Risposte Domande Reti 1

1) *Definire il concetto di protocollo e poi spiegare come funziona un protocollo particolare a scelta. Perché gli standard sono importanti nei protocolli?*
   Un protocollo è un'insieme di regole che definisce il formato e l'ordine dei messaggi e il tipo di comunicazioni che gli host devono mantenere per essere compresi dagli altri, un esempio è HTTP. Questo protocollo permette la visualizzazione di pagine web nei browser, la comunicazione tra gli host avviene tramite TCP e può essere persistente, ovvero che viene utilizzata una sola connessione per richiedere la pagina web e i relativi oggetti, e non persistente, cioè vengono instaurate più connessioni per ogni oggetto. Gli standard nei protocolli sono importanti perchè essi permettono una semplificazione dell'implementazione di questi ultimi.

2)*Quale vantaggio presenta una rete a commutazione di circuito rispetto ad una a commutazione di pacchetto? Quali vantaggi ha TDM rispetto a FDM in una rete a commutazione di circuito?*
Le reti a commutazione di circuito hanno il vantaggio di avere una banda minima garantita rispetto alle reti a commutazione di pacchetto, dato che riservano parte delle risolse al circuito. TDM rispetto a FDM passa l'intera banda a ogni frame a cui è associata una connessione, in questo si ha uno sfruttamento più efficiente della banda.

3) Considerate l'invio di un pacchetto da un host ad un altro lungo un percorso fisso. Elencate le componenti di ritardo complessivo, indicando quali sono costanti e quali variabili.
Le componenti del ritardo end-to-end di un pacchetto sono:

- Ritardo di elaborazione: ritardo dovuto all'elaborazione da parte del router, che controlla l'indirizzo di destinazione per l'inoltro;
- Ritardo di accodamento: ritardo dovuto al traffico presente in quel commutatore e indica il tempo in cui il pacchetto rimane all'interno del buffer;
- Ritardo di propagazione: ritardo che indica il tempo in cui il primo bit del pacchetto ri propaga nel collegamento, non dipende dalla velocità di trasmissione o dalla dimensione del pacchetto;
- Ritardo di trasmissione: ritardo che indica il tempo in cui il pacchetto viene trasmesso da un commutatore a un altro, non dipende dalla lunghezza del collegamento.

Le costanti che riguardano i ritardi sono la lunghezza del buffer, il numero di router da attraversare e la velocità di trasferimento di un collegamento mentre quella variabile è l'intensità di traffico.

5)*Il tempo di propagazione dipende dalla lunghezza del pacchetto? Dipende dalla velocità di trasmissione?*
Il tempo di propagazione non dipende nè dalla lunghezza del pacchetto, nè dalla velocità di trasmissione, esso dipende dalla lunghezza del collegamento.

8)*Quali sono i cinque livelli della pila di protocolli Internet? Quali sono i loro principali compiti?*

- Applicazione: livello in cui risiedono le applicazioni di rete in esecuzione negli host, ogni applicazioni di rete è distribuita tra gli host è ogni processo comunica con l'altro attraverso appositi messaggi;
- Trasporto:livello che permette il trasferimento dei messaggi applicativi incapsulati in segmenti tra processi appartenenti a due host differenti;
- Rete: livello che permette l'instradamento dei pacchetti nel cuore della rete fino ad arrivare a destinazione attraverso l'ausilio di appositi algoritmi che calcolano il percorso ottimale in base determinate variabile quali il traffico;
- Collegamento: livello che si occupa dello spostamento dei frame da un elemento di rete a quello adiacente, i servizi forniti dipendono dal protocollo utilizzato.
- Fisico: livello il cui compito è trasferire i bit del frame da un nodo all'altro attraverso il collegamento dato, i suoi protocolli dipendono dal livello stesso e dal mezzo utilizzato.

9)*Spiegare i campi del datagram IP.*

- numero di versione: indica la versione del protocollo per una corretta interpretazione da parte del router;
- lunghezza intestazione: indica l'inizio effettivo dei dati, la maggior parte non contiene opzioni e quindi è di solito di 20 bit;
- tipo di servizio: indica il servizio utilizzato;
- lunghezza del pacchetto: lunghezza del pacchetto (intestazione compresa) in byte;
- Id, flag e offset: campi che servono per la frammentazione;
- TTL: indica il numero di hop percorribili da quel pacchetto, viene scartato quando arriva a 0;
- Protocollo: indica il protocollo di trasporto utilizzato;
- Checksum: campo che serve per il controllo degli errori, viene ricalcolato in ogni router;
- Opzioni: estendono l'intestazione;
- Dati: segmento o altri tipi di dati;

10) *Come è composto l'header di un pacchetto TCP e che compito svolgono i campi?*

- porta di origine e destinazione;
- checksum: utilizzato per il controllo degli errori;
- sequence number: numero del primo byte del flusso;
- ACK number: indica il sequence number successivo da attendere;
- finestra di ricezione:utilizzata per il controllo di flusso;
- lunghezza: indica l'inizio della parte dati;
- opzioni: utilizzati per negaziare MSS;
- flag: 6 bit aventi un ruolo specifico, ACK indica che il segmento è un ACK, RST, SYN e FIN sono utilizzati per impostare e chiudere la connessione, CWR e ECE per il controllo della congestione PSH indica che i dati deve essere inviati immediatamente all'applicazione e URG indica dati marcati come urgenti;

11) *Discutere, fare esempi e confrontare le architetture client-server e P2P.*

Nell'architettura client-server la comunicazione avviene tra due host principali: il server è un host sempre acceso il cui compito è fornire servizi utilizzati dai client, questi ultimi non

comunicano direttamente tra loro. In questa architettura il server deve inviare una copia di un file grande F a tutti gli N client, quindi se il server ha una banda s e il client i ha una banda p(i) differente da quella degli altri, allora il tempo richiesto affinchè tutti abbiano il file dipende sia dal tempo che ci mette il server a inviarlo a tutti (N*F/s) e il tempo del client "più lento" (F/p(i)), quindi:
Tcs= max{NF/s , F/p}
Com'è possibile notare dalla formula, il tempo in questa architettura è lineare nel numero di client presenti nella rete. Nella rete P2P, invece, ogni host può essere sia client che server in quanto ognuno può richiedere parti di un determinato file e distribuire agli altri quelle a sua disposizione. Inizialmente il file grande F è sul server, esso lo trasmette in ogni collegamento in un tempo F/s in cui s indica la banda del server. Il client più lento ha banda p e riceve il file per ultimo, quindi in un tempo F/p tutti i client hanno ricevuto il file direttamente dal server. La velocità di upload totale equivale alla somma di tutte le velocità, quindi il sistema deve consegna in essa N*F bit, il tempo totale del P2P è:
Tp2p= min{ F/s , F/p , N*F/(s+somma(p(i)))
Questo tempo è logaritmico rispetto al numero di host ed è quindi minore rispetto al client-server, tuttavia quest'ultima garantisce maggior sicurezza e affidabilità.

12) *Se una transizione tra client e server deve essere la più veloce possibile, cosa è meglio tra TCP e UDP (motivare la risposta)?*
Per una transizione più veloce possibile è meglio utilizzare UDP perchè non vi sono controlli o rallentamenti causati dal flusso/congestione come in TCP e non bisogna effettuare handshaking, tuttavia l'affidabilità ne risente.

13) L'handshaking è una metodo che serve per sincronizzare due host in cui si stabiliscono velocità, crittografia, controllo degli errori e altre informazioni. Esso viene utilizzato in TCP per instaurare una connessione tra due host e quindi garantire il trasferimento affidabile dei dati.

14) *Si supponga che Alice debba mandare un messaggio di posta elettronica a Bob e che quest'ultimo scarichi la posta con il POP3. Descrivere il "viaggio" di questo messaggio dettagliando i protocolli coinvolti.*
Attraverso la sua casella di posta, Alice scrive il messaggio, inserisce l'indirizzo e-mail di Bob e dice alla propria casella di inviarlo, esso utilizza SMPT (Simple Mail Transfer Protocol) per trasmetterlo DIRETTAMENTE alla casella di Bob senza passare per caselle intermedie anche se sono molto distanti tra loro. Arrivato nella casella di Bob, il messaggio viene scaricato utilizzando il protocollo POP3 (Post Office Protocol 3) nel dispositivo e da qui può essere possono essere fatte eventuali operazioni che verranno in seguito inviate alla casella.

15) *Che differenza c'e' tra POP3 e IMAP?*
POP3 e IMAP svolgono la stessa funzione, la differenza è che il primo protocollo scarica la posta dalla casella al dispositivo mentre il secondo esegue una sincronizzazione tra i 2, quest'ultimo permette anche la possibilità di organizzare in cartelle i messaggi nella casella stessa, cosa che in POP3 è possibile solo localmente. IMAP inoltre permette anche di ottenere singole parti di un messaggio, utile quando la connessione è limitata.

16)*Descrivere la sequenza di system call sia per il client, sia per il server, per il protocollo TCP.*

- Socket: crea un socket description indicando il tipo di indirizzo, la porta e il protocollo utilizzato;
- Bind(solo server): vincola un socket a una data porta per l'interfaccia di rete;
- Listen(solo server): si attende che un client si connetta al server;
- Accept(solo server): alla connessione di un client a un server, la connessione viene accettata e quindi si può passare all'esecuzione dell'applicazione vera e propria;
- Connect(solo client): prova ad aprire una connessione col server specificato tramite l'indirizzo nei parametri;
- Read: permette la lettura di un socket descriptor;
- Write: permette la scrittura in un socket descriptor;
- Close: chiude la connessione.

17)*Descrivere la sequenza di system call sia per il client, sia per il server, per il protocollo UDP.*

- Socket: crea un socket descriptor utilizzando i parametri dati;
- Bind: lega il socket all'indirizzo del server;
- sendto: invia un pacchetto all'indirizzo specificato;
- recvfrom: rimane in attesa di un pacchetto da parte dell'indirizzo specificato;
- close: chiude il socket descriptor;

18)*Spiegare come funziona la select per il multiplexing*

select permette il multiplexing tra socket che sono in attesa di ricezione, quando bisogna inviare dati a essi senza attendere oppure in caso di eccezione, essa funziona attraverso delle macro, come parametri prende il numero di socket, tra puntatori che indicano i tre casi descritti prima e un puntatore a una struttura timeval che indica il tempo d'utilizzo di un socket prima di passare al successivo. Come già detto in precedenza, select utilizza delle macro:

- FD_SET: aggiunge un descriptor al set;
- FD_CLR: rimuove un descriptor dal set;
- FD_ISSET: restituisce vero se il descriptor fa parte del set, falso altrimenti;
- FD_ZERO:rimuove tutti i descriptor dal set.

19)*Spiegare i parametri e come funzionano le system call read/write*

le system call read e write hanno gli stessi parametri, essi indicano:

- il descriptor su cui leggere/scrivere;
- un puntatore che indica cosa scrivere nel descriptor oppure dove inserire la porzione letta da esso;
- il numero di byte da leggere/scrivere;

20) *Discutere vantaggi e svantaggi dell'approccio concorrente (fork) rispetto a quello multiplexing (select)*

- Nel multiplexing non vi è concorrenza, nel fork c'è;

- Di conseguenza, nel primo caso ci sono lunghi tempi di attesa ma poco tempo di esecuzione, nel secondo è il contrario;
- Nel multiplexing l'ordine dei job è prevedibile, nel fork non è così;
- I dati non vengono copiati e non vi sono errori relativi a processi nel multiplexing, al contrario nella fork queste cose sono presenti.

21) *Quali tecniche si possono usare per rendere una applicazione di rete tollerante ai guasti?*

22) *Come funziona l'applicativo traceroute? Che valori restituisce?*
Il traceroute restituisce i router appartenenti a un percorso verso una data destinazione, misurando i tempi di risposta di ognuno 3 volte.

23)*Commentare il seguente codice:*

```
1: rset=allset;

2: if(ready=select(maxd+1,&rset,NULL,NULL,NULL)<0) perror("select problem");
3:                  if(FD_ISSET(sockfd,&rset))          {
4:                                          sin_size = sizeof(their_addr);
5:                                          new_fd = accept(sockfd, (struct sockaddr *)&their_addr,
&sin_size);
6:                                          if(write(new_fd,"Riprovare più tardi\n",strlen("Riprovare più
tardi\n")) == -1) perror("write");
7:                                  }
```

24) *Supponete che un server web sia in esecuzione sull'host C sulla porta 80. Supponete che questo web server usi le connessioni persistenti stia al momento ricevendo richieste da diversi host, A e B. Tutte le richieste vengono inviate attraverso la stessa socket? Se vengono fatte passare attraverso socket diverse, entrambe hanno la porta 80? Discutete e spiegate questa soluzione.*
Tutte le richieste possono essere inviate anche utilizzando numeri di porta differenti se gli host A e B si trovano nella stessa rete, in caso contrario possono anche avere lo stesso numero perché sono anche identificate dall'indirizzo. Tutte le richieste verso l'host C avranno la porta 80 come porta di destinazione dato che essa è quella specifica del protocollo utilizzato, ovvero HTTP.

25)*Perchè in questi protocolli abbiamo introdotto*

- *Il numero di sequenza?*
- *Il timer?*

Motivare le risposte.
Il sequence number nei pacchetti è stato aggiunto per evitare i casi in cui ACK/NAK vengano persi o corrotti, infatti se arriva ACK/NAK, esso si riferisce al pacchetto trasmesso più di recente. Il destinatario deve controllare il sequence number per informare il mittente di

eventuali ritrasmissioni o meno, infatti se riceve un pacchetto con sequence number diverso da quello richiesto, invia un ACK ma non passa subito i dati all'applicazione, lo fa solo quando vi è la sequenza corretta. Se invece il destinatario riceve il pacchetto con sequence number corretto, invia un'ACK e passa i dati all'applicazione. Il timer è stato aggiunto per risolvere i problemi riguardo le perdite dei pacchetti, infatti un pacchetto è considerato perso quando lo è per davvero oppure quando non arriva un ACK in tempo, in poche parole quando vi è un timeout.

26) *Supponendo che il ricevente UDP calcoli il checksum per il segmento UDP ricevuto e trovi che corrisponda al valore trasportato nel campo checksum. Può il ricevente essere assolutamente certo che non vi siano stati errori sui bit?*
Motivare la risposta.

27) *Rtd 3.0: commentare la figura*
Il mittente è in attesa di una send per poter inviare i pacchetti al destinatario, quando ciò avviene, il pacchetto ha sequence number 0 e si mette in attesa dell'ACK 0, se nel frattempo riceve pacchetti dal destinatario, non deve fare nulla. Quando è in attesa dell'ACK 0, la ricezione di un ACK 1 e il timeout causano la ritrasmissione del pacchetto, se invece arriva il pacchetto 0, il mittente torna in attesa di invio ma il sequence number a 1. L'attesa di invio e quella dell'ACK 1 funzionano allo stesso modo dei due precedenti, l'unica differenza sta nelle condizioni: se da una parte c'è 1, dall'altra c'è 0 e viceversa.

28) *Come è possibile garantire la correttezza nell'invio dei dati, pur usando un protocollo UDP?*
Perchè alcune volte si opta per questa scelta, pur avendo TCP come protocollo affidabile?
Il trasferimento dati affidabile con UDP è possibile attraverso QUIC (Quick UDP Internet Connection), un protocollo che implementa il trasferimento dati affidabile a livello applicazione. Spesso si opta per questa scelta per evitare gli onerosi meccanismi di controllo di TCP.

29)*Il server DHCP serve per ottenere la maschera di rete, il router di default (gateway), un indirizzo IP e quello del DNS. Spiegare il formato e significato di ogni valore.*
Il gateway indica il router che permette l'entrata/uscita di pacchetti dalla sottorete, l'indirizzo IP è quello relativo all'host che l'ha richiesto e il DNS

30) *Come fa il livello di rete, quando riceve un pacchetto, a conoscere quale protocollo di livello di trasporto e' destinato il pacchetto stesso?*
Il livello di rete riesce a capire a quale protocollo di trasporto è destinato il pacchetto grazie al campo TOS (Type of service).

31) *Confrontare ed evidenziare similitudini e differenze nei campi di intestazione IPv4 e IPv6.*
A differenza dell'IPv4, in IPv6 non è presente il checksum (ridondante perchè presente anche in altri livelli), le opzioni non più più integrate nell'intestazione e non più possibile eseguire la frammentazione nei router intermedi. I campi simili invece sono la versione del

protocollo, il limite di hop (simile al TTL), la classe di traffico (simile a TOS) e l'IP sorgente e di destinazione, oltre che il campo dati.

32)*È necessario che ogni AS utilizzi lo stesso algoritmo di instradamento interno? Motivare la risposta.*

In ogni AS non è necessario che vi sia lo stesso algoritmo di instradamento dato che Un sistema OSPF può essere configurato il tale modo