

RETI DI CALCOLATORI – prova scritta del 19/01/2017

Per essere ammessi alla prova orale è necessario ottenere una valutazione sufficiente sia della prima parte che dell'intera prova.

Prima parte (15 punti)

Q1. Si consideri il blocco di indirizzi 129.66.35.48/X. Quanto può valere X? Giustificare la risposta.

Q2. Un host A utilizza webmail per comunicare con un altro host B. A invia a B una email con testo di 1Kbyte e contenente tre oggetti di dimensione 4, 2 e 32 Kbyte rispettivamente. Se il maximum segment size è di 2 Kbyte, la maximum transmission unit sul percorso è di 4 Kbyte, e i segmenti viaggiano lungo un percorso con router tutti IPv6, quanti datagram contenenti la email riceverà B? Giustificare la risposta.

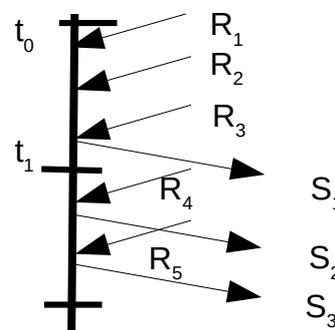
Q3. Al tempo t_0 , due host H e J hanno appena stabilito una connessione TCP senza inviare ack in piggybacking durante il three way handshake. Il valore di ssthreshold dell'host H e quello di rwnd sono uguali a 5 MSS. Indicare – giustificando la risposta- il numero minimo e massimo di segmenti che H invierà tra il tempo t_0 e il tempo t_1 (tempo in cui passa nello stato di congestion avoidance), nell'ipotesi che nessun segmento e nessun riscontro vada perduto e tutti i segmenti e i riscontri siano ricevuti correttamente.

Q4. Si consideri una rete locale IEEE 802.11 con frequenza effettiva di trasmissione di 16 Mbps, raggio di trasmissione di 300 metri, velocità di propagazione di $2 \cdot 10^8$ m/s, e con RTS e CTS di 20 byte, ACK di 14 byte, DIFS di 50 microsecondi e SIFS di 16 microsecondi. Il valore del timeout per decidere che il CTS relativo ad un RTS non è stato inviato è uguale a due volte il tempo che intercorre tra l'inizio dell'invio di un RTS e il tempo di ricezione del CTS relativo. Nel caso di non ricezione del CTS, si adotta la tecnica del binary exponential backoff, con tempo di backoff T_b uguale al tempo di trasmissione di 64 byte. Un nodo N della rete inizia ad eseguire il protocollo per la trasmissione di un frame di 2700 byte al tempo t_0 e, la prima volta che trasmette il RTS, non riceve il CTS relativo, mentre la seconda volta lo riceve. Supponendo che nessun altro host della rete voglia trasmettere frames, dopo quanto tempo da t_0 N inizierà *in media* a trasmettere il frame? Giustificare la risposta.

Q5. Quali protocolli di livello di rete vengono utilizzati nella fase di agent discovery di IP Mobile? Giustificare la risposta.

Seconda parte

E1 (8 punti). Si consideri lo scenario TCP relativo a due host A e B che hanno già stabilito una connessione TCP ed hanno iniziato a scambiarsi dati. Dall'istante t_0 e fino all'istante t_2 l'host A riceve i riscontri e invia i dati come mostrato nella figura a lato. Al tempo t_0 , nella finestra che inizia con $S_f = X$, A ha 5 MSS di dati in volo (inviati con 5 segmenti full size) e 5 MSS di nuovi dati non ancora inviati. Inoltre, il contatore di ack duplicati vale 1. I segmenti S_1 ed S_2 contengono 1 MSS di nuovi dati, mentre S_3 ne contiene 0,5 MSS. Al tempo t_1 il TCP di A si trova nello stato di *congestion avoidance*, con $ssthresh = 2,5$ MSS, $cwnd = 125/22$ MSS e $rwnd = 4$ MSS. Indicare – giustificando la risposta – lo stato e i valori di $ssthresh$, $cwnd$ (che si può lasciare anche come somma di frazioni), e $rwnd$ del TCP di A ai tempi t_0 e t_2 . Si supponga che tra t_0 e t_2 non scada alcun timeout e non si verifichi nessun altro evento oltre a quelli disegnati in figura. Inoltre, si supponga che R_1 ed R_2 siano riscontri duplicati. *Per un errore nel testo, si è ipotizzato che nel passaggio di stato tra FR e CA le variabili non vengano cambiate, quindi che non si ponga: $cwnd = ssthresh$.*



E2 (7 punti). Descrivere, con uno pseudocodice, il comportamento del router demone, *routedaem*, di RIP di un nodo R. Per semplicità, si supponga che i vicini di R siano rappresentati dai primi M numeri interi, e si supponga di avere a disposizione:

- *UDP_receive(j,A)* per ricevere un aggiornamento A dal vicino j

- *UDP_send(i,A)* per inviare l'aggiornamento A al vicino i

- *tabinoltro* tabella di inoltro di n righe con i campi, tra gli altri, *costo* (che contiene il costo del percorso memorizzato in quella riga), e *nexthop* (che contiene il nome dell'entità a cui inviare i datagram se si usa quel percorso). Una riga che non contiene informazioni valide avrà valore *NULL*.

Per semplicità, si supponga che la tabella di inoltro contenga già tutte le n righe (alcune con contenuto valido ed altre che saranno necessarie in futuro, queste ultime inizializzate a *NULL*): quindi non bisogna inserire nuove righe.

Q1. Siccome il primo indirizzo del blocco si ottiene ponendo a 0 gli ultimi 32-X bit dell'indirizzo (e l'ultimo tutti a 1), e siccome 48 in binario è 00110000, allora $X=28$, come valore minimo, e $X=32$ come valore massimo.

Q2. Webmail fa solo da interfaccia, ma il protocollo utilizzato dai mailserver per inviare in internet e ricevere la email sarà comunque SMTP. Inoltre, come fa A a sapere se B userà webmail? Quindi, il testo e tutti gli oggetti verranno messi in un unico messaggio, di 39 Kbyte. A questo si aggiungeranno gli header SMTP, TCP e IP. Quindi, in rete viaggeranno $39K+X+Y+Z$ byte per i datagram, dove X, Y e Z sono le lunghezze totali degli header. Si avranno quindi $\text{ceil}[(39+X+Y+Z)/2]$ datagram, dove $\text{ceil}[n]$ restituisce l'intero non inferiore di n, visto che $MSS=2K\text{byte}<MTU$. Ad esempio, se $X+Y+Z<1K\text{byte}$, allora avremo 20 datagram.

Q3. All'inizio H invierà un segmento ed aspetterà il riscontro, perché $cwnd=1$ all'inizio. Ricevuto questo riscontro, $cwnd=2$. A questo punto H può iniziare ad inviare i segmenti uno dopo l'altro, e ricevere i riscontri. Riceverà almeno un riscontro ogni 2 segmenti inviati (se questi arrivano a J uno subito dopo l'altro e J non ha segmenti da inviare ad H), oppure uno ogni segmento inviato (se arrivano molto separati nel tempo tra loro oppure J ha segmenti da inviare ad H). Il TCP di H passerà nello stato di congestion avoidance dopo aver ricevuto il quarto riscontro. Per ricevere tale riscontro, H dovrà aver inviato almeno 4 segmenti ed al più 7 segmenti (1 quando $cwnd=1$ e 2 successivamente).

Q4. Indichiamo con TO_{CTS} il tempo che intercorre tra l'inizio della trasmissione di un RTS e la fine della ricezione dell'eventuale CTS. Allora (i tempi sono tutti in microsecondi), $TO_{CTS}=10$ (tempo di trasmissione di RTS)+3 (due ritardi di propagazione)+ 10 (tempo di trasmissione del CTS)+ 16 (SIFS)=39. Dal tempo t_0 , A dovrà attendere mediamente 50 (DIFS)+39 (Primo tentativo di RTS)+ 39 (prima che scada il timeout: $\text{timeout} = 78$)+16 (metà tempo di backoff, visto che la media sarà di 0,5)+50 (: ripete il protocollo – DIFS)+39 (secondo tentativo con successo)+16 (SIFS prima di iniziare la trasmissione del frame)=249 microsecondi.

Q5. ICMP che usa IP: quindi ICMP ed IP.

E1. Si ipotizzi che nel passaggio di stato tra FR e CA le variabili non vengano cambiate, quindi che non si ponga:

$$cwnd=ssthresh$$

Il valore di $ssthresh$ cambia solo in caso di timeout (che non si verificano nel nostro caso) oppure di ricezione di 3 ack duplicati. Quindi, siccome R_2 è il terzo ack duplicato, subito dopo la sua ricezione, $ssthresh$ cambia e diventa 2,5 e si passa nello stato di fast recovery. Quindi, subito prima $cwnd$ valeva 5 e subito dopo vale $5,5=11/2$. Al tempo t_1 , subito dopo la ricezione di R_3 (che riscontra 2 MSS di dati perché si invia 1 MSS di dati), $cwnd=11/2+2/11=125/22$. La ricezione di ack duplicati non modifica il valore delle variabili in gioco, e quindi al tempo t_0 si ha che $cwnd=5$, $rwnd \geq 5$ e $ssthresh > 5$ se si era nello stato di slow start, oppure $ssthresh=5$ se si era appena passati nello stato di congestion avoidance. In seguito, e fino a t_2 , il valore di $ssthresh$ non cambia e quindi rimane 2,5 anche a t_2 . Subito dopo la ricezione di R_4 , $cwnd=125/22+22/125=16109/2750$ e il valore di $rwnd$ trasportato da R_4 sarà di $4-Y+1$, se vengono riscontrati Y MSS di dati. Subito dopo la ricezione di R_5 $cwnd=16109/2750+2750/16109$ cioè circa 6, e il valore di $rwnd$ trasportato da R_5 sarà di $4-Y-Z+1,5$ MSS, se R_5 riscontra altri Z MSS di dati.

E2. Si suppone che la tabella abbia n righe. Si può supporre che se $\text{tab}[i]=\text{NULL}$, allora $\text{tab}[i].\text{costo}$ è infinito.

```
UDP_receive(j,tab);
```

```
for i=1 to n, i++
```

```
    { if (tab[i]!=NULL)&&((tabinoltro[i]==NULL) | |(tabinoltro[i].costo>tab[i].costo) | |(tabinoltro[i].nexthop==
        tab[i].nexthop)))
```

```
        {tabinoltro[i]=tab[i]}
```

```
    }
```

```
for i=1 to n, i++
```

```
    { tabxvicini[i].costo=tabinoltro[i].costo++;
```

```
      tabxvicini[i].nexthop=R;
```

```
    }
```

```
for i=1 to M, i++
```

```
    {UDP_send(i, tabxvicini)}
```