



RETI DI CALCOLATORI E INTERNET (9 CFU)

A.A. 2021/2022

Docente: Prof. Alessandra De Paola

(Durata 2:30h)

Parametri della prova

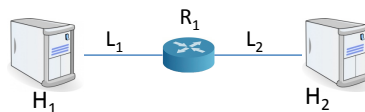
Sia $[X_0 \ X_1 \ X_2 \ X_3 \ X_4 \ X_5 \ X_6]$ il numero di matricola del candidato.

Si calcolino i seguenti parametri:

| | |
|--|-----------------|
| $Y_1 = \lfloor \frac{2 \cdot X_6}{5} \rfloor + 50$ | $Y_4 = Y_2 + 5$ |
| $Y_2 = \lfloor \frac{X_5}{5} \rfloor + 2$ | $Y_5 = Y_2 + 2$ |
| $Y_3 = \lfloor \frac{X_4}{5} \rfloor + 7$ | |

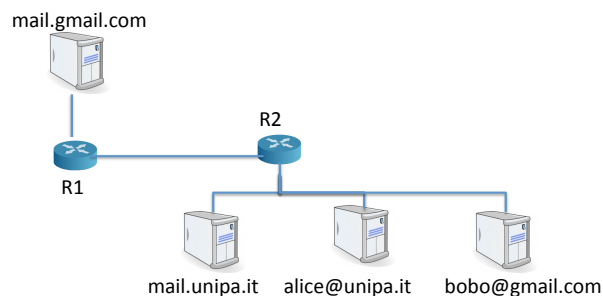
Quesito 1

Dati due host collegati come in figura, si supponga che l'host H_1 invii all'host H_2 un file di dimensioni Y_4 KB tramite il protocollo *GoBackN* con valore della finestra di trasmissione pari a Y_5 MSS, MSS di 512 byte ed overhead trascurabile. Si assuma che i link abbiano entrambi ampiezza di banda pari a $R_1 = 100$ Mbps e tempo di propagazione pari a 0.2 ms. Si determini il tempo necessario al completamento della trasmissione, nel caso in cui si perda l'ack del segmento Y_5 -esimo e che il valore di timeout sia fissato in modo da scadere 2.3 ms dopo il termine della trasmissione del segmento a cui è assegnato. Si calcolino infine il throughput e il goodput della trasmissione.



Quesito 2

Data la configurazione illustrata in figura, si supponga che l'utente Bobo debba inviare una email all'utente Alice, contenente un testo di $Y1$ KB e un $Y2$ allegati ciascuno di $Y3$ MB. Si assuma che tutti i collegamenti siano caratterizzati da ampiezza di banda pari a 100 Mbps, tempo medio di propagazione pari a 0.1 ms e MTU pari a 1500 Byte, ed si ipotizzi un overhead complessivo di pacchetto pari a 40 Byte. Si descrivano i dettagli delle comunicazioni necessarie a recapitare l'email al destinatario e si calcoli il tempo complessivo dall'invio della email alla sua ricezione.



Quesito 3

Si descriva il funzionamento dell'algoritmo di routing Distance Vector, anche attraverso un esempio illustrativo, evidenziando vantaggi e svantaggi. Si descrivano inoltre possibili soluzioni volte a risolvere eventuali problematiche.

Quesito 4

Completare il codice fornito per realizzare il server di una semplice applicazione di *random chat* tramite protocollo TCP. L'applicazione prevede che il server attenda la connessione di due client qualsiasi per poi metterli in comunicazione. All'interno di una coppia di client, ogni messaggio digitato dall'uno viene recapitato all'altro e visualizzato sullo standard output, secondo uno schema rigido client1-client2-client1-client2- Se uno dei due client digita il comando QUIT la chat viene interrotta.



```
#include <stdio.h>
#include <stdlib.h>      /* exit() */
#include <strings.h>     /* bzero(), bcopy() */
#include <unistd.h>
#include <netinet/in.h>
#include <sys/types.h>   /* tipi di dati di sistema */
#include <sys/socket.h> /* definizioni utili per le socket() */

#define MAXLINE 4096

void error(char *msg) {
    perror(msg);
    exit(1);
}

int main(int argc, char *argv[]) {

    int sockfd, newsockfd1, newsockfd2;
    int server_port = 6543, clilen;
    char buffer[256];
    struct sockaddr_in servaddr, cliaddr1, cliaddr2;
    pid_t pid;

    .....

    .....

    .....

    bzero((char *) &serv_addr, sizeof(serv_addr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
    servaddr.sin_port = htons(server_port);

    .....

    .....

    .....
```

Note:

Per le dimensioni relative ai file si considerino le grandezze come potenze di 2 e quindi in particolare:

1 MB = 1.024 kB 1 kB = 1.024 byte

Per le dimensioni relative ai tassi di trasmissione e alle ampiezze di banda si considerino le grandezze come potenze di 10 e quindi in particolare:

1 kbps = 1.000 bps 1 Mbps = 1.000.000 bps