

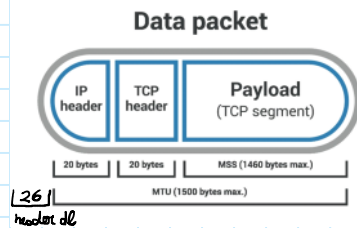
# Formulario

mercoledì 26 gennaio 2022 17:11

Livello trasporto => segmento

Livello rete => datagramma

Livello datalink => trama oppure frame



Tempo di trasmissione frame =  $\frac{\text{lunghezza frame (con header)}}{\text{capacità del canale da sorg a dest (velocità di trasmissione)}}$

Tempo di propagazione =  $\frac{\text{distanza tra sorgente e destinazione}}{\text{velocità di propagazione}}$

Tempo di trasmissione ack =  $\frac{\text{lunghezza ack}}{\text{capacità del canale da dest a sorg}}$

Se la velocità di propagazione è assente =>  $2 \cdot 10^8$

Se la dimensione ack è assente => dimensione minima del frame

Tempo di ciclo S&W (tc) =  $2 \cdot \text{TemProp} + \text{TempoTrasmFrame} + \text{TempoTrasmAck} + 2 \cdot \text{TempoElaborazione}$

Se tempo elaborazione assente => trascurare

MSS = dimensione segmento trasporto (senza header)

Numero segmenti generati =  $\frac{\text{dimensione flusso informativo}}{\text{MSS}}$

Durata trasmissione stop & wait = Numero segmenti generati \* Tempo di ciclo

Numero di trame inviate nel tempo di ON =  $\left\lfloor \frac{\text{tempo di on}}{\text{tempo di ciclo}} \right\rfloor$

Tempo prima trama persa (TPP) = Numero di trame inviate nel tempo di ON \* tempo di ciclo

Tempo in cui la trama persa viene inviata e ripersa (TR) = tempo di fine del periodo di OFF - TPP

Numero ritrasmissioni (cicli di tout) per raggiungere il primo ciclo a buon fine (TB) =  $\frac{\text{TR}}{\text{timeout}}$

Istante in cui avviene la trasmissione corretta del pacchetto perso = TPP + TB \* timeout

Durata trasmissione stop & wait con perdita = numero segmenti generati \* tc + TB \* timeout

Se timeout assente => 2 \* tempo di ciclo (forse)

Dimensione flusso che genera la sorgente = rate di generazione \* durata trasmissione (tempo di on)

Tempo di ciclo GB-n (tcc) =  $2 \cdot \text{TProp} + \text{dimFinestraSorg} \cdot \text{TTrasmFrame} + \text{TTrasmAck} + 2 \cdot \text{Telab}$

Num di finestre che saranno trasmesse (NF) =  $\frac{\text{Numero segmenti generati}}{\text{dimensione finestra sorgente}}$  *si intende la finestra che si usa durante la trasmissione.*

Trame trasmesse nell'ultima finestra che non è completa (Tu) = Numero seg generati - NF \* dim finestra

Tempo di ciclo trame residue (tctu) =  $2 \cdot \text{Tp} + \text{tu} \cdot \text{TtFrame} + \text{TtAck} + 2 \cdot \text{Te}$

Durata trasmissione GB-n quando tutto va a buon fine ack cumulativi =  $\text{tcc} \cdot \text{NF} + \text{tctu}$

Dur trasm GB-n buon fine ack singoli = (num seg generati - residue) \* TtFrame + tctu

Dur con perdita trama ack cumulativi =  $\text{tcc} \cdot \text{NF} + \text{tctu} + \text{timeout}$

Dur con perdita ack con ack cumulativi =  $\text{tcc} \cdot \text{NF} + \text{tctu} + \text{timeout}$

Dur perdita frame con ack singoli = (num trame trasmesse - residue) \* TtFrame + tctu + timeout

x = numero frame compreso il perso

Dur perdita frame intermedio con ack singoli =  $(x) \cdot \text{TTFr} + \text{tout} + (\text{num seg gen} - x - \text{residue}) \cdot \text{TTFr} + \text{tctu}$

Dur perdita ack con ack singoli = (num trame trasmesse - residue) \* TtFrame + tctu

Dur trasm non continua ack singoli = NF \* tcc

Go back-n frame inviati in maniera continua => dimensione finestra sorg \* Tra > Ttra + 2Tpro

Go back-n frame inviati in maniera non continua => dimensione finestra sorg \* Tra <= Ttra + 2Tpro

Durata trasmissione SR = (num seg gen - 1) \* tempoTrasmissioneFrame + tc

z = trame trasmesse prima della perdita

y=trame trasmesse dopo perdita

$D1 = z * \text{tempoTrasmTrama} + t_{out} + t_c$

$D2 = [n - (x + y) - 1] * \text{tempoTrasmTrama} + t_c$

Durata SR perdita ack =  $D1 + D2$  (finestra trasmittente bloccata)

Durata SR perdita trama =  $D1 + D2$  (tutte e due le finestre bloccate)

$\lambda$  = rate medio di arrivo dei frame

numero di pacchetti che mando in un secondo sul canale ( $G$ ) =  $\lambda * \text{TempoTrasmissioneTrame}$

Throughput (S) aloha =  $G * e^{-2G}$

$S_{\max} \text{ aloha} = \frac{1}{2e} = 18,3\%$

Tempo in cui aloha ritrasmette ( $Z$ ) = (somma cifre istante inizio trasmi \* num collisioni) +  $T_{\text{trasFrame}}$  (forse)

S aloha slotted =  $G * e^{-G}$

$S_{\max} \text{ aloha slotted} = \frac{1}{e}$

$G_{\max} = 1$

$a = \frac{t_p}{t_t}$

Funzionamento corretto csma-cd =>  $t_t \geq 2t_p$

S csma =  $\frac{G e^{-aG}}{G(1 + 2a) + e^{-aG}}$

Indirizzi ip

128	1000 0000	(128)
192	1100 0000	(64)
224	1110 0000	(32)
240	1111 0000	(16)
248	1111 1000	(8)
252	1111 1100	(4)
254	1111 1110	(2)
255	1111 1111	(1)

notazione "numerica"	10000101	01100000	00011001	01011111	
					notazione "puntata" (dotted) o decimale
		160.97.25.95			

Classe	Bit Iniziali	Net_Id	Host_Id	"reti logiche" disponibili	"indirizzi" disponibili
A	0	7 bit	24 bit	128	16 777 216
B	10	14 bit	16 bit	16 384	65 536
C	110	21 bit	8 bit	2 097 152	256
D	1110	indirizzi multicast: 28 bit indirizzi possibili: $2^{32/16} = 268\,435\,456$			
E	11110	riservati per usi futuri e ricerca: 27 bit indirizzi possibili: $2^{32/32} = 134\,217\,728$			

Classe A

(0.0.0.0 – 127.255.255.255)

Classe B

(128.0.0.0 – 191.255.255.255)

Classe C

(192.0.0.0 – 223.255.255.255)

Classe D

(224.0.0.0 – 239.255.255.255)

Classe E

(240.0.0.0 – 255.255.255.254)

CIDR	Class	Hosts*	Mask
/32	1/256 C	1	255.255.255.255
/31	1/128 C	2	255.255.255.254
/30	1/64 C	4	255.255.255.252
/29	1/32 C	8	255.255.255.248
/28	1/16 C	16	255.255.255.240
/27	1/8 C	32	255.255.255.224
/26	1/4 C	64	255.255.255.192
/25	1/2 C	128	255.255.255.128
/24	1 C	256	255.255.255.000
/23	2 C	512	255.255.254.000
/22	4 C	1024	255.255.252.000
/21	8 C	2048	255.255.248.000
/20	16 C	4096	255.255.240.000
/19	32 C	8192	255.255.224.000
/18	64 C	16384	255.255.192.000
/17	128 C	32768	255.255.128.000
/16	256 C, 1 B	65536	255.255.000.000

Supernetting terzo ottetto sempre pari.

Utilizzo rete=  $\frac{\text{numero indirizzi usati}}{\text{numero indirizzi totali}}$

Broadcast sempre dispari.

Maschera /24 => 255.255.255.0

Numero indirizzi usati= num host+base+broadcast+ num router

Reti punto-punto router= R1+R2+b+bb con maschera /30

Trasporto

$N_{mss} = \left\lceil \frac{\text{dimensione flusso trasmesso}}{mss} \right\rceil$  = numero mss totali da inviare

Dim max finestra trasmettitore ( $MAX_{mss}$ ) =  $\left\lceil \frac{\text{capacità del canale}}{(mss+H_{tcp}+H_{ip}+H_{dl}) * 8} \right\rceil$   
(per trasformarlo da byte a bit)

Upper bound= min (finestra sorg, finestra dest)=finestra max

Tempo per raggiungere la finestra max ( $T_{ss}$ ) =  $RTT * \log_2(\text{upper bound}) = RTT * k$

MSS trasferiti nel tempo  $T_{ss}$  ( $B_{ss}$ )=  $MSS * (2^{k+1} - 1)$

MSS rimanenti ( $m_r$ )=  $N_{mss} - B_{ss}$

RTT rimanenti ( $r_{im}$ )=  $\left\lceil \frac{m_r}{\text{upper bound}} \right\rceil$

Durata complessiva=  $T_{ss} + r_{im}$

Mss inviati nell'ultima finestra non completa=  $mss \text{ da inviare} - [\text{finestra max} * (\text{num rtt} - 1)]$

Tahoe=fast retransmit

Reno=fast retransmit e fast recovery

Timeout=2RTT

Tahoe

1. CWND = 1; → finestra torna a 1 → finestra max/2
2. SSThreshold = CWND/2 ~~min(2, SSThreshold)~~
3. Inizio con fase di SS perchè (CWND < SSThreshold);
4. quando CWND >= SSThreshold => Inizio CA;  
↓  
la finestra incrementa

3. Inizio con fase di SS perchè ( $CWND < SSThreshold$ );
4. quando  $CWND \geq SSThreshold \Rightarrow$  Inizio CA;

↓  
 la finestra incrementa  
 di 1 e non esponenzialmente  
 fino alla finestra max.

#### Reno

1.  $SSThreshold = CWND/2 = 32/2 = 16$ ;
2.  $CWND = CWND/2$
3.  $CWND \geq SSThreshold \Rightarrow$  Inizio CA;

$$SRTT(k+1) = a SRTT(k) + (1-a) RTT(k+1)$$

$$RTT = T_1 + T_2 + T_3 + T_{A1} + T_{A2} + T_{A3} + 2(T_1 + T_2 + T_3)$$

$$RTO = 2SRTT$$

Per link state e dijkstra.

- 1) fare le propagazioni con gli lsp dei lsdB a partire dal tempo di hello+ritardi
- 2) ripetere il passo 1), ma aggiungendo solo i ritardi al tempo, per raggiungere la topologia completa dei nodi (nt), di cui si vuole calcolare la tabella di routing.
- 3) applicare dijkstra (per tutti i nodi nt) con la tabella che contiene come colonne T e tutti i nodi:  
 nelle celle =  $\infty$ , se cammino non esiste  
 = costo da radice a nodo/precedente, se cammino esiste  
 un nodo si aggiunge a T quando nella riga il suo costo è il minore.
- 4) trovato il cammino minimo, con dijkstra fare la tab di routing TO-NEXT HOP-COST(per ogni nodo nt).
- 5) se cade un nodo o un collegamento ripetere i passaggi precedenti e il tempo iniziale sarà l'istante di caduta +3hello+ ritardi

NB: gli hello scorrono dal tempo 0 a intervalli regolari (tempo di hello) indipendentemente dalla caduta.

