

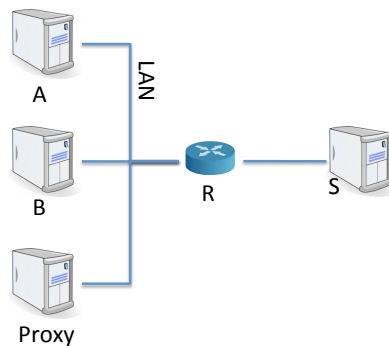
## CORSO DI RETI DI CALCOLATORI E INTERNET A.A. 2013/2014

*Docente: Ing. Alessandra De Paola*

### Quesito 1

Si consideri la configurazione schematizzata in figura, caratterizzata dalla presenza di un proxy server interno alla LAN e si assuma che il server *S* ospiti una pagina web di 10 KB contenente un'immagine di 14 MB.

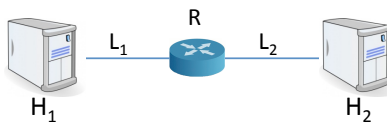
Si assuma che l'host *A* richieda la pagina web e che la stessa richiesta venga effettuata dall'host *B* nello stesso istante in cui l'host *A* termina la ricezione, e che in tutte le interazioni venga usato il protocollo HTTP/1.0. Si assuma che la LAN sia caratterizzata da ampiezza di banda  $R_{LAN} = 100$  Mbps e tempo di propagazione medio pari a 0.5 ms, che il collegamento *L* sia caratterizzato da ampiezza di banda  $R_L = 10$  Mbps e tempo di propagazione pari a 2 ms, e che si usi un  $MSS=1024$  Byte e overhead trascurabile. Si calcoli il tempo necessario a completare tutte le interazioni, e si discuta qualitativamente e quantitativamente la differenza con una configurazione priva del Proxy Server.



### Quesito 2

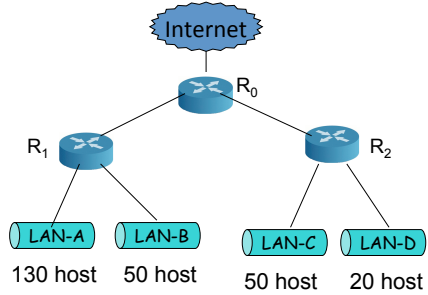
Dati due host collegati come in figura, si supponga che l'host  $H_1$  invii all'host  $H_2$  un file di dimensioni 4KB tramite un protocollo di tipo *Go-Back-N*, con finestra di spedizione statica  $W = 6$  e timeout = 30 ms. Si assuma che il RTT tra i due host sia pari a 1 ms, che le ampiezze di banda per i due link siano rispettivamente  $R_1 = 100$  Mbps e  $R_2 = 50$  Mbps, e che si abbia un  $MSS=512$  Byte e un overhead di pacchetto pari a 40 Byte. Si assuma inoltre che il router *R* abbia un buffer di dimensione 2 MTU.

Assumendo che il router rimuova dal buffer un pacchetto all'inizio della sua trasmissione in uscita, si determini se nel router si verifica congestione, determinandone gli effetti sulla trasmissione. Si determini il tempo totale necessario alla trasmissione del file e il *throughput* medio.



### Quesito 3

Avendo a disposizione il range di indirizzi 147.163.22.0/23 si proponga uno schema di indirizzamento per la configurazione indicata in figura che minimizzi lo spreco di indirizzi per ciascuna sottorete e che risulti coerente con la tabella di inoltro del router  $R_0$  indicata di seguito.



Prefisso	Interfaccia
147.163.22.0/23	Iface-R <sub>1</sub>
147.163.22.128/25	Iface-R <sub>2</sub>
0.0.0.0/0	Iface-Internet

## Quesito 4

Data l'applicazione server realizzata tramite il codice sorgente fornito:

- Qual è il comportamento desiderato dell'applicazione?
- Che tipo di socket viene utilizzata?
- Individuare e descrivere eventuali errori nel codice.
- Quali passi dovrebbe eseguire un client per potere interagire con questo server?



```
int main(int argc, char *argv[]) {
    int sockfd, newsockfd;
    int portno=5432, clilen;
    char buffer[256];
    struct sockaddr_in serv_addr, cli_addr;
    int pid_t pid;

    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0) {
        error("ERROR opening socket");
    }

    while(1){
        clilen = sizeof(cli_addr);
        newsockfd = accept(sockfd,
            (struct sockaddr *) &cli_addr,
            (socklen_t *)&clilen);

        if (newsockfd < 0) {
            error("ERROR on accept");
        }

        pid=fork();

        if (pid == 0) {
            close(sockfd);
            n = send(newsockfd,"Hello, world!", 13,0);
            if (n < 0) {
                error("ERROR sending");
            }
            close(newsockfd);
            return 0;
        }

        close(newsockfd);
    }
    return 0;
}
```



## 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

ssize_t recv(int sockfd, void *buf, size_t len, int flags); // Receive data on a socket

ssize_t send(int sockfd, const void *buf, size_t len, int flags); // Send data out over a 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);
```



**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

**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.**