Multimedia esercizi

Silviu Filote

June 7, 2021

Contents

1 TDM

- Tutto dev'essere espress in bit

$$1\ byte\ = 8\ bit$$

- Induvidiare la fonte con **minor velocità di trasmissione** $\rightarrow \alpha$
- "Let's assume that multiplexing is performed at the level of a singol octet (1 byte = 8 bit)" $\Rightarrow$ **dimensione di uno slot** $\rightarrow \beta$
- **Frame:** se il multiplexer funziona a 8 bit, la frame si compone sa pendo che la seorgente a capacità pià piccola α genera 8 *bit*. Esempio:
  - Sorgerte 1: 64 *kbit/s* $\rightarrow$ 1 *byte* (α)
  - Sorgerte 2: 128 *kbit/s* $\rightarrow$ 2 *byte*
  - Sorgerte 3: 640 *kbit/s* $\rightarrow$ 10 *byte*
- **Frame duration:** è la durata dell'intera frame e non del singolo slot, perchè il multiplex is multiplexing i segnali, ossia nello stesso istante in cui una sorgente termina, terminano anche le altre. Come se il canale fosse diviso in 3 parti.

$$FD = \frac{\beta}{\alpha}$$

- **Transmission rate of the multiplexer:**

$$W = \sum_{1}^{n} velocità\ transmissione\ sorgente_i \qquad con\ i = 0, .., n$$

se le sorgenti hanno tutte la stessa velocità allora:

$$W = n \cdot velocià\ sorgente$$

la velocità di ogni slot è pari a α

$$W = \frac{\#\ bit\ frame}{FD} = \frac{\#\ bit\ slot}{TS}$$

questo si può fare perchè è tutto proporzionale

- **Time slot**

$$TS = \frac{\beta}{W} \qquad TS = \frac{FD}{N} \quad N\ slot$$

2 TDMA

- **Viene aggiunto un T_g ad ogni slot**
- "System for 100 telephone calls" $\rightarrow$ 100 *canali*
- Velocità della luce $\rightarrow$ 300.000 *km/s*
- System efficiency (rispetto alla transmissione dei dati)$\rightarrow$ 0.90%
- Cella ha un raggio di $\rightarrow r$
- **Propagation time:**
$$\tau = \frac{r}{300.000\ km/s}$$
- **Guard time:**
$$G = 2 \cdot \tau$$
- **Transmission burst:** tempo necessario per ogni sorgente di trasmettere i propri bit.
$$\frac{T_s}{T_s + G} = 0.90\%\ \textit{efficency}$$
- **Frame duration:**
$$FD = 100\ \textit{canali}\ \cdot (T + G)$$
- **Number of bits of a transmission burst:**
$$B = \textit{velocità transmissione}\ \cdot FD$$
- **Transmission rate of the TDMA multiplex:**
$$W = \frac{B}{T_s}$$

3 TDMA es:

- Time slots N = 10;
- guard time $T_g = 200\mu s$
- data D = 180 *bit*
- header H = 20 *bit*
- frame time $T_t = 10\ ms$

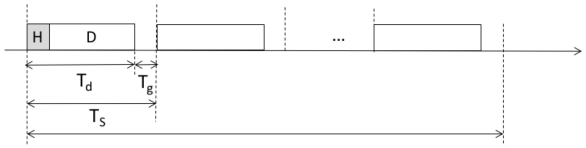


Figure 1: situation

$$k = 180 + 20 = 200\ bit$$
$$TS = \frac{T_t}{N} = 1\ ms$$
$$T_d = 1 - T_g = 0.8\ ms$$

W: velocità della trasmissione effettiva della parte dati + header senza T_g

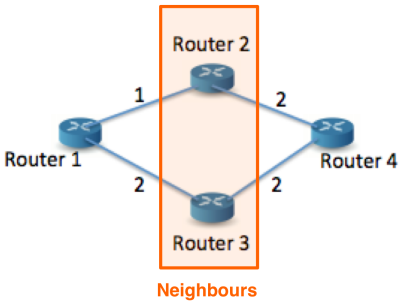
$$W = \frac{k}{T_d} = 250\ kbit/s$$

V_D: velocità della trasmissione effettiva della parte dati rispetto al tempo totale

$$V_D = \frac{D}{T_t} = 18\ kbit/s$$

4 Distance vector

- **Esempio:**



Destinazione	Next Hop	Costo
Net 1	Router 2	4
Net 2	Router 3	3
Net 3	Router 2	2
Net 4	Router 3	3
Net 5	Local	Local

Figure 2: **Neighbor:** sono tutti i router adiacenti a *R1*

- **Distance vector (without split horizon):** manda il distance vector (DV) a tutti i neighbor router.

Net # : costo

Net1:4, Net2:3, Net3:2, Net4:3, Net5:local

- **Distance vector (split horizon):** le net appartenenti allo stesso router di destinazione prendono come costo il Hop limit (se non presente poniamo a infinito), mentre le altre net hanno un costo normale

To router 2:

  - Net 1: **HOP LIMIT**
  - Net 2: 3
  - Net 3: **HOP LIMIT**
  - Net 4: 3
  - Net 5: local

  Nb: quelle con HOP LIMIT appartengono al router 2

To router 3:

  - Net 1: 4
  - Net 2: **HOP LIMIT**
  - Net 3: 2
  - Net 4: **HOP LIMIT**
  - Net 5: local

    Nb: quelle con HOP LIMIT appartengono al router 3

- **OSPF protocol:** si parte da un router principale es: *R2*
  - Se presenti border router questi sono considerati all'interno dell'area di *R2*;
  - Tutto quello che è presente nell'area di *R2* viene rappresentato in dettaglio (link, costi link, router, border routers, nodi)
  - Al di fuori dell'area di *R2*, vengono rappresentati solo i **NODI** che vengono collegati tramite link ai border routers se raggiungibili da tali e il loro **link cost** è dato dalla somma dei link per raggiungerli
- **Link state update messages (LSU)**
  Se il messaggio LSU ha:
  - **Sequence number < di quello presente in tabella**, il messaggio viene scartato.

    Eventualmente il router manda un messaggio dalla sorgente del LSU, ossia il router che ha mandato questo messaggio, il link aggiornato.
  - **Sequence number > di quello presente in tabella**, il riga in tabella viene aggioranta con quella della LSU.
  - delle informazioni non presenti all'interno della tabella viene inserita.
- **Reachability information of router R after DV:**
  - le righe che possiedono next hop = Router del DV vengono tutte aggiornate indipendentemente dal costo
  - se il DV per una destination già presente propone un nuovo next hop, viene aggioranta tale riga sse ha un costo inferiore per quel next hop, altrimenti non fa nulla

– se non esiste una destination che viene specificata all'interno di un DV, viene aggiunta

5 HTTP

- **Persisten connection:** viene instaurata una connessione tcp e viene mantenuta fino a quando non sono stati inviati tutti i file necessari poi la connessione viene chiusa.
- **Non persisten connection:** la connessione viene aperta e chiusa ad ogni elemento da inviare.
- Prima viene scaricata la pagina html e poi in parallelo gli oggetti che sono all'interno.
- l'esecuzione di n connessioni tcp in parallelo fanno in modo che transmissione effettiva per ogni connessione sia di:

$$\frac{C}{n}$$

- Solitamente per instaurare una connesione vi si scambiano control messages che hanno o meno (da esericizio) una lunghezza.
  - S = durata invio control message $[s]$
  - C = capacità di transmissione $[\frac{bit}{s}]$
  - m = dimensione control message $[bit]$

$$S = \frac{m}{C}$$

- **Instaurazione TCP connection:**

$$TCP = 2 \cdot \frac{m}{C} + 2 \cdot \tau$$

- **Invio messaggio effettivo:**
  - g = dimensione richiesta get
  - L = lunghezza oggetto richiesto

$$\frac{g + L}{C} + 2 \cdot \tau$$

  Per ogni get che io faccio devo instaurare una connessione tcp prima e poi ovviamente chiuderla.

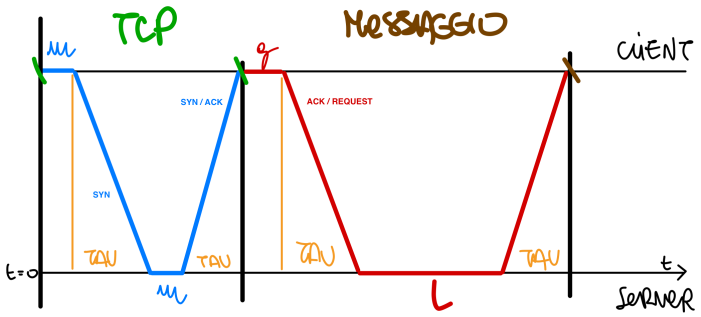


Figure 3: TCP

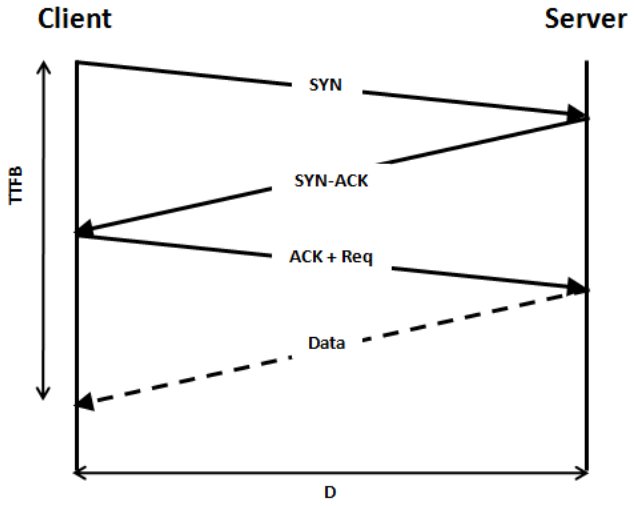


Figure 4: 3 way handshake

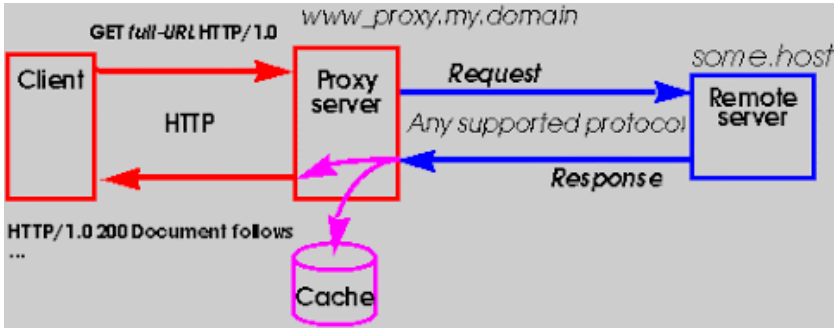


Figure 5: **Proxy:** request, response from

Esercizi http con canali strozzanti / bottleneck

- Indiviare per ogni connessione presente all'interno della rete il canale strozzante;
- Assegnare ad ogni connessione la capacità trasmissiva dividendo il canale oppure facendo la differenza

6 Bluetooth

- Totale slaves $\rightarrow M$
- Active slaves $\rightarrow N$
- Channel capacity $\rightarrow C$ *[kbit/s]*
- Propagation delay between slayes (tutti sono lontani uguali dal master) $\rightarrow \tau$
- **Transmission packet:**

$$T_p = \frac{\textit{packet size in [bit]}}{C}$$

- **Transmission token:**

$$T_t = \frac{\textit{token size in [bit]}}{C}$$

- **Usefull time (tempo effettivamente utilizzato per trasmettere dati):**

$$N \cdot T_p$$

- **Total duration of cycle:**

$$2 \cdot M \cdot (\tau + T_t) + N \cdot T_t$$

- **Efficiency:**

$$\eta = \frac{\textit{Usefull time}}{\textit{Total duration of cycle}} = \frac{N \cdot T_p}{2 \cdot M \cdot (\tau + T_t) + N \cdot T_D}$$

L'efficienza tende a diminuire a mano a mano che il pacchetto di dati da trasmettere ha una dimensione bassa

- **Compute the maximum time (worst case, all slave trasmitting data) for a slave to access the channel:**

$(M-1)\cdot(2\cdot\tau+2\cdot T_t+2\cdot T_p) = p \rightarrow$ *tutti i slave trasmettono tranne 1 di riferimento*

$T_{access} = p+2\cdot(\tau+T_t) \rightarrow$ *dove la prima volta non trasmette nulla worst case*

- **Parked slaves:** sono quello dove non devono trasmettere nulla

$$M = M + parked\ slaves$$