RETI DI CALCOLATORI - Prima verifica in itinere a.a. 2019/2020

RECUPERO 7 dicembre 2019

**********Soluzioni**********

Q1 (3 punti) Data la stringa http://demo.di.unipi.it/service/sum?a=3&b=5, indicare le parti di cui è composta e il loro significato.

Soluzione

La stringa è una http URL. E' composta dalle seguenti parti: schema (http://) indica lo schema per identificare la risorsa, in una http URL indica il meccanismo di accesso i.e. protocollo; authority (demo.di.unipi.it) in questo caso è un nome di dominio di un host, in alternativa potrebbe essere indicato un indirizzo IP; path (service) identifica la risorsa nel contesto dello schema e dell'authority, rappresenta il percorso logico con cui sono organizzate le risorse offerte dal servizio web; query (sum?a=3&b=5) rappresenta una query e i relativi parametri di input e deve essere interpretata dal server.

Q2 (5)

Indicare se le seguenti affermazioni sono vere o false e spiegare perché.

a) L'host A sta inviando un file di grandi dimensioni all'host B su una connessione TCP. B non ha dati da inviare ad A. L'host B non invierà acknowledgement ad A perché non può inviare i riscontri in piggybacking con i dati.

Soluzione

Falso. Il Piggybacking è usato per efficienza. Se non ci sono dati da inviare in piggybacking, B invierà solo gli ack.

b) Si supponga che l'ultimo sampleRTT di una connessione TCP connection sia 1 secondo. Quindi il valore di EstimatedRTT, utile per il calcolo del timeout di ritrasmissione per la connessione, sarà necessariamente un valore >= 1 secondo

Soluzione

Falso. EstimatedRTT = (1- alpha)* last_estimated_RTT + alpha*RTT_sample. Anche se l'ultimo sampleRTT è 1 sec, EstimatedRTT dipende da alpha e last_estimated_RTT. Di conseguenza, EstimatedRTT non è necessariamente più grande di 1sec.

Q3 (8 punti) L'host A invia un file che consiste in 9 segmenti TCP full sized (i.e. ciascun segmento trasporta una quantità di dati pari a 1 MSS) a B su una connessione TCP. Il sesto segmento nella trasmissione viene perso. Considerando TCP Reno, mostrare con un diagramma lo scambio di segmenti tra A e B indicando i valori di SEQ e ACK, e eventuali cambiamenti nel valore di congwin, soglia e stato del TCP Reno di A. Indicare infine il tempo totale per l'invio del file. Si assuma che la connessione TCP sia stata instaurata, che non siano stati ancora inviati dati, che RTT valga *d* ms e, per semplicità, che il retransmission timeout sia >2*d.* Si assuma inoltre che valga il riscontro cumulativo e che venga riscontrato ciascun segmento ricevuto (no delayed ack).

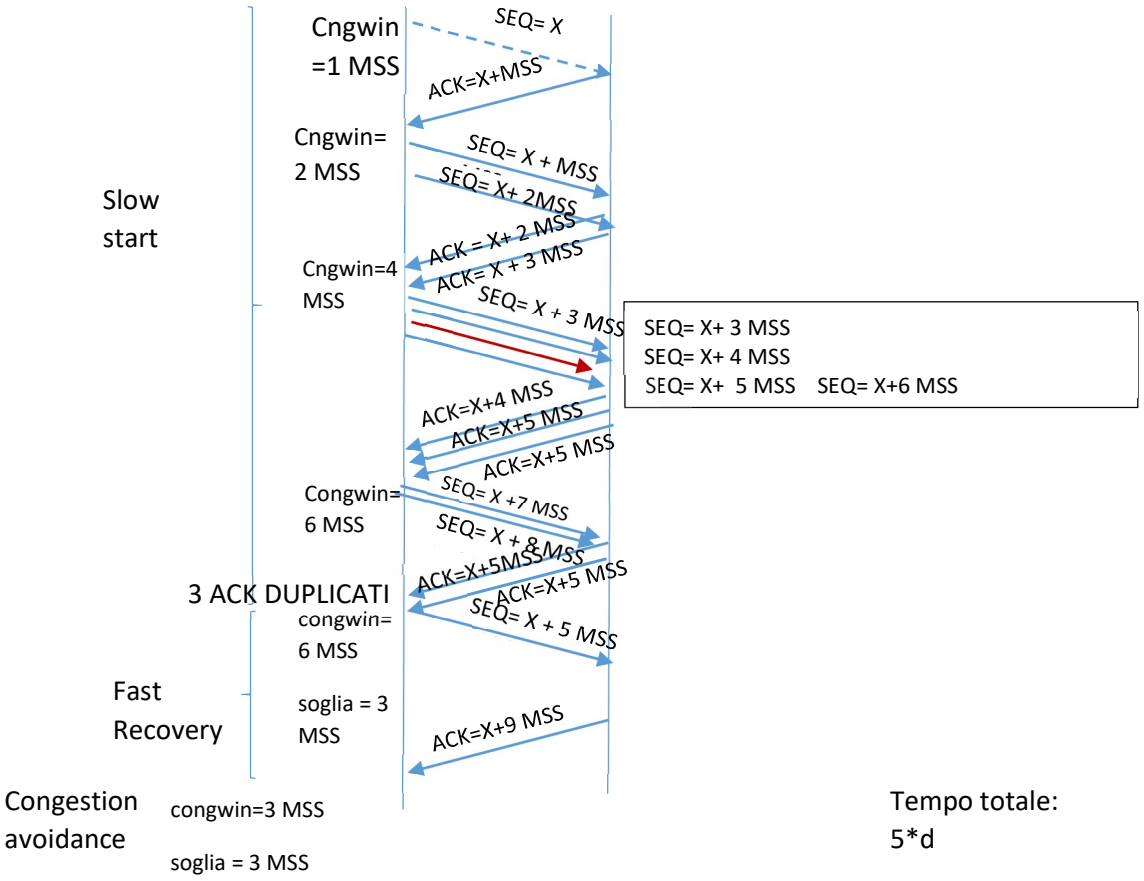
Soluzione

Si ipotizza che la receive window sia sempre maggiore di congestione window e quindi trascurabili.

Il trasferimento dati inizia nella fase di slow start con cong win =1 MSS e soglia alta (es. 64 KB).

Valore di sequenza per A all'inizio del trasferimento è X. B non invia dati utili ad A,quindi possiamo trascurare seq number per i segmenti da B ad A e riscontro da A.

La cong window aumenta esponenzialmente ad ogni RTT (1 MSSper ogniACK) finchè non viene ricevuto un ACK duplicato. Al terzo ACK duplicato il TCP va in FAST RECOVERY e reinvia il segmento mancante.

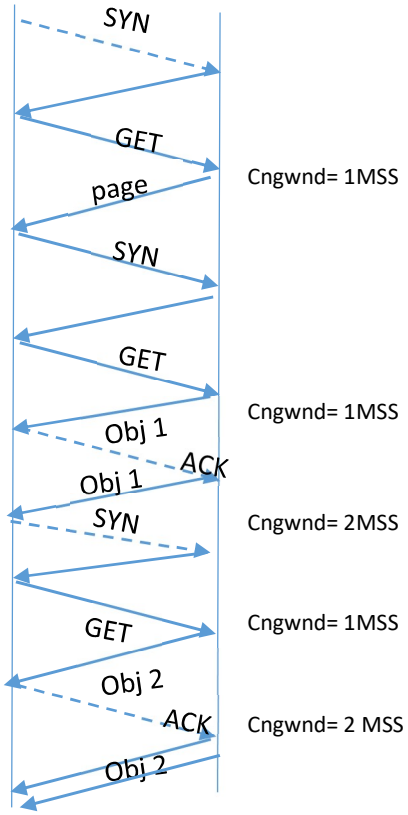


Q4. (8 PUNTI)

Si consideri un client che scarica da un server web una pagina di dimensione di 1000 byte, contenente i riferimenti a 2 oggetti, di dimensione rispettivamente 2000 e 4000 byte. Si assume che il valore di RTT (Round Trip Time) sia di 100 millisecondi e che i tempi di trasmissione e di elaborazione della risposta siano trascurabili e che non ci siano perdite né ritardi dovuti a congestione in rete. Si ipotizzi un MSS di 1460 byte. Stimare il tempo necessario per ricevere la pagina e tutti gli oggetti collegati nei seguenti casi:

1. Il client e il server utilizzano il protocollo HTTP 1.0
2. Il client e il server utilizzano il protocollo HTTP 1.1

Soluzione



HTTP1.0

La connessione è non persistente. Considerando che la connessione è stata instaurata con la prima richiesta e quindi la congwin cresce in modo esponenziale, il tempo totale è 4 RTT. Ipotizzando che cngwindw=1 MSS quando il server invia il primo messaggio sia ha:

syn: 1RTT

tempo per recuperare la homepage: 1 RTT (cngwndw= 1MSS)

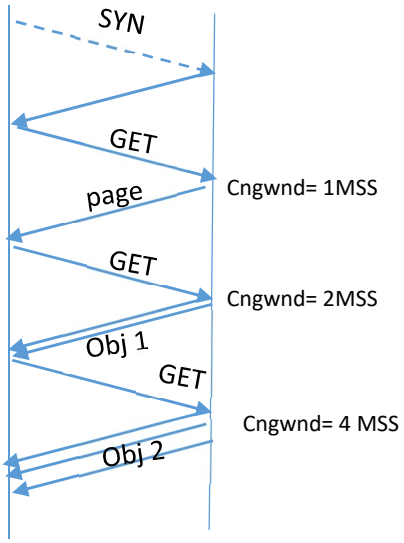
syn: 1RTT

tempo per recuperare object 1: 2RTT (cngwndw= 1MSS e poi 2 MSS)

syn: 1RTT

tempo per recuperare object 2: 2RTT(cngwndw= 1MSS e poi 2 MSS)

tempo totale: 8 RTT = 800ms



HTTP1.1

La connessione è persistente. Considerando che la connessione è stata instaurata con la prima richiesta e quindi la congwin cresce in modo esponenziale e ipotizzando che cnqwindw=1 MSS quando il server invia il primo messaggio sia ha:

syn: 1RTT

tempo per recuperare la homepage: 1RTT (cngwindw=1MSS)

tempo per recuperare object 1: 1RTT (cngwindw=2MSS)

tempo per recuperare object 2: 1RTT (cngwindw=4MSS oppure 3 MSS con delayed + cumulative ack)

totale= 4RTT = 400 ms

Q5 (6 punti) Con riferimento al livello di trasporto, spiegare cosa si intende per multiplexing e demultiplexing e, per il demultiplexing distinguere inoltre la modalità connection-oriented e connection-less

Soluzione

Demultiplexing: lo strato Trasporto provvede allo “smistamento” dei pacchetti fra la rete e le applicazioni (processi).

Multiplexing: lo strato Trasporto provvede all'“accorpamento” dei flussi dati dai processi verso la rete.

Le operazioni di multiplexing e demultiplexing si basano sui socket address dei processi. Il socket address è identificato dalla combinazione indirizzo IP e numero di porta. Il demultiplexing consiste quindi nel consegnare i segmenti ricevuti alla socket appropriata. Il multiplexing consiste nel raccogliere dati da varie socket e consegnarli allo strato di rete dopo aver aggiunto l’intestazione.

Nel demultiplexing senza connessione il datagramma UDP viene consegnato alla socket identificata da IP e porta destinatario. Nel demultiplexing orientato alla connessione il segmento TCP viene consegnato alla socket identificata dalla quadrupla IP e porta sorgente e IP e porta destinatario.