . .	6
4.2	HTTP . . . . .	6
4.2.1	Connessioni parallele e persistenti . . . . .	6
4.3	Distribuzione Client-server e Peer-to-peer . . . . .	6

# Capitolo 1

## Ritardi di trasferimento

### 1.1 Tempo di trasmissione

- $T_{trasmissione}$  = Tempo di trasmissione ( $s$ )
- $L$  = Lunghezza del pacchetto ( $bit$ )
- $R$  = Frequenza (capacità) di trasmissione (bit-rate) ( $\frac{bit}{s}$ )

Trasferimento di un pacchetto da router **A** a router **B**

$$T_{trasmissione} = \frac{L}{R}$$

### 1.2 Tempo di propagazione

- $D$  = Lunghezza del collegamento ( $m$ )
- $v$  = Velocità (ritardo) di propagazione ( $\frac{m}{s}$ )
- $\tau$  = Tempo di propagazione ( $s$ )

Si può ricavare il tempo di propagazione:

$$\tau = \frac{D}{v}$$

Nel caso della suddivisione del canale in  $n$  sotto-canali e considerando la lunghezza del canale totale  $D$ :

$$\tau_n = \frac{\tau}{n}$$

### 1.3 Tempo totale

Si ricava da:

$$T_{tot} = \tau + T_{trasmissione} + T_{accodamento} + T_{elaborazione}$$

## 1.4 Quantità di bit presenti sul canale

Si ricava attraverso:

$$L = R * T_{propagazione}$$

## 1.5 Scenari Cut-Through e Store & Forward

### 1.5.1 Esempio con Store & Forward

Consideriamo  $n$  elementi trasmissivi, avremmo  $n$  tempi di trasmissione:

$$n * T_{trasmissione}$$

Consideriamo ora  $k$  elementi che introducono latenza per accodamento e trasmissione, otteniamo:

$$k * T_{accodamento}$$

Con tempo di propagazione fisico  $\tau$

$$T_{totale} = n * T_{trasmissione} + k * T_{accodamento} + \tau$$

### 1.5.2 Esempio con Cut-Through

In questo caso si considera il tempo di accodamento del solo header e non di tutto il pacchetto, sapendo che per trasmettere un pacchetto trascurando eventuali tempi di elaborazione è:  $T_H + (T - T_H)$ , da questo si ottiene che con header  $H$ :

$$T_{totale} = n * T + k * T_H + \tau$$

## Capitolo 2

# Round-Trip-Time (RTT)

### 2.1 Capacità di propagazione

- $C = R$  come nella sezione precedente 1.1, è la capacità o frequenza di trasmissione.

### 2.2 RTT

- $T_{ack}$  = Tempo acknowledge.

Equivale al  $T_{trasmissione}$  più tutti i tempi di elaborazione e di accodamento  $T_{accodamento}$ , e anche dei tempi di propagazione  $\tau$ , come nella sezione precedente 1.1, ma considerando tutta la tratta da percorrere sia per l'andata che per il ritorno (in genere si ha  $2\tau$  perché va considerata la propagazione di invio e ricezione.).

$$RTT = T_{trasmissione} + 2\tau + T_{ack}$$

## Capitolo 3

# Efficienza protocolli "ARQ"

- MSS = Maximum segment size (equivalente a L) 1.1.
- SSTRESH = Segment size threshold (soglia).
- CWND = Congestion window.
- RCWND = Receiver congestion window.
- $T_{trasmissione} = \frac{MSS}{R}$

Dimensione del file trasferibile nella window (unità di misura MSS) =  $\frac{L}{CWND}$ , considerando che la conversione in MSS si ottiene facendo:  $\frac{RCWND}{CWND}$  e la conversione in MSS del threshold:  $\frac{SSTHRESH}{CWND}$ , in generale, si ha  $\frac{L(\text{byte})}{CWND(\text{byte})} = MSS$ .

### 3.1 Go-back-N

- $W$  Numero di pacchetti della window (dimensione).
- $T$  Tempo per pacchetto.

Dipende dal rapporto tra RTT e la lunghezza della finestra  $NT$ , consideriamo il tempo di trasmissione  $T_{trasmissione}$  e il tempo di propagazione  $\tau$ :

Dobbiamo avere  $WT \geq RTT \implies$  sviluppando i calcoli si ottiene  $W > x$  dove  $x$  è il numero di pacchetti da mandare prima dell'acknowledge per avere un'efficienza 1.

La trasmissione è continua quando  $WT \geq RTT$   
(discontinua quando  $WT < RTT$ )

### 3.2 Stop & Wait

- Efficienza  $\eta = \frac{T_{trasmissione}}{T_{trasmissione} + T_{propagazione}}$

## Capitolo 4

# Livello Applicazione

### 4.1 Differenza di SAP per i protocolli TCP-UDP

Nel server UDP non esiste un socket di benvenuto, mentre in un socket TCP sono richiesti due socket per poter effettuare la connessione, per supportare quindi  $n$  connessioni simultanee, sono richiesti  $n + 1$  socket.

### 4.2 HTTP

- $\Delta$  = Tempo medio richiesto per inviare un oggetto sul collegamento. (In genere questo è riconducibile a 1.1)
- $\beta$  = Frequenza di arrivo di oggetti al collegamento di accesso.
- $\Delta\beta$  = Intensità del traffico. (Può essere ridotta gestendo l'elaborazione in esterno, su altre reti).

Consideriamo il ritardo medio di accesso come:  $\frac{\Delta}{(1-\Delta\beta)}$

#### 4.2.1 Connessioni parallele e persistenti

Quando si trattano  $N$  connessioni parallele, il canale si divide in  $\frac{1}{N}$  parti con capacità uguali, la differenza per le connessioni persistenti consiste nella possibilità di richiedere oggetti su una connessione già stabilita senza dover rieseguire la three-way-handshake (*syn*, *ack*, *syn-ack*).

### 4.3 Distribuzione Client-server e Peer-to-peer

- $N$  = Numero di peer.
- $D_{cs}$  = Tempo minimo di distribuzione (tempo necessario per far sì che tutti gli  $N$  peer ricevano una copia del file) per un servizio client-server.
- $D_{p2p}$  = Come il precedente ma per un servizio peer-to-peer.
- $F$  = Dimensione del file.

- $u_s$  = Velocità di upload del server.

Si ottiene:  $D_{cs} = \max \left\{ \frac{NF}{u_s}, \frac{F}{d_{\min}} \right\}$

Si ottiene:  $D_{p2p} = \max \left\{ \frac{F}{u_s}, \frac{F}{d_{\min}}, \frac{NF}{(u_s + \sum_{i=1}^N (u_i))} \right\}$