



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

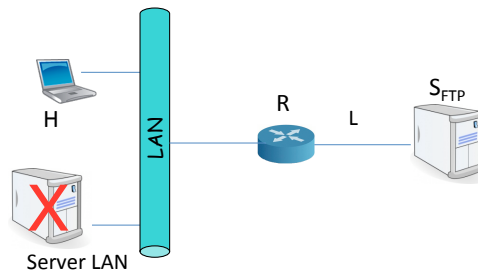
Docente: Ing. Alessandra De Paola

14 Gennaio 2015

Quesito 1

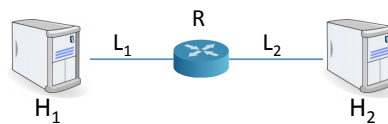
Data la configurazione illustrata in figura, si supponga che l'host H debba richiedere un file di 4 MB al server S_{FTP} tramite protocollo FTP. Si descrivano i dettagli della comunicazione e si calcoli il tempo complessivo necessario alla trasmissione.

Si assuma che la LAN e il link L siano caratterizzati rispettivamente da ampiezze di banda pari a $R_{LAN} = 200Mbps$ e $R_L = 150Mbps$ e che il tempo di propagazione medio sia pari a 1ms. Si assuma inoltre una MTU pari a 1500 byte ed un overhead complessivo di pacchetto pari a 46 Byte.



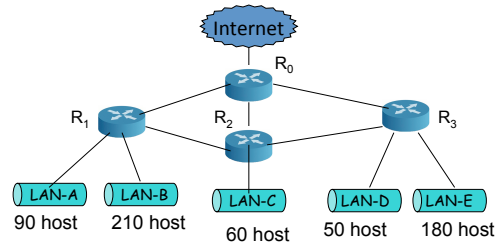
Quesito 2

Dati due host collegati come in figura, si supponga che il primo debba trasferire al secondo un file di 3kB, usando il protocollo GoBackN. Si assuma un RTT tra i due host pari a 0.3 ms ed ampiezze di banda per i due canali rispettivamente pari a $R_1 = 150 Mbps$ e $R_2 = 100 Mbps$, $MSS = 256$ byte, overhead di pacchetto trascurabili e timeout pari ad 1.5 ms. Ipotizzando che si perda il sesto segmento, e che la dimensione statica della finestra di spedizione sia pari a 5MSS, si calcolino il throughput e il goodput della trasmissione.



Quesito 3

Avendo a disposizione il range di indirizzi 192.168.0.0/22 si proponga uno schema di indirizzamento per la configurazione indicata in figura che minimizzi lo spreco di indirizzi per ciascuna sottorete e che ottimizzi le tabelle di inoltro, in base ad un albero di routing opportunamente scelto. Si motivino le scelte effettuate.



Quesito 4 – A.A. 2013/2014

Scrivere il codice *core* di un client in grado di collegarsi ad un server TCP all'indirizzo e porta specificati come primo e secondo parametro sulla riga di comando e che invii al server due numeri interi senza segno letti dallo standard input. Il client deve ricevere come risposta la somma dei due interi e inviare tale valore allo standard output.

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 se lo studente ha seguito il corso nell'A.A. 2013/2014, di 2:00 ore altrimenti, e 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.