

Reti di calcolatori (a.a. 2006/07 – primo appello)

Per la soluzione usare al più un foglio protocollo, indicando in alto e in STAMPATELLO: cognome, nome, numero di matricola e corso (A o B). Non è consentito usare materiale didattico di alcun tipo.

Quesiti - Rispondere in maniera concisa ma esauriente ai seguenti quesiti.

- Indicare tutte le invocazioni di primitive TCP che un server HTTP deve effettuare per ricevere una richiesta da un cliente.
- Consideriamo un router R che gestisce tutto il traffico in entrata e in uscita da una rete locale. Consideriamo un pacchetto IP inviato da un host A della rete locale a un server Web esterno a tale rete e contenente un segmento SYN. Indicare le informazioni relative all'indirizzamento contenute in tutti i preamboli del frame contenente il pacchetto IP ricevuto da R e ritrasmesso da R.
- Indicare che relazione esiste tra SOAP e HTTP.

Esercizio 1.

Consideriamo un router che utilizza il protocollo distance vector (senza poisoned reverse). Supponiamo per semplicità che i nodi della rete siano rappresentati dagli interi $[0, 1, \dots, V-1, V, \dots, N-1]$ dove 0 è il router in oggetto e $[1, \dots, V-1]$ sono i vicini del router. Supponiamo inoltre che il router utilizzi un array C di V interi per rappresentare i costi dei collegamenti con i suoi vicini e un array D di $V \times N$ interi per memorizzare i vettori delle distanze (dove $D[i, j]$ indica la distanza del vicino i dalla destinazione j). **Scrivere lo pseudo-codice** eseguito dal router in seguito alla variazione del costo di un collegamento con un suo vicino.

Esercizio 2.

Supponiamo che un'applicazione A desideri inviare 6 messaggi, ciascuno di 1 MSS, su una connessione TCP appena stabilita con un suo pari B. Supponiamo inoltre che:

- solo il secondo e il quarto pacchetto IP inviato dall'host di A vadano persi;
- la lunghezza del timeout del TCP di A sia maggiore di $2 \times RTT$;
- i processi TCP coinvolti bufferizzino i segmenti "non in ordine" ricevuti e che lo spazio libero nei buffer sia inizialmente di 10 MSS.

Descrivere tutti i possibili segmenti TCP scambiati dai due host, evidenziando di ogni segmento numero di sequenza, numero di riscontro, eventuali bit di controllo attivi e dimensione dei dati contenuti nel segmento.

Traccia della soluzione

Quesiti

- a) Il server dovrà farsi assegnare dal sistema operativo la porta 80, quindi attendere di ricevere una richiesta di connessione su tale porta e infine (una volta creata la connessione) ricevere una richiesta (HTTP) su tale connessione. Ad esempio, usando Java le corrispondenti invocazioni sono:

```
ServerSocket ss = new ServerSocket(80);  
Socket s = ss.accept();  
r = br.readLine();
```

(dove br è un BufferedReader associato all'InputStream del socket).

- b) Supponendo che l'host A e il router R siano direttamente collegati, il frame ricevuto da R conterrà nei vari preamboli le seguenti informazioni relative all'indirizzamento:

SRC: INDIRIZZO MAC DI A DEST: INDIRIZZO MAC DELL'INTERFACCIA DI R SULLA RETE LOCALE DI A	}	preambolo DL
SRC: INDIRIZZO IP DI A DEST: INDIRIZZO IP DEL SERVER WEB		
SRC: PORTA TCP DEL PROCESSO DI A DEST: PORTA TCP DEL SERVER WEB (80)	}	preambolo TCP

Il frame inviato dal router R al di fuori della rete conterrà lo stesso pacchetto IP ma un diverso preambolo, ovvero:

SRC: INDIRIZZO MAC DELL'INTERFACCIA DI R SULLA RETE SU CUI È R' DEST: INDIRIZZO MAC DELL'INTERFACCIA DI R' SULLA RETE SU CUI È R	}	preambolo DL
---	---	--------------

dove R' è il router a cui R indirizza i pacchetti destinati al server Web in oggetto.

- c) HTTP può essere utilizzato da SOAP come protocollo di trasporto (ovvero i messaggi SOAP possono essere trasportati all'interno di richieste HTTP).

Esercizio 1.

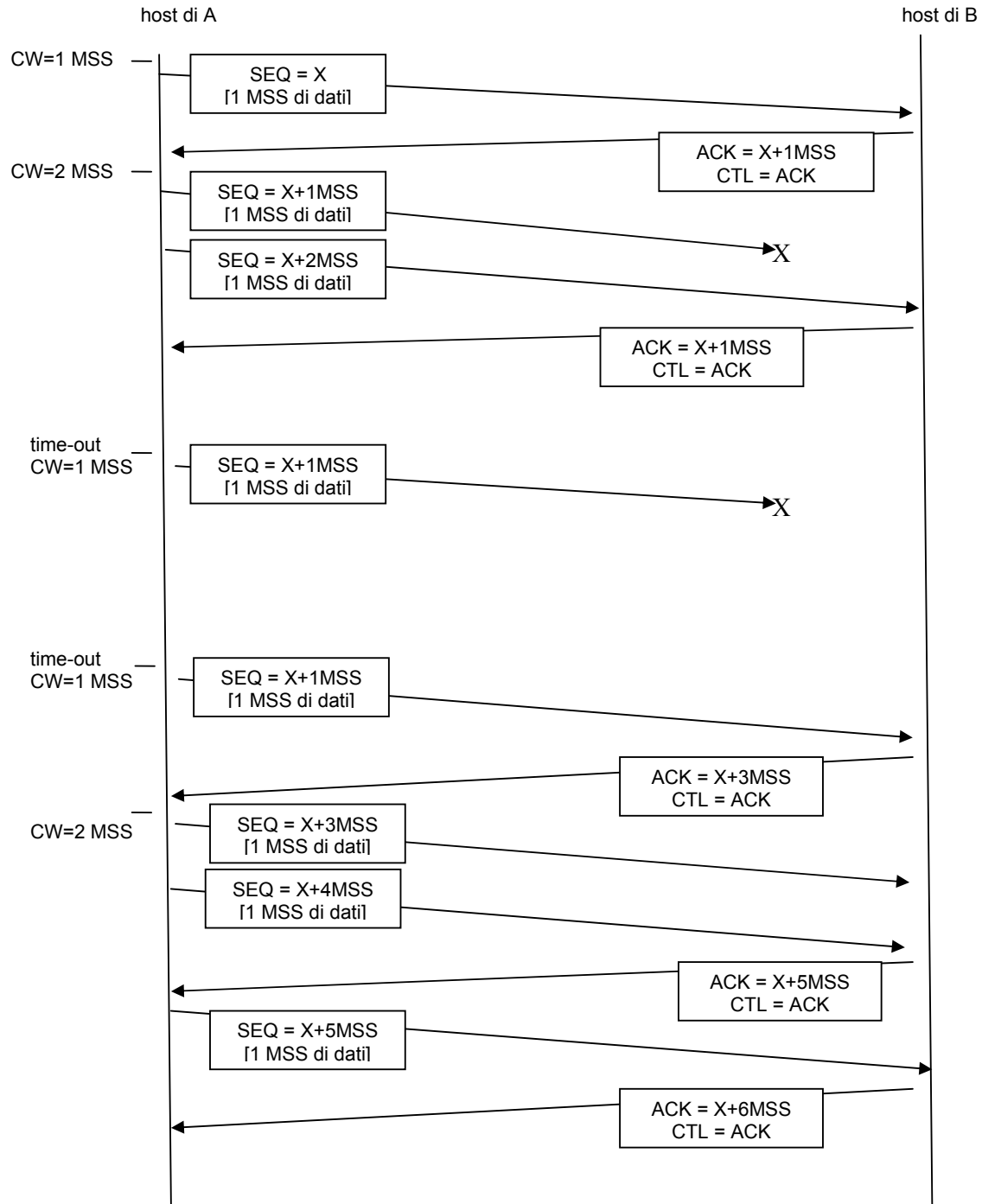
```
minChanged = 0;  
for (j=1; j<N; ++j) {                                // per ogni destinazione j  
    min = MAX_INT;  
    for (i=1; i<V; i++)                                // per ogni vicino i  
        if (C[i]+D[i,j] < min)                        // applica equazione di Bellman-Ford  
            min = C[i]+D[i,j];  
    if (min != D[0,j]) {  
        D[0,j] = min;  
        minChanged = 1;  
    }  
}  
if (minChanged == 1)                                // se è cambiata una distanza minima  
    for (i=1; i<V; i++)  
        "invia al vicino i il vettore D[0]"
```

Esercizio 2.

Supponiamo che X sia il numero di sequenza iniziale scelto in modo casuale dall'host dell'applicazione A.

[A causa del controllo di congestione di TCP, l'host di A invierà inizialmente un solo segmento contenente 1MSS di dati (il primo messaggio dell'applicazione A). Quando riceverà dall'host di B il riscontro di tale segmento, l'host di A invierà due nuovi segmenti contenenti 1MSS di dati, con numeri di sequenza X+1MSS e X+2MSS rispettivamente. Dato che – per ipotesi – il secondo pacchetto IP inviato dall'host di A andrà perso, l'host di B riceverà solo il segmento con numero di sequenza X+2MSS e invierà quindi un unico riscontro (negativo). Allo scadere del timeout, l'host di A invierà di nuovo il segmento con numero di sequenza X+1MSS che –per ipotesi– andrà di nuovo perso. Allo scadere del timeout, l'host di A invierà quindi di nuovo il segmento con numero di sequenza X+1MSS e riceverà quindi (dato che i segmenti "non in ordine" ricevuti vengono bufferizzati) un segmento di riscontro cumulativo con numero di riscontro X+3MSS. A questo punto l'host di S invierà due nuovi segmenti contenenti 1MSS di dati, con numeri di

sequenza $X+3MSS$ e $X+4MSS$ rispettivamente e, quando riceverà il relativo riscontro, potrà inviare l'ultimo segmento con numero di sequenza $X+5MSS$.]



(A. Brogi)