RETI DI CALCOLATORI – Secondo appello, a.a. 2009/2010

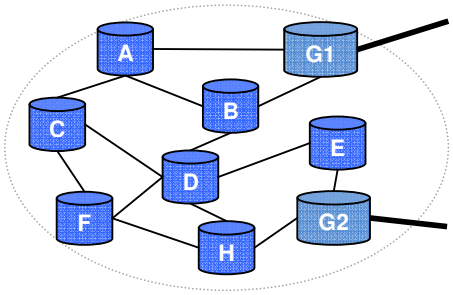
La prova è strutturata in due parti. Per ottenere una valutazione sufficiente dell'intera prova è necessario ottenere una valutazione sufficiente della prima parte. In accordo con quanto deliberato dal Consiglio di Facoltà il 15.12.2009, ogni studente può partecipare a tutti i cinque appelli previsti e consegnare al più quattro prove scritte.

*Prima parte (**10 punti**)*

Q1. Supponiamo che la velocità con cui il segnale si propaga su un collegamento lungo 2 chilometri tra due router A e B sia di $3 \cdot 10^8$ m/s. Indicare – giustificando la risposta – quale deve essere la frequenza con cui A trasmette i dati sul collegamento affinché il primo bit di un pacchetto inviato da A arrivi a B nel momento in cui il terzo bit dello stesso pacchetto viene immesso nel collegamento.

Q2. Indicare – giustificando la risposta – se è possibile o meno che il TCP di un processo applicativo A abbia n MSS byte di dati "in volo" mentre la sua finestra di congestione ha un valore minore di n MSS byte.

Q3. Supponiamo che il sistema autonomo S rappresentato di lato utilizzi RIP come protocollo di instradamento interno al sistema autonomo e sia r una sottorete esterna a S non ancora presente nella tabella di inoltro del router C. Supponiamo che C riceva dal gateway G2 l'annuncio di una rotta per r il cui AS-PATH contiene n sistemi autonomi e subito dopo C riceva da G1 l'annuncio di una rotta per r il cui AS-PATH contiene (n+2) sistemi autonomi. Indicare - giustificando la risposta - in che modo C aggiorna la sua tabella di inoltro se i router di S utilizzano il cosiddetto instradamento "a patata bollente" (hot potato routing).

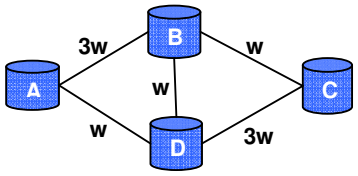


Q4. Consideriamo un'estensione "sicura" di SMTP che utilizzi SSL come servizio di trasporto e supponiamo che un host A debba trasferire un messaggio di email al mailserver B di cui conosce l'indirizzo IP. Indicare – giustificando la risposta – quanti pacchetti IP devono essere inviati da A a B prima che A possa iniziare a inviare a B i dati contenuti nel corpo del messaggio.

Seconda parte

E1 (6 punti). (a) Supponiamo che al tempo t il TCP di un processo applicativo A abbia nel suo buffer di ricezione solo 2 MSS di dati che non sono ancora stati letti da A e che abbia ricevuto tali dati dal suo pari in due segmenti recanti numero di sequenza X e X+2MSS rispettivamente. Supponiamo inoltre che il TCP di A riceva dal suo pari altri 2 MSS di dati: il primo al tempo $t+\Delta$ in un segmento S1 recante numero di sequenza X+1MSS e il secondo al tempo $t+2\Delta$ in un segmento S2 recante numero di sequenza X+4MSS. Supponendo infine che nell'intervallo $[t,t+2\Delta]$ il TCP di A riceva solo i segmenti S1 e S2 e A non legga nessun dato dal buffer, e che l'ultimo valore di RcvWin inviato dal TCP di A prima di t fosse K. Indicare – giustificando la risposta – il valore dei campi AckNumber e RcvWin contenuti nei riscontri A1 e A2 inviati dal TCP di A in risposta ai segmenti S1 e S2. **(b)** Descrivere, esplicitando il signficato delle variabili utilizzate, il modo in cui il TCP di A determina il valore di RcvWin che invierà al suo pari P dopo avere ricevuto un segmento da P.

E2 (6 punti). Consideriamo la rete a lato i cui nodi utilizzano il protocollo distance vector con poisoned reverse. **(a)** Indicare i vettori delle distanze e la tabella di inoltro di B una volta che la rete ha raggiunto lo stato di quiescenza. **(b)** Supponendo che, una volta raggiunto lo stato di quiescenza, il costo del collegamento BC diventi 8w, indicare il modo in cui B modifica la sua tabella di inoltro non appena rileva che il costo di tale collegamento e' aumentato.



E3 (4 punti). Consideriamo una rete Ethernet in cui solo tre nodi devono spedire dati. Supponendo che tutti e tre i nodi inizino simultaneamente a trasmettere e che quindi collidano, indicare –giustificando la risposta– quale è la probabilità che collidano tutti e tre di nuovo insieme nei due successivi tentativi di spedizione che effettueranno.

E4 (4 punti). Consideriamo il protocollo descritto a lato, dove n è un *nonce* generato da A. Indicare –giustificando la risposta- se il protocollo permette ad A di autenticare B.

1. *A invia a B:* $< n >$
2. *B invia a A:* $< K^+_A(B, K^+_B),\ K^+_A(K^-_B(n)) >$

TRACCIA DELLA SOLUZIONE

Q1. Il tempo necessario affinché il primo bit del pacchetto raggiunga B è $1/R + d_{prop}$ (dove R è la frequenza di trasmissione e d_{prop} è il ritardo di propagazione sul collegamento), mentre il tempo necessario affinché il terzo bit del pacchetto sia immesso nel collegamento è 3/R. Deve quindi valere $1/R + d_{prop} = 3/R$ ovvero (dato che d_{prop} è il rapporto tra la lunghezza del collegamento e la velocità di propagazione del segnale sul collegamento) R= 300 Kbps.

Q2. Sì, è possibile. TCP può infatti avere spedito n MSS di byte mentre CongWin aveva un valore maggiore o uguale di n MSS ed avere successivamente ridotto la dimensione di CongWin (dimezzandola o riportandola al valore iniziale di 1 MSS) dopo avere rilevato un evento di perdita (ricezione di un terzo riscontro duplicato oppure lo scadere di un timeout).

Q3. Dato che la distanza da C a G1 è minore di quella da C a G2, C aggiungerà alla sua tabella di inoltro la coppia (r,I) dove I è l'interfaccia di C sulla sottorete a cui appartengono sia C che A.

Q4. Prima di poter iniziare a inviare i dati contenuti nel corpo del messaggio di email, A dovrà inviare 2 pacchetti IP (contenenti il segmento di "syn" e il riscontro del "syn+ack", rispettivamente) per completare la fase di handshake di TCP, 2 pacchetti IP[1] (per trasmettere il comando "SSL hello" e il valore master segreto cifrato, rispettivamente) per completare la fase di handshake di SSL[2] e infine 4 pacchetti IP per trasmettere i comandi "HELO", "MAIL FROM", "RCPT TO" e "DATA" previsti da SMTP.

E1. (a) Se il TCP di A utilizza la formula

dimensioneBuffer – (highestByteReceived – lastByteRead) *(*)*

per determinare il valore di RcvWin[3] da inviare al suo pari – dove *lastByteRead* indica il numero dell'ultimo byte nel flusso di dati che è stato letto da A e *highestByteReceived* indica il numero del byte col numero più alto tra quelli ricevuti – allora (assumendo che A non legga nessun dato dal buffer dopo che il TCP di A ha inviato l'ultimo valore RcvWin=K prima di t):

A1.AckNumber = X + 3MSS[4], A1.RcvWin=K e A2.AckNumber = X + 3MSS e A2.RcvWin=K-2MSS.

(b) Supponiamo che il TCP di A utilizzi le variabili *lastByteRead* e *highestByteReceived* descritte al punto precedente. Quando riceve un segmento (non corrotto) contenente i dati corrispondenti all'intervallo [Y,Y+D] nel flusso di dati, il TCP di A aggiorna il valore di *highestByteReceived* nel modo seguente:

se ((non aveva ancora ricevuto tali dati) **e** (highestByteReceived<Y+D))
allora highestByteReceived=Y+D

Il valore di RcvWin da inviare al pari verrà quindi determinato applicando direttamente la formula (*) descritta al punto precedente.

E2. (a)

	A	C	D
B	2w	w	w
A	0	3w	w
C	∞	0	∞
D	w	∞	0

dest	next
A	D
C	C
D	D

(b)

dest	next
A	D
C	*A*
D	D

E3. Dopo che si è verificata la prima collisione ciascuno nodo attenderà K·512bit, con K valore intero scelto in modo casuale nell'intervallo $[0,2^1-1]$, prima di tentare nuovamente la spedizione. La probabilità che collidano tutti e tre di nuovo al secondo tentativo è quindi 1/4 (ovvero la probabilità che tutti e tre scelgano 0 oppure 1). In modo del tutto analogo la probabilità che collidano tutti e tre di nuovo anche al terzo tentativo è 1/16. La probabilità che collidano tutti e tre di nuovo nei due successivi tentativi è quindi 1/64.

E4. No, il protocollo descritto non permette ad A di autenticare B. Tale protocollo è infatti suscettibile a un attacco "man-in-the-middle" in cui un intruso T intercetta il *nonce* inviato da A e invia quindi ad A il messaggio: $< K^+_A(B, K^+_T),\ K^+_A(K^-_T(n)) >$ che potrebbe indurre A a considerare K^+_T la chiave pubblica di B autenticando in questo modo erroneamente T come B.

---

[1] Trascuriamo per semplicità la possibilità che alcuni dati vengano trasportati in *piggybacking*.

[2] Facendo riferimento alla versione semplificata di SSL descritta nel paragrafo 8.6.1 del testo adottato. In realtà il primo messaggio SSL contiene anche il *nonce* generato da A e la lista degli algoritmi crittografici utilizzati da A, ed è inoltre previsto un terzo messaggio contenente un MAC dei messaggi di *hanshake* scambiati.

[3] Un'alternativa è utilizzare una variabile per memorizzare esplicitamente la quantità esatta di byte presenti nel buffer e non ancora letti.

[4] Nell'ipotesi che X=lastByteRead+1. Altrimenti AckNumber dovrà essere lastByteRead+1 in entrambi i riscontri A1 e A2.