



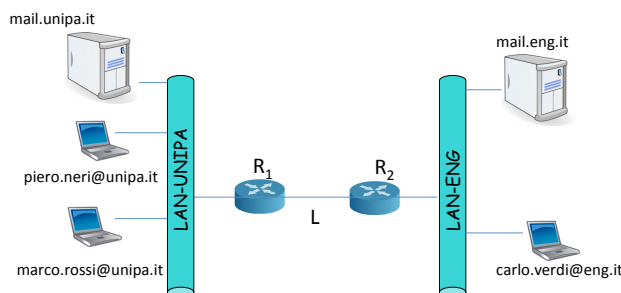
CORSO DI RETI DI CALCOLATORI E INTERNET A.A. 2013/2014

Docente: Ing. Alessandra De Paola

03 Luglio 2014

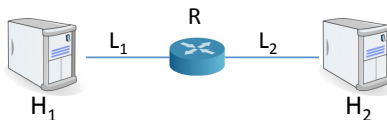
Quesito 1

Data la configurazione schematizzata in figura, si assuma che l'utente **piero.neri@unipa.it** invii una email all'indirizzo **carlo.verdi@eng.it** contenente un testo di 5KB e un allegato di 110KB, e che contemporaneamente l'utente **marco.rossi@unipa.it** invii allo stesso indirizzo di destinazione una email contenente un testo di 15KB e un allegato di 100KB. Si calcoli il tempo necessario alla trasmissione delle due email fino alla ricezione, assumendo che non vi siano tempi morti tra la fase di invio e quella di ricezione e che il destinatario utilizzi il protocollo POP3. Si assuma che le due LAN siano caratterizzate da ampiezza di banda $R_{LAN} = 100$ Mbps, mentre il link L sia caratterizzato da ampiezza di banda $R_L = 1$ Gbps. Per tutti i collegamenti si assuma tempo medio di propagazione pari a 0.5ms, MTU = 1500 Byte e overhead complessivo di pacchetto di 40 Byte.



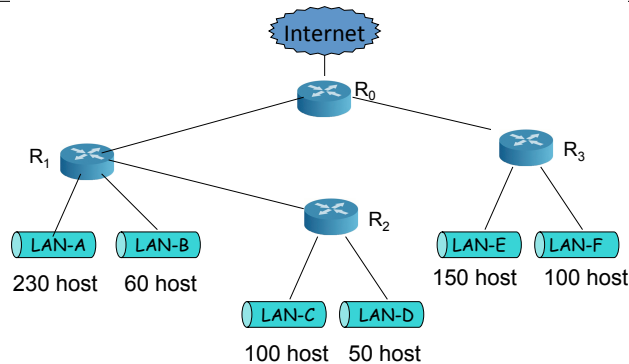
Quesito 2

Dati due host connessi come indicato in figura, si supponga che il primo host inizi a spedire un file di 10 kB verso l'altro mediante il protocollo TCP, con valore iniziale della soglia sulla finestra di congestione pari a 8MSS. Si assuma che i canali abbiano ampiezze di banda rispettivamente pari a 200 Mbps e 100 Mbps, che $RTT=0.4$ ms, $MSS=512$ byte e che gli overhead siano trascurabili. Si determini il throughput della trasmissione, assumendo che non si verifichi congestione. Si evidenzii inoltre l'andamento della finestra di congestione.



Quesito 3

Un amministratore di rete ha a disposizione il range 192.168.144.0/20 per indirizzare la rete indicata in figura. Si proponga uno schema di indirizzamento che minimizzi lo spreco di indirizzi per ciascuna rete e consenta di massimizzare il numero di ulteriori sottoreti da 50 host che è possibile aggiungere in futuro direttamente sotto il router R_2 . E' richiesto che la tabella di inoltro di R_1 non venga modificata e che sia minimizzata rispetto al numero di entry. Si scriva la tabella di inoltro per il router R_1 e si indichi il numero totale di reti aggiuntive.



Quesito 4

Completare il codice fornito per realizzare il server di un'applicazione Fibonacci tramite protocollo TCP. L'applicazione prevede che il client legga un numero N dallo standard input e lo invii al server; il server legge il valore dalla socket calcola la sequenza dei primi N numeri di Fibonacci e la rimanda al client; il client legge la risposta ottenuta e la manda sullo standard output.

```

#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, newsockfd;
    int server_port = 6543;
    struct sockaddr_in servaddr, cliaddr;

    if ( (sockfd = socket(AF_INET, SOCK_STREAM, 0)) < 0 )
        err_sys("errore in socket");

    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);

    if (bind(sockfd, (struct sockaddr *) &serv_addr, sizeof(serv_addr)) < 0) {
        error("ERROR on binding");
    }

    listen(sockfd, 5);

    send-fibonacci(sockfd, (struct sockaddr *) &cliaddr, sizeof(cliaddr));
}

```



```
void send-fibonacci(int sockfd, sockaddr *cliaddr, socklen_t clien)
{
    long n;
    long fibonacci_n;
    socklen_t len;
    char [MAXLINE];
    char* stringPtr;
    while(1) {
        len = clien;

        if ((newsockfd = accept(sockfd,
                                (struct sockaddr *) &cli_addr,
                                (socklen_t *)&len)) < 0) {
            error("ERROR on accept");
        }

        bzero(buffer,256);

        .....

        .....

        .....

        n = strtol(buffer, &stringPtr);

        fibonacci_n = fibonacci(n);

        .....

        .....

        .....

        .....

    }
}
```

Documentazione Programmazione Socket

//Accept an incoming connection on a listening socket

```
int accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen);
```

//Associate a socket with an IP address and port number

```
int bind(int sockfd, struct sockaddr *my_addr, socklen_t addrlen);
```

//Connect - initiate a connection on a socket

```
int connect(int sockfd, const struct sockaddr *addr, socklen_t addrlen);
```

```
int close(int sockfd); //Close a socket descriptor
```

```
struct hostent *gethostbyname(const char *name); //Get an IP address for a hostname
```

//Convert multi-byte integer types from host byte order to network byte order

```
uint32_t htonl(uint32_t hostlong);
```

```
uint16_t htons(uint16_t hostshort);
```

```
uint32_t ntohl(uint32_t netlong);
```



```
uint16_t ntohs(uint16_t netshort);

// Convert IP addresses to human-readable form and back
const char *inet_ntop(int af, const void *src, char *dst, socklen_t size);
int inet_pton(int af, const char *src, void *dst);

int listen(int sockfd, int queuelength); // Tell a socket to listen for incoming connections

// Receive data on a socket
ssize_t recv(int sockfd, void *buf, size_t len, int flags); // for TCP socket
ssize_t recvfrom(int sockfd, void *buf, size_t len, int flags, struct sockaddr *from, socklen_t *fromlen); // for UDP
socket
ssize_t read(int sockfd, void *buf, size_t count); // for TCP socket

// Send data out over a socket
ssize_t send(int sockfd, const void *buf, size_t len, int flags); // for TCP socket
ssize_t sendto(int sockfd, const void *buf, size_t len, int flags, const struct sockaddr *to, socklen_t tolen); // for UDP
socket
ssize_t write(int sockfd, const void *buf, size_t count); // for TCP socket

int socket(int domain, int type, int protocol); // Allocate a socket descriptor

// Structures for handling internet addresses
struct sockaddr_in { // ...
    short sin_family; // e.g. AF_INET, AF_INET6
    unsigned short sin_port; // e.g. htons(3490)
    struct in_addr sin_addr; // see struct in_addr };

struct in_addr { unsigned long s_addr; // e.g. INADDR_ANY };

// Structure for handling host names
struct hostent { // ...
    char *h_name; // The real canonical host name.
    int h_addrtype; // The result's address type, e.g. AF_INET
    int length; // The length of the addresses in bytes, which is 4 for IP (version 4) addresses.
    h_addr; // An IP address for this host. };

// The bzero() function sets the first n bytes of the area starting at s to zero
void bzero(void *s, size_t n);

// The bcopy() function copies n bytes from src to dest.
void bcopy(const void *src, void *dest, size_t n);
```

Nota:

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

Regolamento di esame La prova scritta, della durata di 2:30 ore, riguarda i contenuti coperti durante l'intero corso.



La consegna del compito equivale all'inizio dell'esame, il cui esito finale dipenderà dalla valutazione della prova scritta e di un esame orale da sostenere successivamente. È consentito agli studenti di non consegnare il compito scritto, nel qual caso la prova non verrà conteggiata nel numero massimo di tre prove d'esame che è possibile sostenere in uno stesso Anno Accademico.

Durante lo svolgimento della prova valgono le regole riportate di seguito:

- non è assolutamente consentito collaborare;
- non è consentito portare libri, fotocopie, appunti;
- è consentito l'uso di una calcolatrice;
- non è assolutamente consentito tener acceso il telefonino.

Nel caso in cui una delle sopra elencate regole per lo svolgimento degli esami non venga rispettata, si procederà con il ritiro del compito e con il conseguente annullamento della prova.

NB: nella valutazione dell'elaborato si terrà pesantemente conto della chiarezza espositiva.