Categoria 1:

DNS----------

1) Se non funziona momentaneamente il DNS, cosa succede al protocollo SMTP, ovvero come ne risente?

Se il DNS è momentaneamente fuori uso, per poter inviare la mail, è necessario che nella cache DNS sia disponibile la corrispondenza con l'IP del server di posta di cui so il nome.

2) Nell'ambito del DNS che differenza c'è tra query ricorsiva e query iterativa?

In una *query iterativa* , interrogo il mio DNS locale il quale mi invia la corrispondenza hostname/ip o, nel caso non ne sia in possesso, restituisce l'indirizzo del server da contattare per avere la risoluzione della richiesta.
In una *query ricorsiva*, il server contattato avrà la responsabilità di tradurre l'hostname richiesto. Nel caso in cui non abbia già questa corrispondenza, interroga i vari server fino a quando non avrà la risposta da restituire.

3) Spiegare cosa si intende per caching DNS e quali sono i possibili vantaggi- svantaggi

Per caching DNS si intende la memorizzazione delle corrispondenze IP-Hostname nella cache propria del server DNS. Ad esempio di server DNS locali possono mantenere in cache gli IP dei vari server TLD in modo da non interrogare spesso i server radice alleggerendo il traffico verso questi e il loro lavoro. Usare la cache DNS comporta la velocizzazione del processo e cala il numero di messaggi DNS che "girano" in rete. Alcuni possibili svantaggi sono 1) se la cache non è aggiornata si rischia di mandare i dati ad una corrispondenza sbagliata. 2) La cache DNS può essere modificata in modo da "dirottare" una corrispondenza 3) se la corrispondenza è nella cache DNS allora la ricerca si velocizza, ma se non è in cache, si è perso tempo nel ricercarla lì. Le corrispondenze in cache sono mantenute per un tempo generalmente di 2 giorni.

4) In un record DNS cosa indica type=MX?

Con type MX il *name* del record è il sinonimo del server di posta destinatario mentre *value* è il nome canonico di tale server, quindi dal sinonimo conosciuto mi faccio restituire, dopo la richiesta DNS, il nome canonico. Ad esempio: pinco@libero.it mi restituisce libero.it

5) Spiega come è fatto un record DNS di tipo A.

Col type A, nel record al campo *name* ho l'hostname del server mentre in *value* ho il relativo indirizzo IP. Serve quindi a risolvere la corrispondenza hostname-IP. Ad esempio si usa dopo CNAME ed MX per ottenere l'IP dopo aver trovato il nome canonico di un sinonimo.

6) Spiega cosa sono e come funzionano i DNS radice e di competenza.

I server DNS sono organizzati in una gerarchia che vede, in cima, il server DNS radice. Tale server DNS viene interrogato dall'host locale. Se io devo ottenere l'IP di *www.amazon.com*, il server radice mi restituisce l'insieme degli IP dei server TLD che si occupano dei domini *.com* . Tali TLD restituiscono il server che si occupa di *amazon.com* ovvero i server di competenza che sapranno restituire l'IP di *www.amazon.com*. I server radice sono 13 ma sono tutti lo stesso server replicato per motivi di sicurezza. Mentre i server di competenza includono i domini di ogni sito internet pubblicamente accessibile.

7) A cosa serve campo identificativo DNS? e spiega come funziona il protocollo.

Il campo identificativo è un campo di 16 bit che indica il numero di richiesta/risposta. Questo in modo da sapere associare ogni risposta alla propria richiesta. Il protocollo DNS serve per "recuperare" da ogni hostname il rispettivo IP e tale recupero avviene interrogando varie tipologie di server DNS (attraverso query ricorsive o iterative) e ricevendo come risposta tale IP.

8) Se un computer non può fare una query DNS , come fa a visualizzare una pagina di cui sa solo il nome?

Se un computer non può fare una query DNS, quindi non può richiedere la corrispondenza hostname/IP, deve cercare tale corrispondenza nella cache, rischiando però, nel caso in cui questa corrispondenza sia presente, che non sia aggiornata.

Categoria 2:

Socket----------

1) Spiegare dettagliatamente cos'è e come viene usata una datagram soket da un server. spiegare inoltre quali protocolli verranno sicuramente utilizzati.

Una datagram socket, altro nome di socket udp, è la socket che permette l'interazione client server. Il server infatti rimane in ascolto in attesa di un segmento udp da un client qualsiasi. Non abbiamo infatti alcuna instaurazione di connessione con uno specifico client. Ad ogni segmento, il server manda una risposta, deve quindi procurarsi ip e porta del mittente. Al contrario di tcp, qui

abbiamo un'unica socket, al quale arrivano le richieste dei vari client. Infatti se ho due o più client, il server elabora i segmenti in ordine di arrivo (ovviamente fin quando il buffer non è pieno).

2) Spiegare lo schema di funzionamento di un'applicazione server TCP e relative socket.

Un'app server tcp deve mantenere in ascolto una socket, la welcome socket, per accettare eventuali richieste entranti da parte del client. Non appena tale richiesta viene accettata, client e server effettuano un 3way handshake (che avviene completamente a liv di trasporto, quindi invisibile alle applicazioni client e server) ed instaurano una connessione tcp. Durante tale handshake il server crea una socket specifica (socket di connessione) per la richiesta del client. Ora può esser fatto l'invio dei dati.

3) Descrivere un'applicazione da livello server che si basa su UDP con particolare riferimento alla socket.

Il *server udp* rimane in ascolto in attesa di un segmento udp da un client qualsiasi. Non abbiamo infatti alcuna instaurazione di connessione con uno specifico client. Ad ogni segmento, il server manda una risposta, deve quindi procurarsi ip e porta del mittente. Al contrario di tcp, qui abbiamo un'unica socket, al quale arrivano le richieste dei vari client. Infatti se ho due o più client, il server elabora i segmenti in ordine di arrivo (ovviamente fin quando il buffer non è pieno).

4) Commentare riga per riga il codice delle socket (è stato fornito il codice nel compito).

UDP -> Socket Client Java

```
import java.io.*;  →  gestisce classi per input/output
import java.net.*; → gestisce classi per applicazioni di rete
class UDPClient {
public static void main(String args[]) throws Exception
{
BufferedReader inFromUser =
   new BufferedReader(new InputStreamReader(System.in)); → crea flusso d'ingresso prendendolo da tastiera

DatagramSocket clientSocket = new DatagramSocket();→ crea una datagram socket

InetAddress IPAddress = InetAddress.getByName("hostname"); →Mi procuro l'ip dall'hostname (uso dns)

byte[] sendData = new byte[1024]; -----| alloco due buffer per i dati da inviare e ricevere
byte[] receiveData = new byte[1024]; --| dimensione 1024 perché la max dimensione UDP è 2^16

String sentence = inFromUser.readLine();  → metto il contenuto della lettura in sentence

sendData = sentence.getBytes();→ devo estrarre dei byte dalla frase

DatagramPacket sendPacket = ----------------------------------------------------| Crea il datagramma con i dati da
  new DatagramPacket(sendData, sendData.length, IPAddress, 9876); ---|mandare (Ip, porta, lunghezza…)

clientSocket.send(sendPacket); → Spedisco il datagramma al server

DatagramPacket receivePacket = ----------------------------------| Datagramma dove metterò la risposta,
   new DatagramPacket(receiveData, receiveData.length); ---| ovvero l'array dove mettere il payload

clientSocket.receive(receivePacket); → legge il datagramma dal server (nel frattempo si addormenta)

String modifiedSentence = ----------------------| Stringa che prende in input la sequenza di byte presa dal receive
  new String(receivePacket.getData());-------| packet

System.out.println("FROM SERVER:" + modifiedSentence);  → Stampa a video
clientSocket.close();
}
}
```

UDP -> Socket Server Java

import java.io.*; → gestisce classi per input/output
import java.net.*; → gestisce classi per applicazioni di rete
class UDPServer {
public static void main(String args[]) throws Exception
{
DatagramSocket serverSocket = new DatagramSocket(9876); -> Creo una datagram socket

byte[] sendData = new byte[1024]; -----| alloco due buffer per i dati da inviare e ricevere
byte[] receiveData = new byte[1024]; --| dimensione 1024 perché la max dimensione UDP è 2^16

while(true)
{
DatagramPacket receivePacket = → alloco spazio per il datagramma. Ora ho tutto pronto
new DatagramPacket(receiveData, receiveData.length);

serverSocket.receive(receivePacket); → mi metto in ascolto finchè non arriva un datagramma
String sentence = new String(receivePacket.getData()); -> prendo il pacchetto arrivato, estraggo i dati e creo stringa

InetAddress IPAddress = receivePacket.getAddress(); ---| devo ricavare IP e porta mittente per poter preparare risp
int port = receivePacket.getPort(); ----------|

String capitalizedSentence = sentence.toUpperCase(); → la stringa ricavata dal payload la metto in maiuscolo

sendData = capitalizedSentence.getBytes(); -> trasformo la stringa in byte

DatagramPacket sendPacket =----------| preparo datagramma da inviare al client
new DatagramPacket(sendData, sendData.length, IPAddress, port);-----|

serverSocket.send(sendPacket); -> mando datagramma
} -> Fine del ciclo while, ricomincia il ciclo e attende un altro datagramma
}
}

TCP -> Socket Client Java

```
import java.io.*;  →  gestisce classi per input/output
import java.net.*; → gestisce classi per applicazioni di rete
class TCPClient {
public static void main(String argv[]) throws Exception
{
String sentence; -> stringa d'origine
String modifiedSentence; -> stringa modificata

BufferedReader inFromUser =  → leggo blocchi di caratteri grazie al buffer che fa legger stringhe da tastiera
   new BufferedReader(new InputStreamReader(System.in));

Socket clientSocket = new Socket("hostname", 6789); -> crea una socket client connessa al server dove servirà la
                                                                                    Risoluzione DNS per ricavare l'IP
dall'hostname
DataOutputStream outToServer =  -> chiedo alla socket uno stream in scrittura per scriver sulla socket e
contattare
  new DataOutputStream(clientSocket.getOutputStream());    il server

BufferedReader inFromServer =  -> "dammi un input stream da usare per leggere le risposte del server"
  new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

sentence = inFromUser.readLine(); -> metto il contenuto della lettura in sentence

outToServer.writeBytes(sentence + '\n');  -> prendo il flusso in uscita dalla socket e ci scrivo la stringa

modifiedSentence = inFromServer.readLine(); -> attendo la risposta dal server con la stringa modificata

System.out.println("FROM SERVER: " + modifiedSentence);

clientSocket.close();
}
}
```

TCP -> Socket Server Java

```
import java.io.*;  →  gestisce classi per input/output
import java.net.*; → gestisce classi per applicazioni di rete
class TCPServer {
public static void main(String argv[]) throws Exception
{

String clientSentence;
String capitalizedSentence;
ServerSocket welcomeSocket = new ServerSocket(6789);

while(true) {
Socket connectionSocket = welcomeSocket.accept();  -> attendo un contatto dal client, se arriva, faccio
handshake

BufferedReader inFromClient =  -> crea flusso d'ingresso collegato alla socket
    new BufferedReader(new InputStreamReader(connectionSocket.getInputStream()));

DataOutputStream outToClient =  ->cerco di ottenere un output stream
    new DataOutputStream(connectionSocket.getOutputStream());

clientSentence = inFromClient.readLine();   -> legge da tastiera e mette in sentence

capitalizedSentence = clientSentence.toUpperCase() + '\n';

outToClient.writeBytes(capitalizedSentence);
} -> Fine del ciclo while, ricomincia il ciclo e attende un'altra connessione con il client
}
}
```


Categoria 3:

IP----------

1) Come fa un host che non ha un indirizzo IP a riceverne uno? Qual è il protocollo usato?

A questo scopo utilizziamo il protocollo DHCP che consente ad un host di ricevere un IP dinamico (che cambia ogni volta che si collega alla rete) e in modo automatico, senza che l'utente debba configurare nulla. Quando il client, ovvero il nostro host, si collega alla rete, la prima cosa da fare è quella di trovare l'indirizzo IP del server DHCP con il quale dovrà interagire. Manda quindi a questo un messaggio di indentificazione DHCP in un pacchetto UDP contenente come IP di origine 0.0.0.0 e di dest 255.255.255.255, ovvero mandando il messaggio in broadcast. Il server DHCP, dopo aver ricevuto tale messaggio, ne manda uno di offerta al client, con l'IP proposto, la maschera di rete e la durata di concessione dell'IP. Il client quindi si trova a dover scegliere tra le varie offerte. Una volta presa la decisione manda la richiesta DHCP con tutti i parametri per la configurazione. Alla fine, il server DHCP manderà un ACK DHCP per confermare il tutto e finalmente il client avrà ricevuto il suo IP.

2) come sono fatti i datagrammi IPv6 e quali sono le differenze rispetto a quelli IPv4 ?

3) Descrivere struttura di ipv4 e ipv6

- Un datagramma IPv4 si compone più campi rispetto un datagramma IPv6, nonostante abbia intestazione di 20 byte (che può ampliarsi in presenza di opzioni aggiuntive) rispetto ai 40 byte fissi di IPv6. Entrambi hanno un campo versione IP e uno che indica, se a livello trasporto, il datagramma passa in mano di TCP e UDP. Un campo simile a entrambi è quello che indica il tipo di servizio di rete/classe di traffico. Entrambi i datagrammi hanno poi il campo –lunghezza di intestazione/carico utile- che permette di capire dove inizia il contenuto vero e proprio del datagramma. Ovviamente entrambe le versioni di IP hanno il campo di indirizzo di origine e destinazione e quello del TTL. Invece, i campi dedicati alla frammentazione, -flag, identificatore, spostamento laterale di frammentazione-, non sono presenti in IPv6 perché non ammette frammentazione. In caso di datagrammi troppo lunghi, si limita a mandare un messaggio ICMP per notificare tale fatto. In questo modo si toglie carico di lavoro ai router, lasciando la frammentazione ai sistemi terminali. Anche il campo checksum non è presente in IPv6 in quanto considerato ridondante siccome il controllo degli errori è presente anche a livello collegamento e trasporto.

4) A cosa serve il campo TTL nel pacchetto ip?

Serve a dare un tempo di vita al datagramma IP. Infatti questo campo si decrementa ad ogni salto (hop), ovvero ad ogni router incontrato, fino ad arrivare a zero. In questo modo si evita che il datagramma resti in rete troppo tempo, evitando così di congestionare la rete e produrre traffico.

5) Illustra il tunneling in ipv6

Il metodo del tunneling consente "il passaggio" da IPv4 a IPv6, ovviando al problema dato dal metodo del nodo a doppia fila, ovvero, convertire un datagramma da IPv6 a IPv4 porterebbe a non avere un riscontro per tutti i campi del datagramma IPv6 in quanto dal 4 al 6 alcuni campi spariscono e/o si modificano. Il tunneling infatti consiste nell'inserire all'interno di un datagramma IPv4 un datagramma completo versione 6 così da poterlo inoltrare per nodi che ancora non supportano la nuova tipologia di indirizzi. Raggiunto un nodo versione 6, leggerà nel datagramma della presenza dell'IPv6 e lo inoltrerà come se lo avesse ricevuto da un router già IPv6

6) Intestazione ip: contiene i numeri di porta?

Non li contiene, in quanto non sono campi di interesse del livello di rete che si occupa della comunicazione tra host mittente e ricevemente. I numeri di porta servono infatti per le socket che identificano un processo, quindi tali numeri interessano solo al livello di trasporto e non a quello di rete.

8) Spiega come avviene la frammentazione IP e fai un esempio

La frammentazione IP è necessaria quando il datagramma supera l'MTU, ovvero l'"unità massima di trasmissione" che indica il massimo di dati trasportabili da un frame. quindi per trasportare un datagramma è necessario spezzettarlo in modo che rientri in tale vincolo. Siccome TCP e UDP richiedono segmenti già completi, l'host destinazione deve riassemblare il segmento e per farlo utilizza tre campi: -flag, identificatore e spostamento laterale di frammentazione-. L'identificativo è dato dall'host mittente alla creazione del datagramma e viene incrementato per ogni datagramma successivo inviato. Ogni frammento mantiene tale campo di identificazione. Il destinatario riceve una serie di datagrammi dallo stesso host mittente e capisce dall'identificatore quali frammenti sono dello stesso datagramma in modo da ricomporre il tutto.

9) La frammentazione del datagramma differisce se UDP o TCP? Motivare la risposta e spiegare la frammentazione

La modalità di frammentazione non cambia in quanto a questi due protocolli a livello di trasporto interessano solo i segmenti completi, che vengono recapitati dal livello di rete già assemblati. TCP

e UDP quindi non hanno nulla di cui occuparsi sulla frammentazione in quanto è prettamente del livello di rete.

(per come funziona la frammentazione, vedere domanda sopra)

10) Datagrammi IP, chi li compone e perché?

I datagrammi IP sono creati dall'host a partire dai segmenti di livello di trasporto, e vengono inviati al router a cui l'host è connesso. Il router prenderà quindi in input tale datagramma e lo inoltrerà all'interfaccia di uscita mediante tabelle di instradamento che vengono create tramite protocolli di instradamento che determinano il cammino minimo per raggiungere il router (e successivamente l'host) di destinazione. Dal livello di rete, i datagrammi passano a quello di collegamento diventando frame, in questo modo potranno intraprendere tale percorso per arrivare a destinazione.

Categoria 4:

Sniffing----------

1) Utilizzando wireshark quale tipo di traffico è possibile intercettare su una rete con topologia a stella?

2) Spiega in modo MOLTO dettagliato il packet sniffing in una rete a bus e in una a stella

3) un utente che usa wireshark cosa può rilevare sniffando da una rete ethernet a bus?

In una rete con topologia a stella, il tipo di traffico che è possibile intercettare dipende dal tipo di adattore. Se al centro della stella abbiamo un hub, l'host che ha wireshark attivo può sniffare tutti i pacchetti, in quanto l'hub li invia in broadcast a tutti (e sta nell'host destinatario a prendere i pacchetti, e non a scartarli come fanno gli altri host) . Se abbiamo uno switch invece, posso sniffare solo i pacchetti che vengono mandati a me (host con wireshark attivo) in quanto lo switch manda i pacchetti solo al destinatario. In caso di pacchetti mandati consapevolmente in broadcast, anche quelli possono ovviamente esser sniffati nonostante la presenza di uno switch. Sulla rete a bus invece, è possibile sniffare tutti i pacchetti in circolazione nel bus.

4) il protocollo http è sicuro allo sniffing? per quale motivo?

5) HTTP come evita lo sniffing? Motivare la risposta in modo dettagliato

http non ha alcun modo di proteggersi dallo sniffing, preso l'host di cui vogliamo sniffare i pacchetti, non ci sono impedimenti nel farlo, nel caso abbiami però il protocollo https, allora certi pacchetti contenenti password e altri dati sensibili saranno criptati, nonostante si possano sniffare senza problemi.

6) Cos'è il packet sniffing? in che ambito viene spesso usato? si può fare solo su reti wireless o anche su Lan cablate?

7) cosa si intende per sniffing e per cosa può essere usato, può essere fatto anche nelle reti wired??

Fare packet sniffing significa vedere tutto quello che passa per la scheda di rete ed è possibile farlo sia su rete wireles che lan cablate. Viene usato sia per testare e violare la sicurezza.

8) l'invio di posta elettronica tramite protocollo SMTP è sensibile allo sniffing? Argomentare la risposta dettagliatamente.

L'invio di posta tramite SMTP è sensibile a sniffing in quanto i pacchetti , anche se eventualmente crittografati, passano per la scheda di rete.

Categoria 5:

Switch----------

1) Illustra la differenza tra switch e router

I router e gli switch sono commutatori di pacchetto store and forward che inoltrano pacchetti usando rispettivamente IP (ind di rete) e MAC (ind di collegamento) quindi sono rispettivamente commutatori di livello 3 e 2. Gli switch, al contrario dei router, sono plug and play in quanto hanno ritmi di filtraggio e inoltro dei pacchetti relativamente alti (lavorano fino a livello 2). I router non sono plug and play perché il proprio IP e quello degli host collegati ad esso deve esser configurato. I router invece devono elaborare invece pacchetti fino a livello 3. Gli switch hanno una topologia di rete ristretta ad albero per evitare cicli broadcast, e contro le tempeste broadcast non hanno alcun tipo di protezione. I router invece non ammettono cicli in quanto gli indirizzi di rete sono gerarchici (lineari sono i mac).

2) Illustra la differenza tra switch e hub

Un hub è un dispositivo che, quando riceve in un interfaccia d'entrata un bit, ne amplifica il segnale, e lo trasmette in tutte le interfacce d'uscita, di conseguenza i nodi riceventi dovranno capire quale frame è per loro e quale no (in tal caso scartarlo). C'è un alto rischio di collisione, quindi i nodi dei frame interessati, dovranno ritrasmetterli. Lo switch invece riesce a gestire le collisioni in quanto ad un frame in entrata fa corrispondere la sua precisa interfaccia in uscita, senza generare traffico inutile.

3) Cosa significa che gli switch fanno autoapprendimento?

Si intende la capacità dello switch di costruire in modo autonomo e dinamico le tabelle di commutazione senza utilizzare alcun protocollo o intervento umano. Queste tabelle contengono la corrispondenza interfaccia –indirizzo mac

4) Gli switch usano ARP? motiva la risposta e fai una breve descrizione degli switch

No, gli switch non utilizzano tale protocollo in quanto ARP serve alla risoluzione indirizzo IP-MAC. Gli switch invece per inoltrare i frame all'interfaccia di uscita relativa, si limitano all'utilizzo delle tabelle di inoltro che mantengono la corrispondenza mac-interfaccia d'uscita. Queste tabelle sono configurate in modo autonomo dallo switch stesso, il quale è un apparato al centro della topologia a stella utilizzata dalle LAN Ethernet. Ha il compito di ricevere i frame in ingresso dal liv di collegamento e inoltrarli nei collegamenti in uscita.

5) In che modo si configurano le tabelle d'inoltro di uno switch?

Le tabelle di inoltro di uno switch si autoconfigurano, in quanto gli switch fanno autoapprendimento della situazione della rete, riuscendo a costruire autonomamente e dinamicamente le tabelle di commutazione senza utilizzare alcun protocollo o l'intervento umano.

6) Descrivere a livello collegamento l'apparato switch.

Lo switch è un apparato al centro della topologia a stella utilizzata dalle LAN Ethernet. Ha il compito di ricevere i frame in ingresso dal liv di collegamento e inoltrarli nei collegamenti in uscita. I nodi indirizzano i frame ad altri nodi, senza sapere degli switch nel mezzo. Tramite filtraggio, lo switch capisce se inoltrare o scartare i frame e tramite tabella di inoltro, capisce a quale interfaccia in uscita mandare il frame (usando la corrispondenza MAC –interfaccia). Le tabelle di inoltro si autoconfigurano, lo fanno quindi in maniera autonoma e dinamica, senza protocolli e interventi umani.

Categoria 6: Pila protocollare/ livelli

---

1) Spiega la differenza tra pila protocollare osi e quella utilizzata attualmente; per quest'ultima indicare per ogni livello un protocollo e spiegarlo brevemente

2) Descrivere la pila protocollare, fare l'esempio di un protocollo per ogni livello. Fare poi il confronto con il modello ISO-OSI

3) Quali sono i servizi messi a disposizione dal livello di trasporto in Internet ?

4) Descrivi la pila dei livelli Internet ed elenca, dove possibile, almeno un protocollo per livello

5) Descrivi quali sono gli obiettivi e le funzioni del livello di collegamento in Internet

6) Parla del Livello di trasporto

Nell'attuale pila protocollare abbiamo 5 livelli/strati:
-> *Liv applicazione*: è il livello in cui risiedono le applicazioni, del sistema terminale, che necessitano di interagire con la rete per interagire tra loro (per scambio di dati etc). I pacchetti di questo livello sono chiamati messaggi. Alcuni protocolli sono *SMTP,DNS,FTP,HTTP* <-livello implementato via software
-> *Liv trasporto*: livello che trasferisce i segmenti (pacchetti di liv trasporto) tra il lato client e server di un applicazione. Alcuni protocolli sono *TCP* e *UDP* <- liv implementato via software
-> *liv rete*: si occupa di trasferire i datagrammi da un host all'altro. Quindi da sorgente a destinazione (dei quali vede i due router, tralasciando quelli intermedi del percorso). Comprende i *protocolli di instradamento (IP)* che decidono il percorso da intraprendere tra i due host. <- implementazione hardware/software
-> *Liv di collegamento:* livello che instrada i frame nei router che si interpongono fra origine e destinazione. Protocollo *Ethernet* <- via hardware
-> *Liv fisico:* trasferire i bit da un nodo al successivo. Il mezzo fisico può variare.
Il modello *ISO-OSI* comprendeva, sotto il livello applicazione, due ulteriori livelli, *presentazione* (cifratura, gestione problematiche little/big endian) e *sessione* (sincronizzazione, controllo dei dati). Ora questi servizi sono implementati nel livello applicazione.

7) Cos'è l'incapsulamento della pila protocollare?

E' quella procedura che prende i messaggi a livello applicazione e li passa al livello sottostante

(trasporto) il quale aggiunge la propria intestazione così diviene un segmento che passa al livello di rete diventando un datagramma che passa a livello di collegamento diventando frame. Ogni livello mantiene i dati arrivati dal livello sopra e aggiunge la propria intestazione di livello.

8)individuare nella pila protocollare un protocollo che usa il controllo di flusso

Tale protocollo è il TCP (livello di trasporto) che utilizza il controllo di flusso per non mandare in overflow il buffer dell'host destinatario. Regola infatti l'invio dei segmenti TCP in modo che l'host riesca ad elaborarli tutti senza creare code che potrebbero portare alla perdita del pacchetto. Quindi ad ogni scambio di segmenti, il ricevente informa il mittente di quanto può ancora ricevere grazie alla variabile di ricezione.

9) Descrivi brevemente almeno tre protocolli del livello applicazione.

HTTP-> Definisce il modo in cui i client web richiedono le pagine ai server web. Questa avviene attraverso una richiesta http degli oggetti referenziati all'interno di una pagina web. Tra client e server avviene una connessione TCP, nella quale viene effettuata la richiesta e ricevuta la risposta. I 2 tipi di messaggi hanno diverse caratteristiche.

FTP-> Protocollo di trasferimento file ad un host remoto. Tale trasferimento avviene con l'utilizzo di 2 connessioni TCP (*controllo* -> in quanto è un *protocollo con stato* e *dati*-> per l'invio dei dati vero e proprio)

SMTP-> Protocollo che gestisce l'invio della posta elettronica tra server mittente e destinatario. Utilizza il protocollo di trasporto TCP. È un *push protocol* in quanto avvia la connessione per inviare dei dati (al contrario di http che è pull protocol in quanto avvia la connessione per ricevere dati)

10) Dei seguenti termini descriverne le caratteristiche e descrivi il livello a cui appartengono: messaggio, datagramma, segmento, frame e FTP.

11) Dato il seguente elenco di termini indicare in quale livello si trovano e illustra tale livello: POP (posta elettronica), messaggio, frame , datagramma, IP.

12) Descrivere di cosa si tratta e a quale livello appartiene: messaggio, csma, segmento, frame, skype

POP(posta) -> app MESSAGGIO -> app FRAME -> coll DATAGRAMMA -> rete
IP -> rete CSMA -> coll SEGMENTO -> trasp SKYPE -> app FTP -> app

13) Definizione di Protocollo e definire la sua importanza all'interno della rete.

Un protocollo definisce il formato e l'ordine dei messaggi scambiati tra due o più entità in comunicazione, così come le azioni intraprese in fase di trasmissione e/o ricezione di un messaggio o un altro tipo di evento. E' molto importante all'interno della rete perché permette di far comunicare entità tra loro diverse grazie all'utilizzo di regole e procedure comuni. I protocolli sono infatti standard definiti nell'RFC e IETF

14) Differenze tra datagramma, messaggio, segmento, frame.

Rete, applicazione, trasporto, collegamento.
Ognuno di questi appartiene ad un diverso livello della pila protocollare. Mantiene i dati del proprio livello e quello superiore, e include l'intestazione del proprio livello.

Categoria 7:

Perdita----------

1) Cos'è un evento di perdita? Come reagisce il sistema?

2) Spiega tutti i casi in cui si può verificare la perdita di pacchetti

3) Come può essere perso un pacchetto nella rete Internet?

4) Perché nella rete internet avviene la perdita di pacchetti? Motiva tecnicamente la risposta.

Nonostante la rete cerchi di dare un servizio best-effort, non sempre garantisce che tutti i pacchetti giungano a destinazione, in quanto non tutti i protocolli implementano questa caratteristica. Tuttavia, anche nel caso un protocollo (come tcp) abbia tale controllo, durante il percorso un pacchetto può andare perso, per "recuperarlo" quindi sarà necessario una ritrasmissione (attraverso l'utilizzo di ACK, numeri di sequenza…). Perché un pacchetto può perdersi? Uno dei tanti motivi può essere che, il datagramma IP, giunto al router di destinazione, non trovi posto nel buffer pieno, quindi, o gli viene fatto posto (alcuni protocolli a liv di rete possono decidere se scartare un pacchetto o toglierne un altro in coda per far posto al nuovo) o questo verrà perso nella rete. Stessa cosa può avvenire a livello di collegamento, dove la verifica di collisione può portare alla perdita di frame.

Categoria 8:
Ritardi----------

1) Definire il ritardo di elaborazione del nodo, ritardo di accodamento,ritardo di trasmissione, ritardo di propagazione.

2) Spiegare i vari tipi di ritardo.

3) Parla del ritardo di elaborazione correlato con l'intensità di traffico

4) Spiega a cosa è dovuto il ritardo in un collegamenti end-to-end in modo dettagliato.

5) Parla della congestione legata alla intensità di traffico

Abbiamo diverse tipologie di ritardo che complessivamente producono il ritardo end-to-end, ovvero dall'origine alla destinazione. Questi ritardi sono:

- *Elaborazione (processing delay)* -> tempo richiesto per esaminare il pacchetto (la sua intestazione) e decidere dove mandarlo in uscita. Può effettuarsi anche in caso di eventuali errori sui bit.
- *Accodamento (queue delay)* -> Dipende principalmente da quanti pacchetti ci sono nel buffer e da quanto tempo impiegato per elaborare ognuno di tali pacchetti e in base a come questi arrivano al buffer.
- *Trasmissione (transmission delay)* -> E' il ritardo che tiene conto di quanto ci mette il pacchetto ad uscire dal router, in seguito al fatto che tutti i pacchetti prima di lui siano stati spediti. (store and forward delay) Dipende dalla lunghezza del pacchetto.
- *Propagazione (propagation delay)* -> Tempo richiesto per la propagazione di un bit da un router all'altro, varia in base alla distanza tra i due router e dal mezzo fisico di collegamento.

Tra ritardo di elaborazione e intensità di traffico non vi è alcun rapporto in quanto tale ritardo è trascurabile. *L'intensità di traffico si relaziona invece col ritardo di accodamento* in quanto è associata a come i pacchetti arrivino nel buffer, se a frequenza costante o periodica.

6) Ritardo di accodamento: in che modo è legato al traffico di rete?

Il ritardo di accodamento, che è definito da quanti pacchetti da elaborare e spedire vi sono nel buffer, cambia in base alla frequenza di arrivo dei pacchetti (quindi alla natura di traffico entrante). Infatti questi possono giungere nel buffer con frequenza costante o periodica. Nel primo caso, c'è il rischio che il buffer vada in overflow in quanto la velocità di elaborazione dei pacchetti è inferiore a quella di arrivo, in poche parole, il buffer non è in grado di smaltire velocemente i pacchetti e si satura velocemente. Se invece l'arrivo è periodico, un pacchetto viene smaltito prima dell'arrivo del successivo, e così non si rischia overflow.

Categoria 9: Ethernet

---

1) Il campo "indirizzo destinazione" di un pacchetto ethernet ha al suo interno l'indirizzo del destinatario?

Il campo indirizzo di destinazione contiene l'indirizzo mac dell'adattatore di destinazione, in modo da poter identificare dove mandare il frame.

2) Parla di Ethernet

Ethernet è il protocollo livello collegamento maggiormente diffuso per LAN. E' iniziato con una tecnologia a bus, per poi evolversi a stella con hub centrale, mentre attualmente, lo switch ha preso il posto dell'hub, in questo modo si evitano le collisioni. Ethernet, che fornisce un servizio senza connessione e non affidabile, entra in gioco quando si desidera inviare un datagramma IP da un host all'altro, all'interno della stessa LAN ethernet. Il frame ethernet è composto da 6 campi; preambolo, indirizzo di destinazione/origine (mac), tipo, crc. Nel caso di topologia a stella con hub si utilizza il protocollo csma\cd per la rilevazione delle collisioni, mentre nel caso di switch, non utilizziamo alcun protocollo di accesso multiplo (protocollo MAC).

3) Illustra l'accesso multiplo nella tecnologia Ethernet e cosa succede in caso di backoff/ tempo di attesa esponenziale

Per l'accesso multiplo, ethernet utilizza il protocollo csma/cd (non prevede l'utilizzo di slot) che si basa su due regole ben precise: -rilevazione della portante, ovvero se nel canale vi è qualche altro nodo che trasmette frame, allora io, nodo che devo trasmettere, aspetto un tempo casuale, se si verifica la stessa condizione, allora riaspetto ulteriormente, mentre se il canale è libero da trasmissioni, trasmetto. -rilevazione della collisione, se sto trasmettendo un frame e un altro adattatore sta trasmettendo, interrompo la mia trasmissione e mando un segnale di jam per indicare il disturbo individuato nel canale. Prima di riprovare a ricominciare la trasmissione, aspetto un tempo casuale stabilito in modo arbitrario. La fase in cui decide tale tempo, si chiama attesa esponenziale, ovvero quando riscontra la n-esima collisione consecutiva, stabilisce un valore tra 0 ed 2(m)-1 con m il minimo tra n e 10, ovvero cerca di stabilire il tempo in base a

quanti adattatori sono coinvolti nella collisione.

4) Cos'è il preambolo e quali sono gli altri campi del frame ethernet?

Il frame ethernet è composto da 6 campi. Preambolo, 8 byte, serve per la gestione degli adattatori riceventi e sincronizzare i loro orologi con quello del trasmittente. Abbiamo poi due campi per indirizzo di collegamento (MAC) destinazione e sorgente utilizzati per sapere a quale nodo della sottorete inviare il frame. Un campo tipo, che consente ad ethernet di supportare vari protocolli di rete (nonostante IP sia il più utilizzato, non è l'unico), tutto questo per far comunicare un protocollo con quello di livello superiore. Campo dati, contiene il datagramma IP che rientra nel MTU, ovvero il massimo numero di byte che il frame può assumere. Se il datagramma è più lungo, verrà frammentato dall'host. L'ultimo campo, il CRC, serve per la rilevazione degli errori nei bit del frame.

5)Ethernet: è considerabile un protocollo affidabile?

Categoria 10: Tcp/Udp

---

1) Cosa sono i numeri di sequenza e di riscontro in tcp?

Numeri di sequenza: Sono i numeri che indicano il primo byte del segmento nel flusso. Tcp infatti vede i dati come un flusso di byte ordinati. Servono per capire se i pacchetti sono duplicati o fuori ordine. Questi sono scelti in modo casuale in modo i pacchetti eventualmente smarriti non vengano interpretati come validi in una successiva connessione tcp.
Numeri di riscontro: sono i numeri che indicano il numero di sequenza successivo a quello ricevuto. Quindi indicano il pacchetto che mi aspetto di ricevere dal destinatario.

2) Spiega in dettaglio come reagisce il tcp al timeout

Dopo il timout, si deduce che la rete sia così congestionata da non riuscire a recapitare i pacchetti a livello di trasporto, quindi, nella ritrasmissione del pacchetto (in quanto tcp garantisce l'arrivo di tutti i pacchetti) si attua un controllo di congestione, con lo scopo di non trafficare troppo la rete, andando a modificare la propria frequenza d'invio in base al parametro dato dalla finestra di congestione. Al verificare di questo evento di perdita (timeout) entriamo in una fase di partenza lenta, ovvero la finestra di congestione si setta ad 1 MSS e cresce in modo esponenziale fino ad un valore di soglia (che corrisponde alla metà del valore della finestra prima dell'evento di perdita). Da questo valore, fino ad una prossima perdita, la crescita sarà lineare (aumenta di 1 MSS ad ogni RTT), tipica del meccanismo AIMD "incremento additivo, decremento moltiplicativo".

3) Come avviene la chiusura di una connessione TCP?

Sia client che server possono decidere di porre fine alla connessione TCP. Nel nostro esempio prendiamo che sia il client a decidere. Questo manda un segmento speciale di FYN, quando il server lo riceve manda un riscontro (ACK) più un segmento di FYN. Il client manda un riscontro a quest'ultimo segmento e la connessione è completamente chiusa e tutte le risorse, come i buffer, vengono deallocati.

4) In quale protocollo, tra UDP e TCP, è più facile inserirsi nella conversazione?

E' più facile udp. Nel caso di TCP c'è un three way handshake che comporta uno scambio di numeri di sequenza scelti casualmente, difficili da andare a scoprire. Nel caso di UDP non esiste il numero di sequenza quindi basta sapere chi è il destinatario (il numero di porta del dest) con cui sta
comunicando. In tcp invece c'è una fase di sincronizzazione (SYN) di protocolli che devono garantire
sicurezza (https) viene implementato col protocollo tcp quindi è molto più facile inserirsi in DNS (forgiare
la risposta finta e la domanda finta) che ritrovare la richiesta di una pagina web http.

5) Descrivi i flag di tcp: ack , syn, fin, psh, urg

ACK: settato in caso di avvenuto riscontro del pacchetto da parte del destinatario. Significa che è arrivato correttamente.
PSH: settato nel caso in cui il destinatario deve subito mandare i dati al livello superiore.
URG: Settato nel caso in cui i dati siano stati marcati come urgenti dal mittente
SYN e FYN: servono rispettivamente per aprire e chiudere la connessione TCP

6) Perchè tcp è definito equo? Definire vantaggi e svantaggi. Perché udp non può essere considerato tale?

TCP è definito equo in quanto applica il controllo di congestione e quindi tutte le connessioni TCP sono trattate in modo equo, alla pari, così ciascuna si prende una stessa quantità di banda. UDP non è equo invece perché "spara i dati a raffica" non effettuando alcun controllo e quindi le applicazioni che usano UDP non devono limitare la propria frequenza, anche a rischio di saturare la banda di altri. L'equità in TCP è uno svantaggio nel senso che non rende il protocollo adatto alle

applicazioni che preferiscono perdere qualche pacchetto rispetto al dover moderare l'invio dei dati a causa del controllo di congestione che TCP applica.

7) Quali sono i servizi offerti da TCP?

-*Controllo degli errori:* grazie all'utilizzo del campo checksum (presente anche in UDP) il destinatario può controllare eventuali alterazioni di bit nel segmento
-*Controllo di flusso:* siccome client e server hanno buffer limitati, il mittente deve regolare l'invio dei dati in modo tale da non mandare il buffer destinatario in overwflow. Quindi ad ogni scambio di segmento il ricevente informa il mittente di quando è ancora capace di ricevere grazie alla finestra di ricezione. Questa è condizionata dalla velocità con cui l'applicazione legge il segmento dal buffer.
-*Tutti i segmenti arrivano, in modo ordinato*. Questo grazie all'utilizzo di riscontri (ACK,NAK) e a timeout che portano alla ritrasmissione di pacchetti duplicati/persi/fuori ordine
-*Controllo di congestione:* imporre ad ogni mittente un limite alla frequenza di invio sulla propria connessione in modo che la rete non ne sia troppo congestionata, quindi che il traffico non renda difficile il recapito dei pacchetti da un host all'altro (e ovviamente tra i nodi intermedi della rete).

8) Perché udp è senza connessione?

UDP è senza connessione in quanto non esiste alcuna forma di "handshake". "spara" i dati senza stabilire alcuna connessione, quindi evita il ritardo che si avrebbe nello stabilirla. Questo non aver connessione non garantisce l'arrivo di tutti i datagrammi UDP e neanche un ordine giusto di arrivo. Quindi non implementa molte delle caratteristiche di TCP (come controllo di flusso etc) e rimane un datagramma di intestazione molto più leggero. Ciò lo rende un protocollo più adatto per applicazioni che tollerano ritardi e perdite e preferiscono velocità nell'invio dei dati.

9) Nei segmenti UDP sono presenti l'indirizzo IP del mittente e del destinatario? E la porta del mittente? Motiva le risposte

I campi di indirizzo IP di mittente e destinatario non sono presenti. Sono invece presenti i numeri di porta (origine e dest). Il numero di dest serve per identificare la socket a cui inviare il datagramma UDP in modo da farlo arrivare all'applicazione associata. Quello di origine invece serve nel caso in cui il destinatario debba rispondere al mittente quindi deve sapere la porta del processo a cui rispondere.

10) Come funziona il checksum udp?

E' un campo utilizzato dall'host ricevente per controllare la presenza di errori (come l'alterazione dei bit) nel segmento. Il mittente somma tra loro le parole dei dati e fa poi il complemento a uno (detto anche bit a bit). Tale somma viene inserita nel campo checksum. Al lato ricevente si somma la checksum con le parole iniziali. Se il risultato è composto da tutti uno allora non c'è stata alcuna alterazione dei bit. UPD si limita a rilevare gli errori ma non può fare nulla per correggerli.

11) Controllo di congestione. Cos'è? Da chi è implementato? Come?

E' implementato da TCP. Questo controllo si riferisce al regolare la propria frequenza di invio nella connessione stabilita tra mittente e destinatario in base al traffico presente in rete. Più c'è traffico e più c'è congestione e minore è il tasso trasmissivo. Questo controllo si serve del campo di finestra di congestione grazie alla quale imponiamo il vincolo alla frequenza.
Tre meccanismi appartenenti a questo controllo sono:
-*AIDM (incremento additivo, decremento moltiplicativo):* la finestra di trasmissione è pari ad 1 MSS e aumenta della stessa quantità ad ogni RTT fin quando non si riscontra un evento di perdita (3 ack duplicati o timeout) e in questo caso la finestra si dimezza. Riprende poi a incrementarsi fino a nuova perdita. Con congestion avoidance indichiamo l'incremento lineare del protocollo di congestione tcp.
-*Partenza lenta:* l'incremento della frequenza avviene in modo esponenziale e la finestra di congestione si raddoppia ogni RTT fino ad una perdita di pacchetto. In tal caso tale finestra si dimezza e riprende a crescere linearmente.
-*Timout:* reagisce in modo diverso che nel caso con 3 ack. Infatti abbiamo una fase di partenza lenta fino a quando la mia finestra non raggiunge una soglia (che corrisponde alla metà della finestra prima del timeout). Dopodichè la finestra comincerà ad aumentare linearmente come avviene in AIDM

12)Perché la connessione tcp può essere definita orientata alla connessione?

Perché prevede un 3 way handshake tra mittente e destinatario, ovvero avviene lo scambio i 3 segmenti (i primi due non portano dati utili) che fanno instaurare la connessione (full duplex) nella quale passerà un flusso di segmenti il cui arrivo al destinatario è garantito. Il tcp quindi prevede comandi di apertura e chiusura della connessione. Durante la fase di instaurazione vengono allocate da client e server le risorse necessarie, come i buffer.

13) nel protocollo di trasporto TCP in cosa consiste il meccanismo della ritrasmissione rapida?

Tale meccanismo consiste nella trasmissione del segmento prima che scada il timer, dopo aver riscontrato 3 ack duplicati. Serve per evitare ritardi end-to-end dati dal dover aspettare la

scadenza del timer prima di poter rinviare il pacchetto. Infatti il riscontro di tali tre ack, porta il mittente a rilevare la perdita dei pacchetti senza dover aspettare il timeout per capirlo.

14) Utilizzo del piggyback, cosa si intende, spiegare dettagliatamente

E' un metodo che prevede di includere un dato in un messaggio successivo. Per la precisione, nel protocollo TCP, il mittente si aspetta un ACK di riscontro (ovviamente non ci riferiamo ad ACK che possono andare perduti) dal destinatario per il segmento che da questo è stato appena ricevuto. nel caso in cui tra mittente e destinatario ci sia l'invio di ulteriori segmenti, il destinatario invia l'ACK non in un messaggio a sé, ma all'interno del messaggio successivo che deve inviare al mittente. Con questa pratica si riducono il numero di messaggi complessivi.

15) cosa si intende per incremento additivo e decremento moltiplicativo

Meccanismo utilizzato nel controllo di congestione del protocollo TCP.
-AIDM (incremento additivo, decremento moltiplicativo): la finestra di trasmissione è pari ad 1 MSS e aumenta della stessa quantità ad ogni RTT fin quando non si riscontra un evento di perdita (3 ack duplicati o timeout) e in questo caso la finestra si dimezza. Riprende poi a incrementarsi fino a nuova perdita. Con congestion avoidance indichiamo l'incremento lineare del protocollo di congestione tcp.

16) definire il segmento di synack e in che modo viene utilizzato.

Il segmento di synack detto anche di "connessione garantita" è il secondo segmento utilizzzato per l'apertura di una connessione TCP. Viene mandato dal server al client in risposta al segmento di syn (il primo). Sta a significare che il segmento è stato correttamente ricevuto. Il client manda poi il suo ack (come riscontro del synack) e la connessione è aperta. Questa procedura è detta 3way handshake in quanto avviene lo scambio di 3 segmenti (i primi due non trasportano dati utili)

17) Cosa si intende per partenza lenta?

E' un meccanismo attuato nel controllo di congestione del protocollo TCP. Con questo metodo, la finestra di congestione si raddoppia ad ogni RTT mentre la frequenza cresce in modo esponenziale fin quando non si verifica un evento di perdita che può essere il timeout o i 3 ack duplicati. Dopodichè infatti la finestra si dimezza e riprende a ricrescere in modo lineare.

18) Spiegare cos'è il synflood, e con quale protocollo funziona

Funziona col protocollo di trasporto TCP. È un attacco di tipo DOS che prevede nell'inviare segmenti di SYN (se questi sono inviati da diverse sorgenti, l'attacco diventa di tipo DDOS) in modo che il server allochi numerose risorse (come buffer), mantenendo numerose connessioni mezze aperte, in modo da non averle disponibili per chi deve instaurare in modo lecito una connessione.

19) Tcp è half duplex o full duplex? Esistono condizioni in cui tcp è half duplex?

TCP è di norma full duplex, ovvero il flusso dei dati tra due host può verificarsi contemporaneamente nelle due direzioni. Può considerarsi half-duplex dopo che il server ha ricevuto il segmento di FYN per la chiusura della connessione e l'ha riscontrato, quindi la comunicazione client-server non può più avvenire. Il server invece può ancora comunicare col client fin quando non chiude la connessione a sua volta.

Categoria 11: router/instradamento

1) Cosa sono inoltro e l'instradamento?

Inoltro: è l'azione (interna al router, quindi locale) con cui il router determina a quale interfaccia d'uscita mandare i pacchetti arrivati dalle interfacce d'entrata.
Instradamento: consiste nel determinare il percorso che il pacchetto a livello di rete deve fare per passare dal router mittente a quello destinazione. Si stabilisce grazie alle tabelle di inoltro che permettono di stabilire algoritmi di instradamento.

2) Differenza tra protocolli instradamento intra e inter?

Affinchè un pacchetto arrivi da un router origine ad uno destinazione, è necessario stabilire un percorso da seguire, ovvero bisogna instradare il pacchetto. L'instradamento inter si riferisce a quello tra *autonomous system* (ovvero tra host e router che appartengono alla stessa gestione amministrativa) mentre quello intra, si riferisce all'instradamento tra *autonomous system (AS)*.All'interno dello stesso AD, i pacchetti sono instradati tramite protocolli link state (RIP) o dv (OSPF), mentre tra AS, col protocollo BGP. Gli *autonomous system* sono stati concepiti per risolvere i problemi di scala e autonomia amministrativa.

3) spiegare il problema di blocco in testa nell'ambito dei router (HOL)

All'interno di un router, nella struttura di commutazione, i pacchetti vengono inoltrati dalle rispettive origini alle rispettive destinazioni. A causa di un eventuale accodamento nelle porte d'uscita e entrata, possiamo avere il problema del blocco in testa alla fine. Questo consiste nell'avere un pacchetto che potrebbe essere inoltrato in quanto la sua destinazione è libera, tuttavia è bloccato da un pacchetto in testa alla fila che non può essere inoltrato a causa della propria destinazione bloccata.

4) Parla della patata bollente

Siamo a livello di rete. E' un tipo di instradamento che prevede che l'AS si sbarazzi il prima possibile di un pacchetto, nel modo meno costoso. Il pacchetto viene quindi inviato dal router gateway (quello di confine) che richiede un costo minore per raggiungerlo. Ovviamente tale router sarà collegato alla destinazione che il pacchetto deve prendere.

5) Router, descriverne la struttura e le operazioni che esegue se arriva in input un datagramma IP

Il router è composto da porte d'ingresso, uscita, struttura di commutazione e processore di instradamento. Le porte d'ingresso ricevolno i pacchetti di rete e sono le terminazioni elettriche di linea. Hanno poi la funzione di elaborare a livello di collegamento e vi è al loro interno un modulo di elaborazione/ricerca/inoltro dove molto spesso risiedono le tabelle di inoltro (copie sempre aggiornate di quella che risiede nel processore) in modo da poter inoltrare direttamente i pacchetti (in caso di buona capacità di elaborazione delle porte) senza interpellare il processore. Un datagramma infatti quando arriva viene o mandato al processore o direttamente inoltrato nelle porte d'uscita. Il processore, con la tabella d'inoltro stabilisce a quale porta di destinazione mandare il datagramma. L'inoltro "fisico" è compito della struttura di commutazione la quale attraverso bus recapita i datagrammi alle rispettive porte d'uscita (analoghe a quelle d'entrata). Se vi è accodamento nelle porte possiamo avere una perdita di pacchetti di rete.

6) Spiegare dettagliatamente il funzionamento dell'algoritmo di instradamento a vettore distanza e spiegare il meccanismo con il quale determina i cammini minimi

Un algoritmo di instradamento a vettore distanza (DV) è asincrono, non tutti i nodi devono operare al passo con gli altri, iterativo, processo si ripete fin quando non vi è ulteriore scambio di informazioni tra router vicini (è anche auto-terminante), e distribuito (ogni nodo riceve le info di costo del collegamento dai router vicini). All'inizio ogni router conosce solo il costo di collegamento dei vicini, dopo un procedimento di tipo iterativo, il nodo conosce il percorso fino alle destinazioni. E' un algoritmo di tipo decentralizzato. Come funziona? Talvolta, un nodo invia una copia del proprio vettore di stanza ad ogni vicino. Quando un nodo riceve tale vettore, aggiorna il proprio con la formula di *Bellman Ford*, se si è verificato tale aggiornamento il nodo invia il nuovo vettore distanza ad ognuno dei vicini che a sua volta aggiornerà tale vettore. In riassunto, ogni noto attende aggiornamenti dai propri vicini, quando li riceve calcola il proprio nuovo vettore distanza e lo distribuisce ai vicini.

7) Discuti del costo computazionale e il numero di messaggi richiesti di un algoritmo a stato della connessione linkstate

Il costo computazione è di n(n+1)/2. Infatti, nella prima iterazione dell'algoritmo di Dijsktra per determinare il cammino minimo, abbiamo n nodi da visitare per poter determinare quello di costo minimo, alla seconda iterazione, n-1 e così via. Nel caso peggiore invece arriviamo ad una complessità quadratica di O(n^2). Quindi per migliorare le prestazioni dell'algoritmo possiamo ricorrere ad una struttura ad heap riuscendo a determinare ogni minimo ad un tempo logaritmico invece che lineare. Il numero di messaggi richiesti da LS è invece di complessità O(|N|*|E|) poiché bisogna sapere di ogni nodo, il costo del relativo collegamento nella rete. Se un collegamento cambia costo, bisogna comunicarlo a tutti i nodi.

Categoria 12:

Mail----------

1) In cosa consiste MIME e perché è massicciamente utilizzato?

Il MIME è un meccanismo che permette di inviare contenuti diversi dal testo ASCII nella mail. Introduce due nuove righe di intestazione in modo che il ricevente sia in grado di convertire il messaggio nella forma originaria. È massicciamente utilizzato in quanto al giorno d'oggi è molto frequente scambiarsi allegati multimediali come foto o video.

2) A che cosa serve SMTP?

SMTP *(simple mail transfer protocol)* è il protocollo applicazione utilizzato per l'invio di posta dal server di posta mittente a quello del destinatario. Utilizza il protocollo di trasporto TCP, in quanto affidabile, ed è un protocollo lato client per quanto concerne l'invio della posta dal proprio server, mentre è lato server per quanto riguarda la ricezione della mail (quindi server web del destinatario). Intestazione e corpo del messaggio sono in ASCII a 7 bit e oggi questa è una forte limitazione. Utilizza connessioni persistenti in quanto nella stessa connessione può mandare più mail. E' un *push protocol* perché la connessione TCP viene inizializzata dalla macchina che vuole spedire la mail.

3) Nell'ambito dell'invio di un e-mail che ruolo svolge Pop3?

Nell'ambito di invio della mail non svolge alcun ruolo in quanto POP3 è solo un protocollo di accesso alla posta per vedere le mail dal proprio server di posta al pc o un altro dispositivo. Consente una gestione basilare della posta in quanto permette di scaricarle, visualizzarle ed

eventualmente cancellarle.

4) Descrivere il protocollo POP3 e a cosa serve.

Pop3 è un protocollo di accesso alla posta elettronica utilizzato al fine di visualizzare sul proprio pc i messaggi che risiedono sul proprio server di posta. Pop3 entra in azione quando il client apre una connessione TCP verso il server. Quando tale connessione è stabilita, pop3 procede con la fase di autorizzazione (l'utente invia username e password), transazione (l'agente utente recupera i messaggi) e nella fase di aggiornamento vengono eventualmente cancellati i messaggi marcati da cancellare. Ad ogni comando, il server risponde con +OK e –ERR. POP3 ha la modalità '*scarica e cancella'* e *scarica e 'mantieni'*.

Categoria 13: FTP

---

1) FTP e' un protocollo con o senza stato? descriverne il funzionamento

2) Parla di FTP in termini tecnici

3) FTP: come funziona la comunicazione?

FTP *(file transfer protocol)* è un protocollo di livello applicazione di trasmissione di file ad un host remoto. Utilizza due connessioni TCP parallele di controllo e dati. La prima serve a mandare username e password (per identificare l'utente) , i comandi per cambiare la directory remota e quelli per inviare e ricevere file. La connessione dati serve invece al trasferimento vero e proprio dei dati. Siccome sono usate 2 connessioni TCP separate tra loro, FTP si dice *fuori banda* (la connessione di controllo resta aperta per tutta la sessione, mentre per ogni dato inviato si crea una nuova connessione dati) ed è un protocollo con stato in quanto mantiene informazioni sull'utente che ha la connessione aperta. Tutti i comandi client-server e le risposte server-client sono inviate in ASCII a 7 bit quindi sono comprensibili all'utente (come in http)

4) FTP e NAT che problemi ci sono? che soluzioni possono essere adottate?

Un router sotto NAT è in grado di fare solo connessioni uscenti e non ne accetta di entranti. Siccome FTP è un protocollo applicazione che utilizza TCP, il client deve instaurare una connessione. Il client quindi potrà usare una connessione uscente per informare il server che ne vuole aprire una per mandare i file, ma la connessione che il server instaura verso il client è per quest'ultimo entrante, quindi non potrà accettarla e terminare il 3way handshake. Per ovviare al problema, quando il cclient avvisa il server di voler instaurare una connessione, può richiedere che la seconda connessione (normalmente server-client) avvenga client-server quindi uscente e accettabile per il router sotto nat.

Categoria 14: P2P/client server

---**1) Differenza tra p2p e architettura client-server**

Nell'architetture client-server gli host hanno funzioni distinte di server e client. Il primo ha un IP fisso e deve essere sempre attivo mentre il client ha indirizzo ip dinamico e può avere attività saltuaria. I client non possono comunicare tra loro direttamente ma possono comunicare col server in qualsiasi momento richiedendo un oggetto che verrà inviato dal client. Solitamente i server sono in cluster di host detti server farm. Questo tipo si architettura richiede alti costi di manutenzione dell'infrastruttura.

Nell'architetture P2P invece la presenza di server è inesistente (o quasi) e tutti i client possono comunicare tra loro (coppie di host-> peer) siccome svolgono funzioni sia di client che di server. Il loro IP non deve essere necessariamente fisso. È un architettura scalabile in quanto i peer richiedono un servizio generando lavoro ma allo stesso tempo forniscono una parte del servizio agendo da server, risponendo alle richieste di altri peer. Non serve una grande infrastruttura. il problema è gestire la dispersione dei file.

2) p2p sotto nat: come funziona?

La tecnologia nat permette di risparmiare indirizzi IP, evitando che host fuori dalla rete domestica creino una connessione con un host all'interno di questa. Il problema però è quello che un host sotto nat non può ricevere connessioni entranti. Nel p2p, affinchè due host sotto nat possano comunicare è necessario che usino un intermediario, ovvero un nodo chiamato *relay* (senza nat) selezionato dai leader di gruppo dei 2 host che vogliono comunicare. L'host 1 si collega al proprio leader di gruppo, stessa cosa fa l'host 2 e questi host comunicheranno poi tramite relay.

3) cosa significa che p2p è scalabile? Esempio del trasferimento del file.

La scalabilità è data dal fatto che ogni host fa richieste al server generando un carico di lavoro ma

al tempo stesso fa da server riuscendo a soddisfare richieste di altri host (client). Di conseguenza iil lavoro si bilancia in modo da non gravare su un unico server (o server farm).
Il trasferimento del file avviene col la richiesta del client di uno specifico file presente in un host server. Il client può ricevere pezzi di file da diversi server, fino al suo completamento. Nel frattempo il client funziona da server dando le parti di file, che lui ha ottenuto, agli altri client che cercano il file.

4) Come funziona il protocollo Bit Torrent? Illustrare le specifiche tecniche.

Il funzionamento di questo protocollo si basa sul *torrent*, ovvero l'insieme di tutti i *peer (coppie di host)* che partecipano alla condivisione di un file. I vari peer (che non hanno necessità di essere sempre attivi, e anno IP dinamico) scaricano parti (*chunck* da 256kbyte) del file l'uno dall'altro e quando un peer entra per la prima volta nel torrent non ha alcuna parte di quel file ma la ottiene scaricandola dagli altri peer. Quando un peer ha ottenuto l'intero file può rimanere in upload a condividerlo oppure può lasciare il torrent. NB: per evitare che tutti si limitino a scaricare i file, bit Torrent implementa la caratteristica di, oltre a scaricare, fare upload delle parti del file scaricate fino a quel momento. Quando un peer entra nel torrent si registra al *tracker* che tiene traccia di tutti i peer connessi al torrent. Ogni peer si collega a peer vicini, richiede le parti di file per lui più rare (quelle possedute dal numero minore di peer)

Categoria 15: Liv Collegamento

---

1) A cosa servono il checksum e il CRC? Che succede ad un frame che arriva a un adattatore e che non supera il controllo del CRC?

Il checksum (liv 2) è utilizzato dall'host ricevente per controllare la presenza di errori (come l'alterazione dei bit) nel segmento UDP/TCP. Il mittente somma tra loro le parole dei dati e fa poi il complemento a uno (detto anche bit a bit). Tale somma viene inserita nel campo checksum. Al lato ricevente si somma la checksum con le parole iniziali. Se il risultato è composto da tutti uno allora non c'è stata alcuna alterazione dei bit. Nel livello di rete (liv 3) è presente nel datagramma IP per controllare alterazioni nell'intestazione. Il CRC (codici di controllo a ridondanza ciclica) è invece una metodologia di controllo di livello di collegamento (liv 4). La stringa di bit da trasmettere è vista come un polinomio avente i bit per coefficenti. Su tale polinomio vengono effettuate operazioni aritmetiche. E' un metodo implementato via hardware. Se un frame non supera il controllo CRC viene scartato.

2) Cos'è e come funziona ARP. sicurezza in arp?

Arp è un protocollo per la risoluzione degli indirizzi di rete. Ovvero, dato un indirizzo IP (indirizzo di livello di rete), il protocollo restituisce il relativo indirizzo MAC. Questo protocollo risolve la corrispondenza IP-MAC solo per gli host della sottorete. La tabella ARP, che contiene le corrispondenze, è all'interno della RAM dei nodi. Ogni corrispondenza viene mantenuta per un un tot di tempo. Si crea un pacchetto ARP di richiesta che viene mandato a tutti i nodi della sottorete alla ricerca del MAC associato (è infatti un frame broadcast). Quello di risposta è un frame normale. Il protocollo non è molto sicuro perché non autentica le risposte quindi la corrispondenza potrebbe in realtà non essere corretta.

3) Come funzionano aloha slotted e aloha? illustrane i tempi di efficienza

Sono due *protocolli ad accesso casuale* per la gestione di acceso multiplo al collegamento. *Slotted Aloha*: Il canale è diviso in slot all'interno dei quali i frame vengono mandati alla velocità massima consentita. Ogni nodo invia quando parte il proprio slot. Se non avviene alcuna collisione, il frame giunge a destinazione, altrimenti ritrasmette con probabilità P il frame nello slot successivo. *Aloha* procede nello stesso modo, con l'unica differenza che non abbiamo slot quindi i nodi inviano il loro frame non appena ne hanno uno. L'efficienza si indica come la frazione di slot vincenti in presenza di un alto numero di nodi attivi che devono inviare più di un frame. *Slotted aloha: eff 37%, aloha: 18%*

4) Illustra cos'è e come funziona il CSMA

E' un protocollo (liv collegamento) ad accesso casuale. Si basa sulla regola di "rilevazione della portante", ovvero, se nel canale c'è qualcuno che già sta trasmettendo, allora il nodo che vorrebbe inviare, attende un intervallo di tempo casuale e poi riprova a inviare applicando lo stesso criterio. Non è però in grado di rilevare le collisioni, infatti nel caso se ne presenti una, il nodo continuerà a inviare e c'è un rischio di perdita di frame.

5) che cos è il protocollo CSMA/CD? in che cosa differisce dal protocollo CSMA?

E' un protocollo ad accesso casuale che ha in comune con CSMA la regola della rilevazione della portante, ovvero se nel canale qualche nodo sta già inviando, io, nodo attivo, non trasmetto ma attendo un tempo casuale per riprovare. Differenza invece con CSMA, quindi peculiarità di /CD è il rilevare le collisioni. Quando se ne verifica una, il nodo attiva termina la trasmissione e determina il momento in cui proverà a rinviare.

6) Cos'è un indirizzo a livello di collegamento (MAC address)?

Sono indirizzi delle schede di rete dei nodi (host e router). Sono di 6 byte, per un totale di 2^(48) possibili indirizzi. Sono in notazione esadecimali, 2 cifre per ogni byte. Sono indirizzi univoci e permanenti.

7) Perchè esistono gli indirizzi MAC? Non sarebbe sufficiente avere solo gli indirizzi IP?

Abbiamo diverse motivazioni per l'utilizzo di due indirizzi distinti. *1)* Se gli adattatori usassero l'IP, dovrebbero essere riconfigurati ogni volta che vengono accesi (in quanto l'ip cambia) *2)* Le LAN sono progettate all'utilizzo di molteplici protocolli, non solo IP e internet, quindi utilizzare indirizzi IP potrebbe ad una difficile versatilità nel caso di utilizzo di altri protocolli. Avere due diversi indirizzi, in pratica, garantisce un indipendenza tra livelli, i quali hanno i propri schemi di indirizzamento.

8)In cosa consiste il controllo di parità (monodimensionale e bidimensionale)? vantaggi e svantaggi

Il controllo di parità è una tecnica di rilevazione degli errori a livello di collegamento. Il controllo monodimensionale vede l'utilizzo di uno schema di parità pari, ovvero, il numero di bit uguali ad 1 (tra tutti quelli trasmessi) deve essere in numero pari. Abbiamo quindi d informazioni da inviare, il bit aggiuntivo d+1 dovrà prendere il valore idoneo a mantenere uno schema di parità pari. Quando il ricevente controlla, se il numero di bit uguali ad 1 è pari, non abbiamo errori, altrimenti si è verificato almeno un errore. Tale modalità permette di rilevare la presenza di errori, ma non precisamente dove questo si sia verificato. La parità bidimensionale invece, suddivide i bit dei dati in righe e colonne. Per ogni riga ed ogni colonna viene effettuato il calcolo di bit di parità avendo infine un bit di parità che permette di individuare l'errore.

9) cosa sono i protocolli ad accesso casuale? vantaggi e svantaggi. fai un esempio

I protocolli ad accesso casuale (liv di collegamento) Sono protocolli utilizzati nel problema dell'accesso multiplo ad un collegamento condiviso, quindi quando abbiamo più mittenti e riceventi connessi allo stesso canale. Il nodo trasmette sempre alla massima velocità possibile e non appena si verifica una collisione, i nodi coinvolti ritrasmetteranno il loro frame ripetutamente fin quando non riuscirà ad arrivare a destinazione. La ripetizione però non avviene subito, ma dopo un tempo deciso dai rispettivi protocolli di questo tipo. Es: aloha, slotted aloha, csma e csma/cd .

10) In un frame ethernet c'è il campo dell'indirizzo ip del ricevente? Perchè?

Nel frame ethernet non è presente alcun indirizzo IP in quanto sono presenti gli indirizzi di collegamento, MAC, utilizzati per inviare frame all'interno della stessa sottorete. Gli indirizzi IP infatti appartengono al livello superiore, quello di rete, e servono per identificare un host che si connette alla rete (che può essere internet o no). Al fine dell'invio del frame infatti non è necessario l'utilizzo degli indirizzi IP

11) Accesso al canale condiviso ,problemi e spiegare il protocollo ideale.

Il problema dell'accesso al canale condiviso (liv 4, collegamento) rappresenta il problema di coordinare più nodi mittenti e destinatari aventi un unico canale. Può quindi capitare che più nodi inviano frame contemporaneamente, di conseguenza possono verificarsi collisioni che portano ad una perdita di frame. Un protocollo ideale quindi sarebbe quello in grado di evitare le collisioni ed avere in contemporanea un alta efficienza, quindi avere un alta percentuale di slot vincenti su tutti i nodi attivi (che hanno frame da inviare) .

---**VARIE**-----

1) Illustrare la differenza tra la commutazione di pacchetto e di circuito.

Nelle reti a commutazione di circuito vengono riservate, per tutta la durata della comunicazione, le risorse necessarie perché tale comunicazione avvenga. Questo metodo è utilizzato dalle reti telefoniche,. Tali risorse garantite sono ad esempio la frequenza di trasmissione costante. Non vi è ritardo o perdita di pacchetti e quando i due host desiderano comunicare si stabilisce una connessione end-to-end. Se il circuito però resta inutilizzato abbiamo uno spreco notevole di risorse. Questi circuiti sono implementati in 2 modi: *divisione di frequenza* -> banda di frequenza a ciascuna connessione (non è mai la massima) e *divisione di tempo* -> banda massima per uno slot di tempo.

La commutazione di pacchetto invece prevede un flusso di dati diviso appunto in pacchetti. Le risorse sono condivise e il canale viene sfruttato interamente portando ad eventuali ritardi o fenomeni di congestione. Ogni commutatore di pacchetto è *store and forward* ovvero deve attendere che il pacchetto sia arrivato nella sua interezza prima di poterlo spedire. L'eccessiva congestione può portare anche a perdita di pacchetti che trovano il buffer pieno. Tale tipologia di comunicazione però consente a più utenti di usufruire della rete. Utilizza multiplexing statistico delle risorse, ovvero la loro condivisione.

2)Cos'è il dos (denial of service)? Illustrane la tecnica.

Il denial of service è un attacco mirato a rendere inutilizzabile un servizio, host etc ai legittimi utenti. Rientra generalmente in tre categorie: 1) si sfrutta la vulnerabilità del sistema per mandare messaggi in grado di alterare il funzionamento di tale sistema fino a poter portare ad uno spegnimento dell'host.

3) Principio end to end

Si riferisce al principio che prevede che le operazioni di comunicazione debbano avvenire nei sistemi terminali, quindi agli estremi della rete. Abbiamo quindi sist terminali intelligenti mentre la rete è "stupida". Ma il principio end to end può avere un altro significato, ovvero che , avendo la rete suddivisa per livelli, ognuno di questi non sa assolutamente cosa accade al livello sopra e sotto di lui.

4) Quando un browser richiede una pagina cosa succede? Spiega tutti i protocolli che vengono utilizzati

Client-server instaurano una connessione TCP (quindi si effettua 3way handshake) per far avvenire lo scambio dei dati in seguito ad una richiesta HTTP. Nella richiesta viene indicato il nome dell'host destinazione, e qui entra in gioco il protocollo DNS per poter trovare l'indirizzo IP associato. Il datagramma IP viene inviato fuori dal router tramite interfaccia e passa a livello collegamento. In questo livello viaggia attraverso i collegamenti fisici come frame, utilizzando il protocollo ethernet.

5) Dare la definizione di mezzo di trasmissione e indicare esempi e per ognuno di essi indicare i vantaggi e svantaggi

Mezzo trasmissivo/fisico: Mezzo attraverso il quale si propaga il bit, sotto forma di onda elettromagnetica o impulso ottico, quando viaggia da origine a destinazione, passando per un serie di coppie trasmettitore-ricevitore. Tali mezzi si dividono in: -*Mezzi guidati* (onde contenute in un mezzo fisico) e –*Mezzi ad onda libera* (onde si propagano nello spazio e atmosfera). 1) *Doppino di rame intrecciato*, meno costoso e più utilizzato. Utilizzato per le LAN. 2) *Cavo coassiale*, usato soprattutto nella tv via cavo, raggiunge altre frequenze di trasmissione. 3) *Fibra ottica*, conduce impulsi di luce, ciascuno dei quali rappresenta un bit, immune all'interferenza elettromagnetica e difficile da intercettare. 4) *Canali radio terrestri*, fornisce connettività agli utenti mobili. 5) *Canali radio satellitari*.

6) Cos'è l'imcp e come funziona il programma 'ping'?

L'ICMP è un protocollo di rete utilizzato per scambiarsi, tra host e router, informazioni a livello di rete. L'uso più tipico è quello di notifica degli errori. I messaggi ICMP sono portati all'interno di un datagramma IP,un po' come fossero normali segmenti TCP/UDP. Un mess ICMP contiene un codice, un valore e l'intestazione e i primi 8 byte del datagramma IP che ha causa l'invio del messaggio, questo per identificare il datagramma che ha generato l'errore. Un altro utilizzo di ICMP è per il controllo di congestione. Il programma ping server a mandare un messaggio ICMP di echo (codice 8, valore 0) verso un host destinazione. Questo manderà indietro in automatico la risposta (codice e valore 0). Lo scopo è quello di capire se un host è raggiungibile o meno.

7)Cosa indica il campo TTL e come viene utilizzato in rete?

Time to live è un contatore che mantiene in vita uno specifico dato per assicurare che questo non "giri" per sempre su internet. Per la precisione, troviamo il campo TTL nei record di risorsa DNS, i quali, allo scadere del TTL, verranno rimossi dalla cache DNS. Nel datagramma IP invece, regola la vita di questo. Nei vari casi, il contatore si decrementa ogni volta che il datagramma o il record, passa da un router all'altro.

8) Descrivi i metodi di suddivisione del canale

Abbiamo due tipologie: divisione di frequenza -> ogni utente può sempre usufruire del circuito ma con una frequenza di banda limitata. Divisione di tempo-> la frequenza è massima ma si può sfruttare solo per uno slot di tempo definito, uguale per tutti.

9) Cos'è una cache web e cos'è rfc e indicare due protocolli di due diversi livelli che ne fanno parte

Cache web: (liv applicazione) memoria utilizzata per mantenere dati. Ad esempio nel DNS, la cache mantiene le corrispondenze tra hostname-ip, in modo da non dover sempre interpellare il server radice. E' utilizzata anche per il protocollo http,, mantenendo in memoria oggetti di pagine visitate recentemente. *L'RFC (request for comments)*: è un documento che riporta informazioni o specifiche riguardanti protocolli.

10)Cos'è il principio store and forward?

E' la necessità di ricevere l'intero pacchetto prima di poterlo inviare in uscita. Questa si presenta nelle commutazioni di pacchetto dove si cerca di massimizzare l'utilizzo della rete.

11) Cos'è la modalità stop and wait

12) Livello di trasporto stop and wait

Nel protocollo stop and wait, il mittente dopo aver inviato il pacchetto, prima di poter proseguire a fare qualsiasi azione, deve aspettare un riscontro dal destinatario (riscontro che può essere sia

negativo NAK che positivo ACK)

13) indicare almeno due "valori" del campo metodo della richiesta http e spiegare cosa risponde il server

Get: per ottenere una pagina specifica (indicata tramite url) e tutti i suoi oggetti referenziati.
Delete: cancella nei server web un oggetto
Il server invia un messaggio di risposta contenente come riga di stato un "avviso" su come si è conclusa la richiesta. Ad esempio "200 ok" che significa che la richiesta è andata a buon fine e viene allegato l'oggetto richiesto. Il mess di risposta include anche, nelle righe di intestazione, se la connessione è persistente o meno, la data e l'ora dell'invio della risposta, il tipo di contenuto etc.

14) Definire il throughput. Cosa si intende per throughput medio, istantaneo e collo di bottiglia?

Il throughtput è la frequenza di trasmissione dei bit tra mittente e destinatario. Si dice istantaneo quando si misura in un determinato istante, mentre si dice medio quando lo si misura in un lasso di tempo più lungo. Il collo di bottiglia invece è il collegamento sul percorso punto-punto che limita il throughtput end to end.

15) Il livello di rete instaura la connessione? motivare risposta

16) A livello di rete è prevista una fase di installazione della connessione?

A livello di rete nell'architettura di internet, non è presente alcun tipo di connessione in quanto la rete è a datagramma (quindi una rete per definizione senza connessione). Le architetture di rete come ATM invece usano le reti a circuito virtuale quindi delle connessioni tra i router. L'unica connessione in internet che abbiamo è la livello di trasporto.

17) parla del protocollo http con particolare riferimento al protocollo di trasporto utilizzato

Questo protocollo di livello applicazione, definisce come il client web richiede le pagine al server web, e come questo le trasmetta al client. Lo scambio dei dati avviene con l'utilizzo del prot di trasporto TCP. Questo assicura infatti che tutto ciò che è stato richiesto venga recapitato al client, evitando perdite di dati. Si instaura quindi una connessione client-server, con l'utilizzo delle socket, per comunicare con TCP. Possiamo avere connessioni persistenti, che rimangono aperte per l'invio di tutti i dati e richieste tra client-server. In caso invece di connessioni non persistenti, ognuna di queste, dopo l'invio dell'oggetto, verrà chiusa. Per comodità, si aprono diverse connessioni TCP in parallelo in modo da ricevere più oggetti in contemporanea.

18) il get condizionale, come funziona?

Il get condizionale è un metodo che permette alla cache di verificare che un suo oggetto sia aggiornato o meno. Introduce nella richiesta http la riga if-modified-since. La cache chiede al server se l'oggetto è aggiornato, in questo caso gli verrà recapitato, se non lo è, nella risposta avremo "304 NOT MODIFIED".

19) quali sono i tipi di malware e come funzionano.

Il malware è creato con l'intento di attaccare un sistema informatico. È autoreplicante e può arruolare gli host infetti in botnet. Si dividono in:
-*Cavallo di troia:* è la parte nascosta (infetta) di un software
-*Worm:* sfrutta software già presente sul computer e si auto-esegue
-*Virus:* infezione proveniente da un software che richiede l'interazione con l'utente.

20) Cosa si intende con ack cumulativo? Tutti i protocolli di trasporto ne fanno uso?

Con ACK cumulativo si intende il riscontro di più segmenti TCP in un'unica volta. Ovvero, invece di mandare un ack per ogni segmento riscontrato, si manda un unico ACK per tutti i segmenti riscontrati fino a quel momento. Preso quindi d'esempio il numero di sequenza N, l'ack cumulativo riscontrerà tutti i segmenti con numero di sequenza >= N che sono stati ricevuti correttamente. Viene utilizzato esclusivamente da TCP, e nello specifico è la caratteristica della tipologia di protocolli Go-Back-N (Di cui TCP acquisisce alcune caratteristiche)

21) Quando un router riceve un datagramma IP che operazioni può svolgere? Può effettuare delle modifiche?

Quando un router riceve un datagramma IP, deve stabilire a quale interfaccia d'uscita mandarlo tramite le tabelle di inoltro (presenti nel processore d'instradamento e in certi casi anche nelle porte d'ingresso) che utilizzano la corrispondenza IP-interfaccia. Il router può fare un controllo degli errori (il checksum IP è solo a livello intestazione) e scartare il pacchetto se necessario. Più importante modifica è quella in caso di router NAT. In questo caso i router si comportano come un unico dispositivo con un unico indirizzo IP. Quando uno di questi router fa ad esempio una richiesta http, il router NAT della sottorete, riceve il datagramma e sostituisce l'indirizzo IP di origine col proprio. Quando il datagramma con la pagina richiesta arriva al router NAT, verrà modificato nuovamente il campo IP con quello dell'host che ha effettuato la richiesta.

22) Descrivere brevemente il protocollo Http e descrivere dettagliatamente le connessioni persistenti e non persistenti

Il protocollo applicazione http, definisce come il client web richiede le pagine al server web e

come quest'ultimo le trasmetta al client. E' utilizzato il protocollo di trasporto affidabile TCP in modo che tutto ciò che è stato richiesto venga recapitato. Le connessioni TCP possono essere di due tipologie (la natura della connessione è specificato nel messaggio di richiesta/risposta http dalla voce Connection:). 1) Non persistenti. Queste connessioni si mantengono aperte solo per la richiesta/spedizione di un singolo oggetto, ma non per l'intera durata della sessione. Si possono aprire più connessioni non persistenti in parallelo, in modo da richiedere/ricevere più oggetti in contemporanea. 2) Persistenti. Rimangono aperte durante l'intera sessione e la richiesta e l'invio di tutti gli oggetti avviene nella stessa connessione. E' possibile anche l'invio di più pagine web.

23) Pipeling nell'ambito trasferimento dati, fai un esempio riferendoti ad un Protocollo di trasporto

24) Cos'è un sistema terminale?? E fornire alcuni esempi

25) Spiega dettagliamente tutte le operazioni che avvengono quando arriva un datagramma IP su una porta interna

26) Descrivi brevemente 3 protocolli di livello applicazione

27) Descrivi quale relazione si interpone tra l'MTU(Maximun Transport Unit) e la "sliding window"

--Domande in preparazione d'esame FATTE A LEZIONE

1) Data una sessione di comunicazione tra due processi, com'è possibile individuare qual è il server e quale è il client? (Paragrafo 2.1.2) - FACILE

Il server è colui che fornisce il servizio e attende, il client è colui che inizia la connessione (comunicazione nel caso di udp). Detta in modo più brutale a livello di trasporto il client è colui che invia il primo segmento con SYN – ACK

2) Cosa si intende per trasferimento di dati affidabile? (Paragrafo 2.1.3 pg 81) - FACILE

E' un trasferimento che assicura che i segmenti arrivino in ordine, non devono perdersi i pacchetti e non devono esserci alterazioni all'interno del payload, il destinatario deve essere sempre in grado di riportare nell'ordine corretto (TCP). TCP offre controllo di flusso (non bisogna sovraccaricare il destinatario) e controllo di congestione (non trafficare la rete). TCP è un esempio, è un implementazione di protocollo che assicura affidabilità; esistono altri protocolli che garantiscono affidabilità.

3) Quali sono i servizi di trasporto offerti da Internet? Descriverli brevemente.

Con connessione (TCP), senza connessione (UDP) e spiegarli..

4) Nel caso di internet, a livello di rete è presente una fase i instaurazione della connessione? Motivare la risposta 4.1.1 pg 281 (FACILE)

A livello di rete non c'è nessuna instaurazione della connessione. L'unica instaurazione è a livello di trasporto, perché internet è una rete a commutazione di pacchetto e non a commutazione di circuito.

5)Qual è la differenza tra inoltro (forwarding) e instradamento (routing)? FACILE 4.1.1 pg 278

Inoltro è la scelta locale del datagramma che arriva da una certa porta e deve capire su quale porta deve uscire(problema locale all'interno del singolo apparato di rete) - instradamento qual è il percorso tra l'origine e il destinatario, quindi avrò un insieme di algoritmi; in ogni singolo router attraverso le tabelle di inoltro deve decidere attraverso quale percorso deve passare il pacchetto.

6) Indicare in modo preciso ed accurato le differenze tra "datagramma", "messaggio", "frame" e segmento" FACILE 1.5.2 pg 48

particolarità: Segmento è a livello trasporto noi abbiamo utilizzato la terminologia per segmento udp e
segmento tcp per semplificare ma in realtà negli RFC i segmenti sono solo quelli tcp mentre quelli udp si
chiamano datagrammi utente.

7) Per un eventuale attaccante è più facile inserirsi in una comunicazione TCP oppure UDP? Motivare

adeguatamente e dettagliatamente la propria risposta FACILE

Risposte dagli alunni:
-E' più facile tcp perché guardando un flusso di comunicazione riesco a rimettere in ordine i vari messaggi
-E' udp perché nel caso tcp bisognerebbe incastrarsi bene nella finestra di comunicazione
-La connessione udp è più facile perché con tcp crei un tubo tra te e chi comunica invece bisognerebbe
inserirsi nella comunicazione e sostituirsi a un comunicante
-Tcp è più difficile perché l'attaccante dovrebbe indovinare qual è il numero di sequenza successivo
-Con wireshark abbiamo visto che si può ricostruire la comunicazione in sequenza
-Con TCP si sceglie a caso il primo e poi è difficile andare a vedere i seguenti pacchetti

Prof. E' più facile udp. Nel caso di TCP c'è un three way handshake che si scambiano numeri di sequenza
scelti casualmente che è difficile da andare a ritrovare tutto il percorso. Nel caso di UDP non esiste il
numero di sequenza quindi basta sapere chi è il destinatario (il numero di porta del dest) con cui sta
comunicando. In tcp invece c'è una fase di sincronizzazione (SYN) in protocolli che devono garantire
sicurezza (https) viene implementato col protocollo tcp quindi è molto più facile inserirsi in DNS (forgiare
la risposta finta e la domanda finta) che ritrovare la richiesta di una pagina web http. In che modo http mi
protegge dalle risposte finte? Cosa c'è nell'intestazione delle richieste DNS? (Questo lo lascia a noi) http
inserisce un campo per proteggermi da questi casi, il campo di identificazione che genera un campo da 16
bit che coincidono nella domanda e nella risposta.

8) Descrivere brevemente il protocollo http, motivando per quale motivo viene definito un protocollo "senza stato" MEDIA, perché c'è da discutere paragrafo 2.2.1 (pag 89)

Raccontare com'è fatto http e motivare perché è stateless.

9) Il protocol FTP può essere definito "con" o "senza stato" e motivare la risposta - paragrafo 2.3 pg 104

Con stato perché deve sapere dove si trova all'interno del file system del server e ogni volta che cambio directory nel server il server deve sempre ricordarsi dove sono io. Un effetto collaterale di avere una connessione con stato è quello di avere una connessione persistente.

10) Descrivere l'utilizzo fatto dal programma traceroute del protocollo ICMP - paragrafo 4.4.3 pg 318

ICMP protocollo di segnalazione degli errori. Traceroute con ICMP vengono inviati 3 segmenti udp verso un certo server di cui io dirò l'indirizzo ip verso un numero di porta considerato improbabile, improbabile significa una porta su cui, è un assunzione che sta facendo un client, è improbabile che c'è un server in ascolto (una socket in ascolto), poi a livello di datagramma spedisce il primo datagramma con TTL pari a uno, il router decrementa il TTL, … continua a spedire 2 datagrammi con ttl = 2 poi con ttl = 3 e di conseguenza il mittente riesce a capire che (sceglie tra le porte alte più che altro)

11)Un classico - Nell'ambito del protocollo TCP a che cosa ci si riferisce parlando di " partenza lenta"? paragrafo 3.7 pg 252 (MEDIA)

Che cos'è e quando avviene, bisogna conoscere bene il controllo di congestione TCP. Incremento esponenziale che si dimezza quando c'è congestione.

12)Descrivere i cinque livelli della pila protocollare tipica di Internet, avendo cura di indicare dove possibile almeno un protocollo d'esempio per ogni livello. Discutere inoltre le differenze rispetto al modello di riferimento OSI a sette livelli.(MEDIA)

Livello di sessione quello che noi abbiamo è all'interno dell'idea iso osi il client e il server dovrebbero essere in grado di avere una connessione che quando è necessario non è semplicemente stateless quindi lo state stesso è in grado di offrire un senso di stato alle

applicazione che hanno bisogno. Nel livello di presentazione la cosa più facile da ricordare è come i dati vengono codificati (esempio little endian big endian qual è che è più importante? A seconda delle implementazioni) Quando due dispositivi in rete
parlano tra di loro e hanno architettura diversa, come fanno a mettersi d'accordo??? Attualmente non c'è –
le applicazioni dicono questa è la mia codifica ed eventualmente effettuano una traduzione.

13) Descrivere il problema del blocco in testa alla fila (HOL, head-of-the-line blocking) nell'ambito dei router – par 4.3.4 pg 299 DIFFICILE

E' un problema della switching fabric, ci sarebbe una interfaccia libera quindi posso spostare il pacchetto come voglio ma il pacchetto è bloccato da quello davanti a lui che anch'esso è bloccato dalla trasmissione di un altro pacchetto ancora. Quando c'è un rate di arrivo preponderante rispetto alle altre interfacce quindi ci sono alcuni pacchetti che non hanno un canale condiviso ma un canale unico.

14) Che cos'è e a cosa serve la tecnica del tunneling nell'ambito di IPv6 (DIFFICILE) 4.4.4 pg 324 DIFFICILE

Descrivere cos'è ipv6, perché è difficile il passare da 4 a 6 e spiegare il tunneling

15)Qual è la differenza tra MSS (Maximum Segment Size) e MTU (Maximum Transmission Unti)? 3.5.1 pg 217 DIFFICILE

Distinguere di quali livelli della pila protocollare stiamo parlando MTU – collegamento livello 2 – MSS
– Trasporto livello 4 riguarda sia tcp che udp
Partiamo dall'inizio
MTU = dimensione massima del frame di livello collegamento di conseguenza il contenuto massimo del
datagramma a livello di rete sarà la dimensione massima del frame -. Intestazione e crc e vado a calcolare
quanto può essere grande un datagramma senza che devo frammentarlo (spezzarlo)
MSS = dimensione massima del segmento affinché che il livello di rete non debba frammentare.
Perché c'è frammentazione su internet ? Perché ci sono differenti tecnologie.
Perché TCP si e UDP no ?? Nel caso di UDP no perché il datagramma può essere lungo solo 2^16-1 byte e
se è troppo grande quello che accadrà è che verrà incapsulato in un datagramma che verrà inviato. UDP fa
una verifica in meno in questo caso.