

## RETI DI CALCOLATORI – prova scritta del 16/01/2014

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

### Prima parte (10 punti)

**Q1.** Indicare – giustificando la risposta – quali problemi potrebbero verificarsi se un cliente FTP appartenente a una rete NAT utilizzasse la stessa porta locale TCP per connettersi con due server diversi non appartenenti ad alcuna rete NAT.

**Q2.** Indicare – giustificando la risposta – se è possibile o meno che un sender Selective Repeat con dimensione della finestra uguale a 3 e con 2 segmenti “in volo” non possa inviare segmenti contenenti nuovi dati dopo avere ricevuto un riscontro non duplicato per il segmento più vecchio tra quelli “in volo”.

**Q3.** Supponiamo che un cliente HTTP C appartenente a una rete IEEE 802.11 invii una richiesta R a un server S. Indicare – giustificando la risposta – in che modo C potrebbe verificare che R non è stata ricevuta da S se CSMA/CA non prevedesse l'invio di ACK.

**Q4.** Indicare – giustificando la risposta – quanti nodi devono essere contattati nel caso pessimo per trovare il successore di una chiave in un anello Chord formato da N nodi che utilizzano “Scalable Key Location”.

### Seconda parte

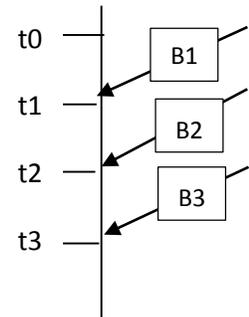
**E1 (5 punti).** Considerare una versione “loss-tolerant” di Go-Back-N, in cui il sender non reinvia una copia di un segmento di cui non ha ricevuto riscontro se il segmento precedente è stato riscontrato (e in cui il receiver restituisce al livello superiore un flusso ordinato di dati senza però attendere di ricevere dal sender un segmento mancante se il segmento precedente è stato correttamente ricevuto). Descrivere –utilizzando un automa a stati finiti– il comportamento del sender di tale versione “loss-tolerant” di Go-Back-N.

**E2 (5 punti).** Supponiamo che al tempo  $t_0$  il TCP di un processo applicativo A si trovi nello stato di slow start, abbia 4 MSS di dati in volo e che la dimensione della sua finestra di congestione sia 3 MSS. Supponiamo che il TCP di A riceva tre ACK non duplicati (B1, B2, B3) ciascuno riscontrante nuovi dati. Indicare – giustificando la risposta – quanti nuovi dati potrà spedire il TCP di A in  $t_3$ :

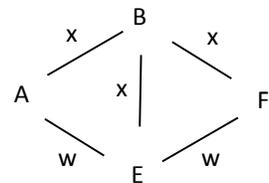
(a) nel caso in cui il valore della finestra di congestione in  $t_2$  sia  $17/4$  MSS, e

(b) nel caso in cui il valore della finestra di congestione in  $t_1$  sia  $10/3$  MSS.

Assumiamo che non scatti alcun timeout nell'intervallo  $[t_0, t_3]$ .



**E3 (5 punti).** Consideriamo la rete a lato in cui i nodi utilizzano l'algoritmo distance vector con poisoned reverse e in cui  $x$  e  $w$  sono interi positivi. Determinare – giustificando la risposta – quale relazione deve valere tra  $x$  e  $w$  affinché, quando la rete ha raggiunto lo stato di quiescenza, nessuna delle ultime copie dei vettori che B ha ricevuto dai suoi vicini contenga distanze “avvelenate”.



**E4 (5 punti).** Supponiamo che solo tre nodi (A, B, C) di una rete Ethernet con topologia a bus debbano trasmettere ciascuno un frame e che inizino tutti e tre a trasmettere simultaneamente il proprio frame: A per la prima volta, B e C per la seconda volta. Indicare – giustificando la risposta – quale è la probabilità che almeno uno dei tre nodi riesca a trasmettere con successo il suo frame solo subito dopo che si sono verificate tre collisioni tra tutti e tre i nodi.

**Q1.** Il cliente non avrà problemi a utilizzare la stessa porta locale per connettersi con i due server. Il router NAT provvederà infatti a tradurre i pacchetti inviati dal client sostituendo l'indirizzo IP sorgente  $IP_c$  e la porta sorgente  $p_c$  con il proprio indirizzo IP pubblico  $IP_e$  e con una porta esterna  $p_e$  ed effettuerà la traduzione inversa per i pacchetti<sup>(1)</sup> destinati all'indirizzo  $IP_e$  porta  $p_e$ . Il TCP del cliente distinguerà le due connessioni dagli indirizzi IP (diversi) dei due server.

**Q2.** La ricezione di un riscontro per il segmento più vecchio ancora in volo provoca sempre lo "scorrimento" della finestra. Se il sender ha nuovi dati da spedire potrà quindi certamente inviare almeno un segmento contenente nuovi dati.

**Q3.** Dato che HTTP utilizza TCP come protocollo di trasporto, l'eventuale non ricezione di R da parte di S verrà rilevata dal TCP di C (che non riceverà il riscontro del segmento contenente R).

**Q4.** Nel caso pessimo nodi e chiavi sono distribuiti sull'anello in modo non bilanciato e la scalable key location non riesce a "saltare" alcun nodo, contattando così N-1 nodi per trovare il successore di una chiave<sup>(2)</sup>.

**E1.**

```

rdt_send(data)
if (nextseqnum < base+N) {
    sndpkt[nextseqnum] = make_pkt(nextseqnum,data)
    udt_send(sndpkt[nextseqnum])
    if (base == nextseqnum)
        start_timer
    nextseqnum++
}
else
    refuse_data(data)

```

```

base=1
nextseqnum=1
flag=false

rcvt=udt_rcv() &&
(corrupt(rcvpkt) ||
! isACKinWindow(rcvpkt))

rcvpkt=udt_rcv() && notcorrupt(rcvpkt) && isACKinWindow(rcvpkt)
base = getacknum(rcvpkt)+1
flag=false
If (base == nextseqnum) stop_timer else start_timer

```

```

timeout
If flag then udt_send(sndpkt[base])
else { base++;
    if (base == nextseqnum) stop_timer
    else start_timer
}
flag=true
}
udt_send(sndpkt[base+1])
...
udt_send(sndpkt[nextseqnum-1])
start_timer

```

*/\* Per semplicità assumiamo che il primo segmento inviato debba sembrare essere rispedito nel caso in cui scatti il timeout relativo. \*/*

**E2.** Determiniamo il valore della finestra di congestione in  $t_3$  ( $cwnd_{t_3}$ ) nei due scenari proposti.

- Affinché  $cwnd_{t_2}$  sia  $17/4$  MSS,  $cwnd$  deve essere stato incrementato di 1 MSS all'arrivo di B1 ( $cwnd_{t_1} = 3 \text{ MSS} + 1 \text{ MSS} = 4 \text{ MSS}$ ) e quindi di  $1/4$  MSS all'arrivo di B2 ( $cwnd_{t_2} = 4 \text{ MSS} + 1/4 \text{ MSS} = 17/4 \text{ MSS}$ ) e quindi il TCP di A si troverà nello stato di congestion avoidance quando riceverà B3 ed incrementerà quindi  $cwnd$  di  $4/17$  MSS, ovvero  $cwnd_{t_3} = 17/4 \text{ MSS} + 4/17 \text{ MSS} = 305/68 \text{ MSS}$ .
- Se invece  $cwnd_{t_1} = 10/3$  MSS allora  $cwnd$  è stato incrementato di  $1/3$  MSS all'arrivo di B1 ( $cwnd_{t_1} = 3 \text{ MSS} + 1/3 \text{ MSS} = 10/3 \text{ MSS}$ ) e quindi il TCP di A si troverà nello stato di congestion avoidance quando riceverà sia B2 che B3 e incrementerà  $cwnd$  prima di  $3/10$  MSS ( $cwnd_{t_2} = 10/3 \text{ MSS} + 3/10 \text{ MSS} = 109/30 \text{ MSS}$ ) e quindi di  $30/109$  MSS ( $cwnd_{t_3} = 109/30 \text{ MSS} + 30/109 \text{ MSS} = 12781/3270 \text{ MSS}$ ).

In entrambi i casi la quantità di nuovi dati che il TCP di A potrà spedire in  $t_3$  (se ne ha, ovviamente) non potrà essere superiore alla differenza tra il minimo tra  $cwnd_{t_3}$  e l'ultimo valore della finestra di ricezione  $rwnd$  ricevuto, e la quantità di byte ancora in volo (che sarà pari alla quantità di dati contenuta nell'unico segmento ancora in volo in  $t_3$  oppure a 0 se il TCP di A non ha dati in volo in  $t_3$ ).

**E3.** I vettori ricevuti da B conterranno il minimo dei valori indicati in ogni casella della tabella a lato, dove i valori evidenziati in grassetto causeranno l'avvelenamento della distanza. Affinché nessuna delle ultime copie dei vettori che B ha ricevuto dai suoi vicini contenga distanze "avvelenate", deve quindi valere  $2x > w$  e  $x > w$ , ovvero  $x > w$ .

	A	E	F
A	-	$\min(2x, w)$	$\min(2x, 2w)$
E	$\min(2x, w)$	-	$\min(2x, w)$
F	$\min(2x, 2w)$	$\min(2x, w)$	-

**E4.** La probabilità che si verifichino tre collisioni tra tutti e tre i nodi è  $1 \times \frac{2}{2 \times 4 \times 4} \times \frac{4}{4 \times 8 \times 8} = \frac{1}{2^{10}}$  e la probabilità che a quel punto almeno uno riesca a trasmettere subito con successo è  $\frac{(15 \times 15) + (7 \times 15) + (7 \times 15)}{8 \times 16 \times 16} = \frac{435}{2^{11}}$ . Quindi la probabilità che almeno uno dei tre nodi riesca a trasmettere con successo il suo frame solo subito dopo che si sono verificate tre collisioni tra tutti e tre i nodi è  $\frac{1}{2^{10}} \times \frac{435}{2^{11}} = \frac{435}{2^{21}}$ .

(1) Una volta stabilite le due connessioni potrebbero verificarsi problemi se il cliente non inviasse il comando PASV e i server cercassero di aprire una connessione dati con il cliente e il server NAT non accettasse sessioni iniziate dall'esterno.

(2) Per esempio in un anello di dimensione  $2^6$  contenente solo i nodi N1, N9, N13 e N15, se N1 inizia la ricerca del successore della chiave 16 tutti gli altri nodi dell'anello verranno contattati.