

Reti di calcolatori (a.a. 2006/07 – secondo 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.

- Q1) Indicare quali sono le invocazioni di primitive (offerte da altri servizi) effettuate dal TCP di un host X quando un'applicazione in esecuzione su X chiede di stabilire una connessione TCP con un mailserver SMTP.
- Q2) Spiegare che relazione intercorre tra il comando `LIST` di FTP e i socket TCP.
- Q3) Spiegare in che cosa consiste un attacco DoS.

Esercizio 1.

Consideriamo un router che utilizza il protocollo link state. Supponiamo per semplicità che i nodi della rete siano rappresentati dagli interi $[0, 1, \dots, N-1]$ dove 0 è il router in oggetto.

a) **Scrivere** (e commentare) il **codice** eseguito dal router per determinare le sue distanze minime dagli altri nodi, rispettando le seguenti specifiche:

- utilizzare un array `C` di $N \times N$ interi per rappresentare i costi (noti) dei collegamenti diretti tra i nodi (dove $C[i,j] = \text{MAX_INT}$ se i e j non sono direttamente collegati);
- utilizzare un array `D` di N interi per calcolare le distanze minime dagli altri nodi;
- utilizzare un array `K` di N interi per tenere traccia dei nodi la cui distanza minima è stata determinata.

b) **Completare il codice** con i comandi eseguiti dal router per determinare la sua tabella di inoltra, rappresentata da un array `R` di $N \times 2$ interi, dove $R[i,0]$ indica la distanza dal nodo 0 al nodo i , mentre $R[i,1]$ indica il nodo a cui 0 invia i pacchetti destinati al nodo i .

Esercizio 2.

Supponiamo che un'applicazione A desideri inviare 5 messaggi, ciascuno di 1 MSS, su una connessione TCP appena stabilita con un suo pari B. Supponiamo inoltre che:

- solo il secondo pacchetto IP inviato dall'host di A vada perso;
- la lunghezza del timeout del TCP di A sia maggiore di $3RTT$;
- la dimensione iniziale di entrambe le finestre di congestione sia di 2MSS;
- i processi TCP coinvolti bufferizzino i segmenti "non in ordine" ricevuti e che lo spazio libero in entrambi i buffer di ricezione sia di 10 MSS.

a) **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.

b) In maniera analoga **descrivere tutti i possibili** segmenti scambiati dai due host nello scenario sopra descritto, supponendo però che la dimensione iniziale di entrambe le finestre di congestione sia di 5MSS (anziché 2MSS).

Traccia della soluzione

Quesiti

Q1) Quando un'applicazione in esecuzione su un host X chiede di stabilire una connessione TCP con un server Y, il TCP di X dovrà cercare di effettuare un "three-way handshake" col TCP di Y, inviando un segmento S1 di syn, ricevendo un segmento S2 di syn+ack e infine inviando un segmento S3 per riscontrare la ricezione di S2. La spedizione e la ricezione di segmenti avverrà utilizzando il servizio sottostante IP, per cui le invocazioni effettuate dal TCP di X saranno del tipo:

```
IP_send (indirizzo_server, 6, S1);
<indirizzo_server,S2> = IP_receive(6);
IP_send (indirizzo_server, 6, S3);
```

Q2) Il comando LIST è uno dei messaggi FTP che due pari FTP possono scambiarsi su una connessione TCP (socket) - dopo ovviamente averla stabilita.

Q3) Un attacco DOS ("Denial Of Service") rende un server (o un'intera rete) non disponibile agli utenti autorizzati. In genere un attacco DoS consiste nell'obbligare un server (o una infrastruttura) a svolgere una tale quantità di lavoro (inutile) da non riuscire più a offrire i propri servizi. Una tecnica ben nota è ad esempio il cosiddetto "syn flooding" con cui l'aggressore inonda un server di segmenti TCP di tipo syn, obbligandolo a svolgere soltanto handshakes (inutili).

Esercizio 1

/ Inizializza il vettore K dei nodi "colorati" (la cui distanza minima dal nodo 0 e' gia' stata determinata), il vettore D delle distanze minime dal nodo 0 e un vettore Pred con cui memorizzare i predecessori lungo il cammino.*/*

```
for (i=1;i<N;i++){
    K[i]=0;           // nodo i non colorato
    D[i]=C[0,i];
    Pred[i]=0;
}
```

/ Effettua (N-1) iterazioni, in cui a ogni iterazione determina un nodo non colorato w tale che D[w] e' la distanza minima da 0 a un nodo non colorato. */*

```
for (i=1;i<N;i++){
    min=MAX_INT;
    for (j=1;j<N;j++){
        if (!K[j]&&D[j]<min){
            min=D[j];
            w=j;
        }
    }
    K[w]=1;           // colora il nodo w
    // aggiorna le distanze da 0 a tutti i vicini di w
    for (v=1;v<N;v++){
        if (C[w,v]!=MAX_INT && K[v]==0)           //se v è vicino di w
            if (D[v]>D[w]+C[w,v]){
                D[v]=D[w]+C[w,v];
                Pred[v]=w;           //aggiorna il predecessore di v
            }
    }
}
```

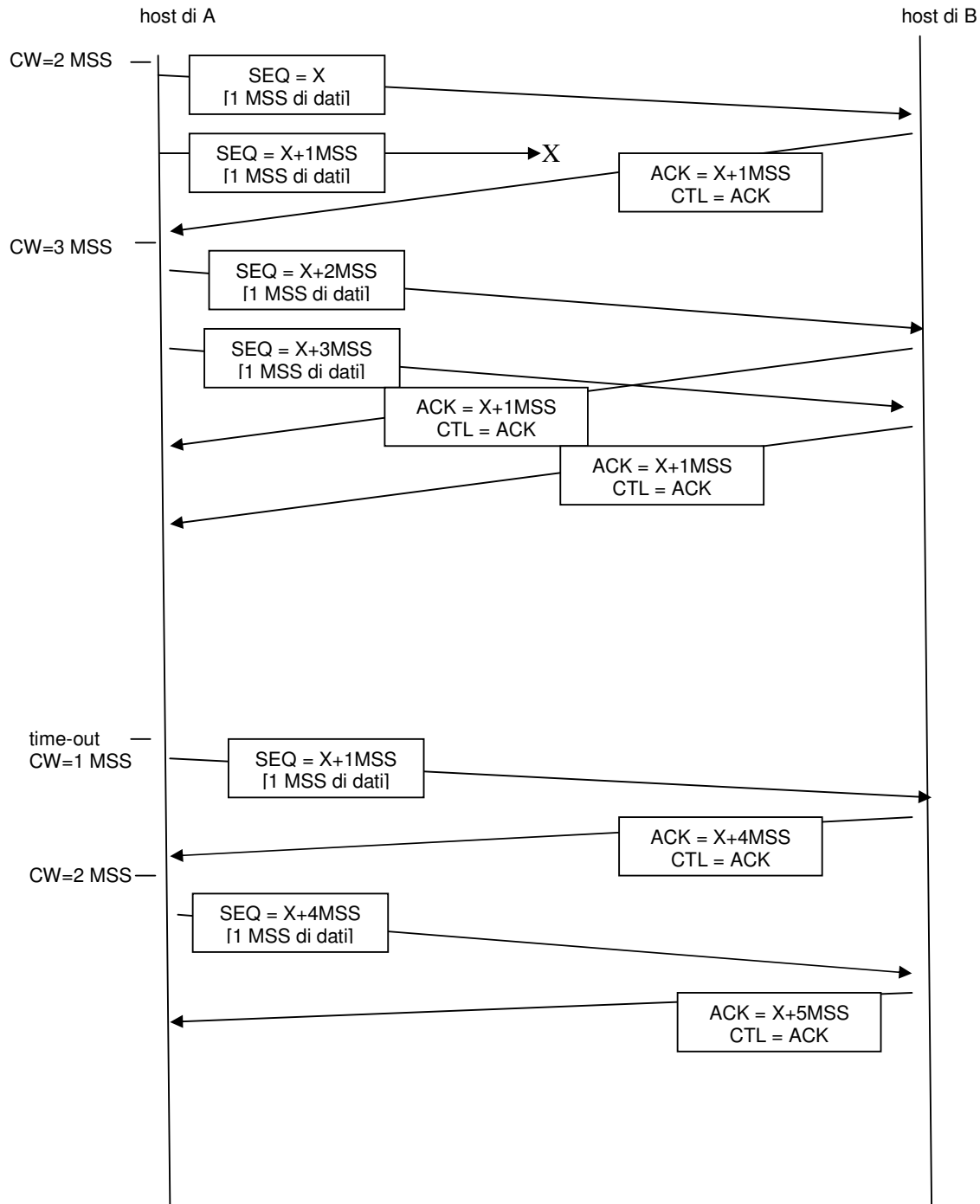
/ Determina tabella di inoltro*

```
for (i=1;i<N;i++){
    R[i,0]=D[i];
    if (Pred[i]==0)
        R[i,1]=i;
    else {p=Pred[i];
        /*cerca il vicino di 0 a cui instradare i pkt destianti a i*/
        while (Pred[p]!=0) p=Pred[p];
        R[i,1]=p;
    }
}
```

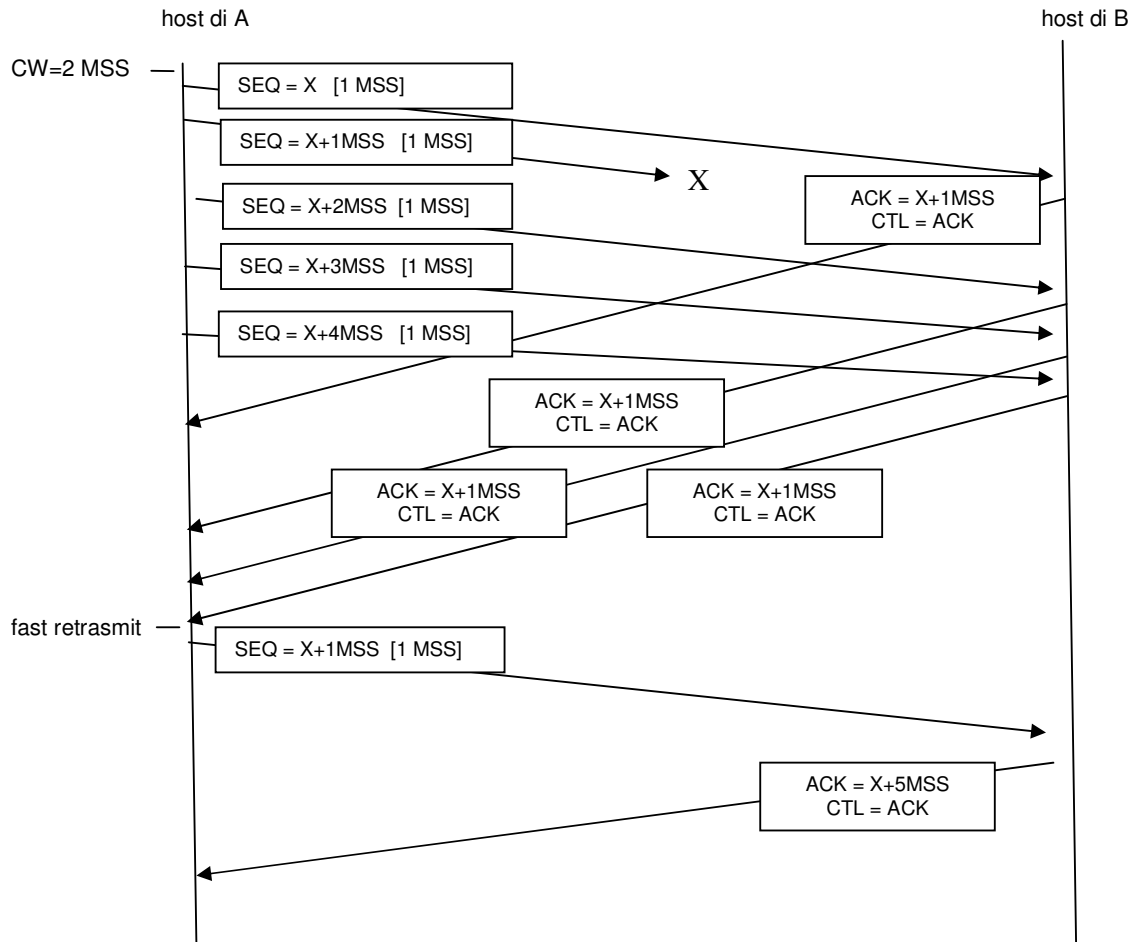
Esercizio 2

a) A causa del controllo di congestione di TCP, l'host di A invierà inizialmente solo due segmenti contenenti 1MSS di dati ciascuno. Quando riceverà dall'host di B il riscontro del primo segmento, l'host di A incrementerà di 1 MSS la dimensione della sua finestra di congestione e invierà quindi altri due nuovi segmenti contenenti 1 MSS di dati. A questo punto l'host di A riceverà due riscontri negativi (dato che il secondo segmento è andato perso per ipotesi) e dovrà quindi attendere che scatti il timeout per poter inviare di nuovo il secondo segmento. Non appena riceverà il riscontro cumulativo dei primi quattro segmenti, l'host di A potrà inviare l'ultimo segmento.

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



b) Nel caso in cui la dimensione della finestra di congestione sia inizialmente di 5 MSS, l'host di A potrà inviare subito tutti e 5 i segmenti. Dato che (per ipotesi) solo il secondo segmento andrà perso, l'host di A riceverà per quattro volte lo stesso segmento di riscontro. Non appena riceverà il terzo riscontro duplicato, procederà alla ritrasmissione veloce del segmento perduto.



(A. Brogi)