

# 1 Esame 1-7-2019

Consideriamo una applicazione web basata su http 1.1. Il browser del Client consente l'apertura di 3 connessioni TCP persistenti in parallelo, chiamiamole T1, T2 e T3 per semplicità:

- Client e Server sono connessi a due diverse subnet IP, ma si trovano nello stesso edificio. La rete che li collega è basata su tecnologia LAN e il router che interconnette le due subnet non è un collo di bottiglia.
- Il client ha una interfaccia a 100 Mbit/s mentre il server è collegato a 1 Gbit/s.
- Il tempo di trasferimento di un pacchetto IP tra Client e Server, inclusi gli switch Ethernet, la commutazione del router e il tempo di trasmissione di un pacchetto è approssimativamente costante e pari a  $600\mu s$ , in media la differenza tra il singolo RTT e SRTT (Smoothed Round Trip Time) è  $|SRTT - RTT| = 100\mu s$ .
- Per il calcolo di RTT si usa l'opzione timestamp (10 byte di overhead). Entrambe le Receiver Windows (RCWND) sono pari a 64 kbyte e all'inizio STHDR = RCWND/2.
- La sessione di lavoro prevede che vengano trasferiti "oggetti" (file, dati, immagini, ...) sia dal client al server che viceversa. Chiamiamo OS1, OS2, ... gli oggetti trasferiti dal server e OC1, OC3, ... quelli trasferiti dal client.
- All'apertura della sessione il client invia al server l'oggetto OC1 di dimensioni 3800 byte. Il server risponde con l'oggetto OS1 di dimensioni 5400 byte.
- Dopo questo primo scambio client e server si scambiano immediatamente e in parallelo i seguenti oggetti delle dimensioni (in byte) specificate:
  - OC2 = 4200 byte e OS2 = 9800 byte
  - OC3 = 10400 byte e OS3 = 12600 byte
  - OC4 = 8700 byte e OS4 = 20500 byte.

## Domande:

1. Calcolare l'RTO (Retransmission Time Out) sia del client che del server.
2. Mostrare in un diagramma lo scambio dei segmenti sulla connessione T1 che corrispondono al trasferimento di OC1 e OS1.
3. Su quali connessioni (T1, T2, T3) vengono trasferiti gli 8 oggetti? Esiste una soluzione unica oppure decide il browser dal lato client e il server web dal lato server?
4. La perdita di pacchetti su una connessione influenza il comportamento dell'altra?
5. Come si influenzano reciprocamente le due "half-connections" di TCP su ciascuna delle connessioni T1, T2 e T3?
6. Mostrare in un diagramma lo scambio dei segmenti relativo all'oggetto OS2 supponendo che il 2° viene scartato dal router.
7. Calcolare il tempo di trasferimento totale degli 8 oggetti in assenza di perdite (bisogna tenere in conto di tutte e tre le connessioni e dell'interazione sulla rete Ethernet). Si trascurino per semplicità gli header di http e altre informazioni che client e server potrebbero aggiungere agli 8 oggetti, ma sientino tutti gli overhead di TCP, IP e Ethernet.
8. Supponiamo ora che il browser consenta l'apertura di 1 sola connessione e non di 3; ricalcolare il tempo di trasmissione totale degli 8 oggetti.
9. Commentare alla luce dello scenario dato i risultati ottenuti ai punti 7 e 8.

**Soluzione:**

1. Visto che i tempi di trasmissione di un pacchetto tra Client e Server, inclusi gli switch e router, si ha che:

$$RTT = 600\mu s \cdot 2 = 1200\mu s$$

- Sappiamo che:

$$\begin{aligned} SRTT &= (1 - \alpha) \cdot SRTT + \alpha \cdot RTT \text{ (con } \alpha = 0.125) \\ RTTVAR &= (1 - \beta) \cdot RTTVAR + \beta \cdot |SRTT - RTT| \end{aligned}$$

- RTO è definito come:

$$RTO = \text{MAX}(\text{MINRTO}, SRTT + 4 \cdot RTTVAR)$$

- MINRTO = 1s (standard)

$$\begin{aligned} SRTT + 4 \cdot RTTVAR &= [(1 - \alpha) \cdot SRTT + \alpha \cdot RTT] + 4 \cdot [(1 - \beta) \cdot RTTVAR + \beta \cdot |SRTT - RTT|] \\ &= (\alpha \cdot RTT) + (4 \cdot \beta \cdot |SRTT - RTT|) \\ &= (0.125 \cdot 1200\mu s) + (4 \cdot 0.25 \cdot 100\mu s) \\ &= 150\mu s + 100\mu s = 250\mu s \end{aligned}$$

- Dunque  $RTO = \text{MAX}(1s, 250\mu s) = 1s$

2. Calcolo dei dati necessari per rispondere alla domanda:

- **Timestamp** è un'opzione che viene usata per il calcolo del RTT su qualunque segmento, semplicemente si ha che ogni segmento ha un overhead di 10 byte in aggiunta all'header TCP, dunque si ha che:

$$MSS = 1500 - 20 - 20 - 10 = 1450$$

- Visto che TCP ragiona in Byte, ma noi siamo più comodi a ragionare con segmenti, converto le RCWND e le dimensioni degli oggetti in segmenti:

$$\begin{aligned} RCWND_{Client} &= \frac{64 \text{ KB}}{1450 \text{ Byte}} = \frac{65536 \text{ Byte}}{1450 \text{ Byte}} = 45.19 = 46 \text{ pacchetti} \\ RCWND_{Server} &= \frac{64 \text{ KB}}{1450 \text{ Byte}} = \frac{65536 \text{ Byte}}{1450 \text{ Byte}} = 45.19 = 46 \text{ pacchetti} \\ OC1 &= \frac{3800 \text{ Byte}}{1450 \text{ Byte}} = 3 \text{ pacchetti} \\ OS1 &= \frac{5400 \text{ Byte}}{1450 \text{ Byte}} = 4 \text{ pacchetti} \end{aligned}$$

- Dunque le connessioni verranno utilizzate nel seguente modo:

$$\begin{aligned} T1 &= OC1 = 3 \text{ segmenti e } OS1 = 4 \text{ segmenti} \\ T2 &= \text{non usata} \\ T3 &= \text{non usata} \end{aligned}$$

- Trasferimento OC1:

RTT	CWND	$T_w$
1	1	[1]
2	2	[2, 3]

- Trasferimento OS1:

RTT	CWND	$T_w$
1	1	[1]
2	2	[2, 3]
3	4	[4]

- Dunque per il trasferimento di OC1 e OS1, il tempo impiegato  $T_t = 5 \cdot RTT = 5 \cdot 1.2 \text{ ms} = 6.0 \text{ ms}$
3. Gli oggetti vengono trasmessi nelle connessioni T1, T2, T3 a discrezione del browser del Client e dal Server-Web del Server, che sceglieranno (per questioni di efficienza, congestione, carico di dati) il modo più appropriato per trasmettere i dati.
  4. La perdita di pacchetti su una connessione non influenza direttamente il comportamento di un'altra connessione. Prendiamo come esempio T1 e T2; se T1 sta inviando dati T2 non può inviare nulla, finché T1 non ha finito di usare il canale; dunque se T1 perde pacchetti, l'unica conseguenza è che T2 subirà un ritardo nel trasmettere i suoi dati.
  5. Essendo un'architettura Client-Server di norma vi è una richiesta o un invio di dati lato client ed una successiva risposta lato server. Le due half-connections funzioneranno quindi a tratti alterni, come se la connessione non fosse full-duplex.
  6. Converto la dimensione dell'oggetto OS2 in segmenti:

$$OC2 = \frac{4200 \text{ Byte}}{1450 \text{ Byte}} = 3 \text{ pacchetti}$$

- Trasferimento OC2:

RTT	CWND	$T_w$
1	1	[1]
2	2	[2, 3]
Il 2° pacchetto viene perso, trascorso RTO = 1s, viene ritrasmesso SSTHR = 46/2 = 23 segmenti		
3	1	[2]
4	2	[3]

- Dunque per il trasferimento di OC2, il tempo impiegato  $T_t = 4 \cdot RTT + RTO = 4 \cdot 1.2 \text{ ms} + 1 \text{ s} = 4.8 \text{ ms} + 1000 \text{ ms} = 1004.8 \text{ ms}$
7. Dunque, sappiamo che per la prima trasmissione (OC1 e OS1) impiego  $3.6 \text{ ms}$ , ora devo calcolare il tempo delle restati trasmissioni:

$$\begin{aligned}
OC2 &= \frac{4200 \text{ Byte}}{1450 \text{ Byte}} = 3 \text{ pacchetti} \\
OS2 &= \frac{9800 \text{ Byte}}{1450 \text{ Byte}} = 7 \text{ pacchetti} \\
OC3 &= \frac{10400 \text{ Byte}}{1450 \text{ Byte}} = 8 \text{ pacchetti} \\
OS3 &= \frac{12600 \text{ Byte}}{1450 \text{ Byte}} = 9 \text{ pacchetti} \\
OC4 &= \frac{8700 \text{ Byte}}{1450 \text{ Byte}} = 6 \text{ pacchetti} \\
OS4 &= \frac{20500 \text{ Byte}}{1450 \text{ Byte}} = 15 \text{ pacchetti}
\end{aligned}$$

- Dunque le connessioni verranno utilizzate nel seguente modo:

$T1 = OC2 = 3$  segmenti e  $OS2 = 7$  segmenti

$T2 = OC2 = 8$  segmenti e  $OS2 = 9$  segmenti

$T3 = OC2 = 6$  segmenti e  $OS2 = 15$  segmenti

- Non si possono considerare separatamente le connessioni. Bisogna considerare che per inviare un segmento ti servono lato client  $100\mu s$ . I segmenti vengono inviati contemporaneamente dalle 3 connessioni, ma uno alla volta sul canale. Se le 3 connessioni partono con finestra 1 contemporaneamente, la prima invia il 1° segmento al tempo 0, la seconda al tempo  $100\mu s$ , la terza al tempo  $200\mu s$ . In un RTT ci stanno circa 12 segmenti. I segmenti sono 12 e hanno spazio sul canale sia che la connessione sia una, che le connessioni siano N (non è che sono 12 per ogni connessione), bisogna tener conto di questo quando si svolge l'esercizio. Poi c'è la parte complicata che i due link hanno delle velocità diverse, quindi bisogna considerare il link più lento come collo di bottiglia, ma queste sono cose decisamente complicate che noi non abbiamo visto a lezione. Sotto viene svolto solo il calcolo del tempo impiegato per trasferire gli oggetti se la connessione fosse una sola e non parallela.

- Trasferimento OC2:

RTT	CWND	$T_w$
1	1	[1]
2	2	[2, 3]

- Trasferimento OS2:

RTT	CWND	$T_w$
1	1	[1]
2	2	[2, 3]
3	4	[4, 5, 6, 7]

- Il trasferimento di OC2 e OS2:  $T_{t1} = 5 \cdot RTT = 5 \cdot 1.2 ms = 6 ms$

- Trasferimento OC3:

RTT	CWND	$T_w$
1	1	[1]
2	2	[2, 3]
3	4	[4, 5, 6, 7]
4	8	[8]

- Trasferimento OS3:

RTT	CWND	$T_w$
1	1	[1]
2	2	[2, 3]
3	4	[4, 5, 6, 7]
4	8	[8, 9]

- Il trasferimento di OC3 e OS3:  $T_{t2} = 8 \cdot RTT = 8 \cdot 1.2 ms = 9.6 ms$

- Trasferimento OC4:

RTT	CWND	$T_w$
1	1	[1]
2	2	[2, 3]
3	4	[4, 5, 6]

- Trasferimento OS4:

RTT	CWND	$T_w$
1	1	[1]
2	2	[2, 3]
3	4	[4, 5, 6, 7]
4	8	[8, 9, 10, 11, 12, 13, 14, 15]

- Il trasferimento di OC4 e OS4:  $T_{t3} = 7 \cdot RTT = 7 \cdot 1.2 \text{ ms} = 8.4 \text{ ms}$

8. Calcolo la somma dei dati che devo inviare:

$$OC = 3800 + 4200 + 10400 + 8700 = 19100 = \frac{19100 \text{ Byte}}{1450 \text{ Byte}} = 14 \text{ pacchetti}$$

$$OS = 5400 + 9800 + 12600 + 20500 = 48300 = \frac{48300 \text{ Byte}}{1450 \text{ Byte}} = 34 \text{ pacchetti}$$

- Trasferimento OC:

RTT	CWND	$T_w$
1	1	[1]
2	2	[2, 3]
3	4	[4, 5, 6, 7]
4	8	[8, 9, 10, 11, 12, 13, 14]

- Trasferimento OS:

RTT	CWND	$T_w$
1	1	[1]
2	2	[2, 3]
3	4	[4, 5, 6, 7]
4	8	[8, 9, 10, 11, 12, 13, 14, 15]
5	16	[16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31]
Nel successivo passo, per lo Slow Start di TCP la finestra dovrebbe passare a 32 segmenti ma visto che SSTHR = 23, il protocollo passa a Congestion Avoidance e incrementa di 1 la finestra		
6	17	[32, 33, 34]

- Il trasferimento di OC4 e OS4:  $T_{t3} = 10 \cdot RTT = 10 \cdot 1.2 \text{ ms} = 12 \text{ ms}$

9. TODO

## 2 Esame 12-6-2019

- Un client HTTP deve trasferire da un server un file di 56200 bytes.
- Gli host (client e server) hanno i seguenti indirizzi IP 128.122.15.171/26 e 128.122.15.16/24, la tecnologia usata per la trasmissione e l'accesso alla rete è Ethernet commutata a 100 Mbit/s dal lato client e 1 Gbit/s dal lato server.
- La rete Ethernet non è congestionata e si misura una latenza (ritardo tra l'inizio della trasmissione lato server e l'inizio della ricezione lato client)  $T_e = 1.0$  ms.
- L'overhead di Ethernet è pari a 36 byte (inclusendo tutti i campi e anche l'inter-packet-gap).

### Domande:

1. La consegna dei pacchetti IP avviene in modo diretto o indiretto? Perché?
2. Che valore assume MSS (Maximum Segment Size di TCP) in assenza di opzioni per TCP e con il normale header IP (anche qui nessuna estensione o opzione viene usata)?
3. Quanti segmenti (e quindi pacchetti IP) verranno trasmessi sulla rete?
4. Che dimensione ha l'ultimo segmento dati della connessione?
5. Si mostrino i segmenti scambiati per l'apertura della connessione TCP conseguente al comando di "get" da parte del client FTP, scegliendo opportunamente le porte (port number) di TCP dal lato client e dal lato server.
6. Si mostri l'intero scambio di segmenti TCP per trasferire il file, calcolando il tempo di trasferimento e il throughput ottenuto a livello HTTP, cioè relativo ai byte utili per l'applicazione; la connessione viene chiusa dal server con un RST appena ricevuto l'ultimo ACK.

La rete perde il 12° segmento (contato ordinatamente nella segmentazione del file: i segmenti di apertura/chiusura e le ritrasmissioni non contano). Dato il basso valore di RTT, il Retransmission Timeout (RTO) di TCP è fissato al minimo ammesso dal sistema operativo:  $RTO = 120$  ms.

7. Si mostri nuovamente l'intero scambio di segmenti tra client e server calcolando anche il tempo di trasmissione del file e il throughput come al punto 6.

### Risposte:

1. La consegna dei pacchetti IP avviene in modo indiretto, in quanto i due indirizzi IP non appartengono alla stessa rete:

$$\text{IP } 128.122.15.171/26 : \begin{cases} \text{IP} & : 10000000.01111010.00001111.10101011 \text{ (128.122.15.171)} \\ \text{SM} & : 11111111.11111111.11111111.11000000 \text{ (255.255.255.192)} \\ \text{Net} & : 10000000.01111010.00001111.10000000 \text{ (128.122.15.128)} \end{cases}$$

$$\text{IP } 128.122.15.16/24 : \begin{cases} \text{IP} & : 10000000.01111010.00001111.00010000 \text{ (128.122.15.16)} \\ \text{SM} & : 11111111.11111111.11111111.00000000 \text{ (255.255.255.0)} \\ \text{Net} & : 10000000.01111010.00001111.00000000 \text{ (128.122.15.0)} \end{cases}$$

Visto che si hanno due indirizzi di rete diversi (128.122.15.128 e 128.122.15.0) allora Client e Server non comunicano direttamente, ma dovranno comunicare tramite un router.

2. MSS assume il valore:

$$MSS = 1500 - 20 - 20 - 36 = 1424 \text{ Byte}$$

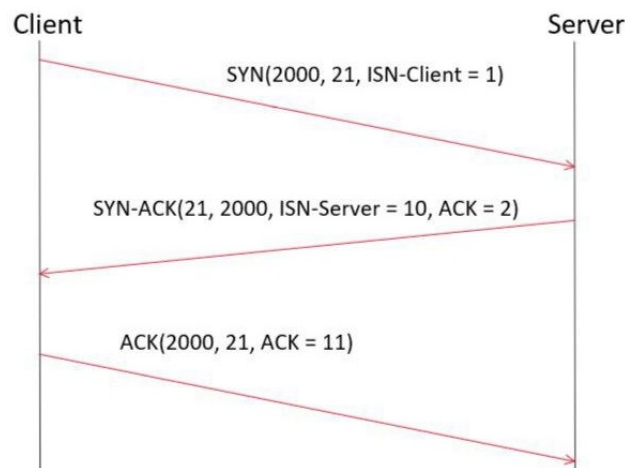
3. In rete vengono trasmessi:

$$N_s = \frac{56200 \text{ Byte}}{1424 \text{ Byte}} = 40 \text{ segmenti}$$

4. L'ultimo segmento ha dimensione:

$$56200 \text{ Byte} - (39 \cdot 1424) \text{ Byte} = 56200 \text{ Byte} - 55536 \text{ Byte} = 664 \text{ Byte}$$

5. TCP utilizza un meccanismo a 3 step per aprire una connessione tra sorgente e destinatario. Questa prevede che il client mandi un pacchetto speciale (con il flag SYN settato ad 1) al Server. Una volta ricevuto dal Server questo decide se accettare la domanda di connessione o rifiutarla. Se la accetta, allora risponde anch'esso con un pacchetto SYN (non è un semplice SYN in quanto oltre al flag SYN ha anche il flag ACK settato così da notificare che ha ricevuto la richiesta precedentemente inviata dal Client). A questo punto il Client invia un ACK per confermare la ricezione del messaggio del Server e finito questo procedimento, esiste una connessione TCP tra il Client e il Server.



Il primo pacchetto SYN inviato dal Client contiene:

- Porta sorgente: 2000 (generato casualmente)
- Porta destinazione: 21 (porta FTP)
- Sequence Number: 1 (generato casualmente)

Il pacchetto SYN-ACK inviato dal Server contiene:

- Porta sorgente: 21 (porta FTP che fornisce il servizio richiesto)
- Porta destinazione: 2000 (porta del Client che ha richiesto il servizio)
- Sequence Number: 10 (generato casualmente)
- ACK: 2 (Sequence number inviato dal Client incrementato di 1 per specificare a quale pacchetto fa riferimento e ha ricevuto)

L'ultimo pacchetto inviato dal Client:

- Porta sorgente: 2000
- Porta destinazione: 21
- Sequence Number: 2 (generato casualmente)
- ACK: 11 (Sequence number inviato dal Server incrementato di 1 per specificare a quale pacchetto fa riferimento e ha ricevuto)