

RETI DI CALCOLATORI – prova scritta del 14/06/2016

Per essere ammessi alla prova orale è necessario ottenere una valutazione sufficiente della prima parte.

Prima parte (15 punti)

Q1. Un router A trasmette in modo continuato dati su un collegamento con un router B. Determinare –giustificando la risposta– quale è la lunghezza di tale collegamento se la frequenza di trasmissione è 100 Mbps, la velocità di propagazione è $2 \cdot 10^8$ m/s e al più 1000 bit possono essere simultaneamente presenti nel collegamento.

Q2. Il TCP di un host si trova nello stato di *fast recovery*, con *ssthresh*=2,5 MSS, *cwnd*=6,5 MSS, *rwnd*=3 MSS, e ha 3 MSS di dati in volo e 2 MSS di nuovi dati da spedire. Indicare –giustificando la risposta– il numero minimo di riscontri che il TCP deve ricevere per passare prima nello stato di *congestion avoidance*, quindi in *slow start* e infine tornare in *fast recovery*.

Q3. In una rete composta da nodi che utilizzano il protocollo *path vector*, il nodo X riceve un *path vector Path*, dal suo vicino Y. Indicare –giustificando la risposta– in quali casi X aggiornerà il suo cammino per la destinazione Z a seguito della ricezione di tale vettore.

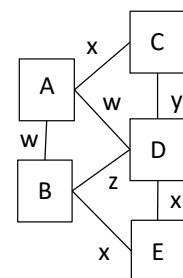
Q4. Il nodo A di una rete Ethernet con topologia a bus deve trasmettere per la prima volta un frame di 86 byte. Al tempo t_1 inizia a eseguire il protocollo MAC per trasmettere quel frame. Supponendo che al tempo t_1 il canale sia libero, e che risulti libero per un tempo sufficiente ad iniziare a trasmettere il frame, che i nodi utilizzino la variante 1-persistente del protocollo, che la frequenza di trasmissione sia 10 Mbps, che il ritardo di propagazione sia 2 microsecondi, che il jamming signal sia 48 bit, che T_{fr} sia pari al tempo necessario per trasmettere 792 bit, che il tempo di sensing sia pari al tempo necessario per trasmettere 12 byte, indicare –giustificando la risposta– quale è il massimo tempo t_2 a cui A termina di trasmettere con successo il suo frame, se questo incorre in una sola collisione.

Q5. Consideriamo un anello Chord che utilizza identificatori a 6 bit ed è formato dai nodi con identificatori 3, 6, 12, 22, 34, 45, 62. Indicare –giustificando la risposta– il cammino della richiesta per recuperare le informazioni associate rispettivamente alle chiavi 30, 44 e 1 assumendo che tutti e tre i cammini inizino nel nodo 6.

Seconda parte (15 punti)

E1. Modificare l'automa a stati finiti che descrive il comportamento del receiver di Selective Repeat in modo che invii un riscontro solo se riceve correttamente il segmento più vecchio tra quelli attesi. Il riscontro inviato deve inoltre contenere il numero di sequenza dell'ultimo segmento passato al livello superiore.

E2. Considerare la rete a lato, i cui nodi utilizzando l'algoritmo di routing Distance Vector con Poisoned Reverse. Indicare –giustificando la risposta– se è possibile o meno assegnare dei valori interi positivi a w , x , y e z in modo che nessuno dei cammini aventi origine e destinazione nei nodi A, B, C ed E passi per il nodo D.



Q1. I bit che possono essere simultaneamente presenti nel collegamento sono quelli che il router riesce a trasmettere mentre il segnale si propaga nel canale, ovvero $R \cdot d_{prop}$. Quindi $10^8 \frac{b}{s} \times \frac{d}{2 \times 10^8 \frac{m}{s}} = 1000 b$ e la lunghezza d del collegamento è 2000 m.

Q2. TCP passa da *fast recovery* a *congestion avoidance* se riceve un riscontro non duplicato. Se l'evento successivo è lo scadere di un timeout passa in *slow start*. Se a questo punto riceve tre riscontri duplicati torna in *fast recovery*. Quindi il numero minimo di riscontri che TCP deve ricevere per passare prima in *congestion avoidance*, quindi in *slow start* e infine tornare in *fast recovery* è 4.

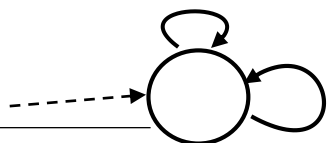
Q3. X aggiornerà il suo cammino $Path_x[Z]$ se $Path_y[Z]$ non contiene X e se X considera il cammino $X+Path_y[Z]$ migliore –secondo la politica adottata– del cammino $Path_x[Z]$ che aveva precedentemente determinato.

Q4. A attende un tempo di sensing (9,6 μs) quindi inizia a trasmettere il frame. Al massimo dopo 4 μs rileva l'avvenuta collisione, trasmette il jamming signal (4,8 μs), attende il valore massimo possibile del tempo di attesa (79,2 μs), attende di nuovo un tempo di sensing (9,6 μs) e quindi trasmette il frame (68,8 μs). Quindi il massimo tempo t_2 a cui A termina di trasmettere con successo il suo frame è $t_1 + (9,6+4+4,8+79,2+9,6+68,8) \mu s = t_1 + 176 \mu s$.

Q5. (a) Il nodo 6, i cui finger sono nell'ordine 12, 22 e 45, inoltra la richiesta al nodo 22. Il nodo 22, i cui finger sono nell'ordine 34, 45 e 62, determina che il nodo 34 è *successor(30)* dato che $30 \in (22,34]$. (b) Il nodo 6 inoltra la richiesta al nodo 22. Il nodo 22 inoltra la richiesta al nodo 34. Il nodo 34, il cui primo finger è 45, determina che il nodo 45 è *successor(44)* dato che $44 \in (34,45]$. (c) Il nodo 6 inoltra la richiesta al nodo 45. Il nodo 45, i cui finger sono nell'ordine 62 e 22, inoltra la richiesta al nodo 62. Il nodo 62, il cui primo finger è 3, determina che il nodo 3 è *successor(1)* dato che $1 \in (62,3]$.

E1.

rcvSgm=UDT_rcv() && corrupted(rcvSgm)



rcv_base=1

forEach y: isReceivedSgmt[y]=false

rcvSgm=UDT_rcv() && !corrupted(rcvSgm)

y=seqN(rcvSgm)

if (isReceivedSgmt[y]==false)

{rcvSgmt[y]=extract(rcvSgm); isReceivedSgmt[y]=true}

if (y==rcv_base) {

while (isReceivedSgmt[rcv_base]==true) do {

deliver_data(rcvSgmt[rcv_base])

isReceivedSgmt[y]=false

rcv_base++

}

sndSgm = make_segment(ACK, rcv_base-1)

UDT_send(sndSgm)

}

E2. Affinché nessuno dei cammini aventi origine e destinazione nei nodi A, B, C ed E passi per il nodo D dovrebbero valere le seguenti disequazioni:

(1) $x + w < y$ (affinché, per esempio, il cammino da C a E non passi per D), e

(2) $w + x < w + x$ (affinché, per esempio, il cammino da A a E non passi per D).

Non è quindi possibile assegnare dei valori interi positivi a w , x , y e z in modo da garantire che nessuno dei cammini aventi origine e destinazione nei nodi A, B, C ed E passi per il nodo D.

¹ Utilizziamo la notazione (a,b] per indicare la porzione dell'anello ottenuta muovendosi in senso orario da a+1 a fino a b.