Domande sulla quarta parte dei lucidi

1) Il livello di trasporto è implementato solo in alcuni nodi di rete. Quali?

Il mittente e la destinazione.

2) In cosa consiste la funzione di multiplexing offerta dal livello di trasporto al livello applicativo?

Nel fatto che il livello di trasporto può essere utilizzato da piu protocolli applicativi contemporaneamente: infatti il livello di trasporto utilizza delle porte per distinguere i protocolli che ne fanno uso e cui deve passare i pacchetti.In particolare la coppia indirizzoip-porta costituisce un socket.

3) A cosa servono i numeri di porta?

Vedi domanda precedente.

4)Perché alcuni numeri di porta sono considerati numeri noti? A cosa servono?

Sono considerati numeri noti perché sono numeri assegnati in modo fisso dal sistema operativo.L'utilità è quella della domanda 2.

5) Cos'è il sockete come si identifica?

Il socket si identifica dalla coppia dell'indirizzo ip e numero porta. Esso porta così a distinguere sul server due connessioni che fanno uso della stessa porta.

6) Che funzioni implementa e che tipo di servizio offre il protocollo UDP?

Il protocollo udp è un protocollo senza connessione che non garantisce il corretto smistamento dei pacchetti.Esso implementa un blando controllo sui pacchetti che gli vengono consegnati.

7) Che cos'è lo pseudo-header?

E' un certo numero di bit che viene aggiunto all'header del pacchetto udp in modo che il protocollo udp possa eseguire un controllo di correttezza calcolando il checksum.Esso è costituito dall'indirizzo ip del mittente, dall'indirizzo ip del nodo di destinazione, da una serie di zeri riservati, dal campo protocol (lo stesso di ip) e dalla lunghezza dell'header udp.

8) Che tipo di servizio offre il TCP ai livelli superiori? Quali meccanismi di controllo adotta?

Il protocollo TCP offre ai livelli superiori la correttezza dei dati in pacchetti con corretta sequenza.Assicura quindi un trasporto affidabile.Si basa in particolare sul controllo di flusso, sul controllo di congestione e sul controllo d'errore per raggiungere questi obiettivi.

9) Cosa vuol dire che il TCP è connection-oriented?

Vuol dire che prima del trasferimento effettivo dei dati, avviene una fase di connessione tra i nodi estremi, ossia tra il mittente e il destinatario.

10) Che differenza c'è tra controllo di flusso e controllo di congestione?

Il controllo di flusso si basa sull'utilizzo di una sliding windowal fine di limitre il ritmo di trasmissione sulla capacità del ricevitore di ricevere i pacchetti. Invece il controllo di congestione è volto a limitare il ritmo di trasmissione a causa di eventi di congestione di pacchetti che si possono trovare sulla rete.Sfortunatamente sulla rete non esiste nessun tipo di notifica di questo tipo, quindi il tcp è costretto a effettuare questo controllo sulla base del numero di perdite di pacchetti.

11) Cosa indica il campo SN (Sequence Number) nell'header TCP?

Dato che il tcp numero tutti i byte che vengono trasmessi, nel sequenze number è riportato il numero del primo byte contenuto nel campo dati.

12) Cosa indica il campo AN (Acknowledgment Number) nell'headerdel TCP?

Indica il numero di byte dal quale il ricevente si aspetta di ricevere i dati.

13) Cosa contiene il campo AN quando il flag ACK vale zero?

Può contenere un qualsiasi valore, comunque il campo non viene considerato valido.

14) Cosa contiene il campo Window dell'header del TCP?

Contiene il valore della finestra di ricezione così come specificato dall'ip receiver.
15) Cosa contiene il campo SN quando il flagSYN vale uno?

Durante le prime 2 fasi del three way handshake, il mittente riempie il sq con un numero casuale per mettersi d'accordo sul numero del primo byte con il ricevente.Nella seconda fase (in cui il ricevente approva il numero di squenza) il sq viene incrementato di uno.

16) A cosa servono il flagURG e il campo UrgentPointer?

Il falg URG segnala se vi sono dati urgenti.Se posto a 1 il campo Urget Pointer punta al primo byte dei dati urgenti.

19) A cosa serve l'opzione MSS dell'header quando viene usata?

Serve per comunicare in fase di connessione il Maximum Segment Size che di default vale 536 byte.

20)In cosa consistono i problemi della "Silly Window Syndrome" lato ricevitore e lato trasmettitore? Come vengono risolti?

Esistono due tipologie di Silly Window Sindrome:dal lato del ricevitore e quella dal lato del trasmettitore. La Silly Window Sindrome dal lato del ricevitore prevede che lapplicazione svuoti molto lentamente la finestra di ricezione:in questo modo invia segmenti con finestra piccola al trasmettitore e il trasmettitore invia segmenti molto piccoli con tanto overhead sprecando quindi risorse di rete.La soluzione quindi prevede che il ricevitore menta al trasmettitore dicendo che ha una finestra nulla fino a quando la finestra non arriva al valore minimo tra la metà del buffer di ricezione e MSS.

La silly window Sindrome dal lato del trasmettitore invece prevede che vengano generati molto lentamente i pacchetti dall'applicazione.Il tcp quindi invierebbe segmenti piccoli consumando risorse di rete.Una soluzione possibile quindi è quella di aspettare che il buffer di uscita si riempia fino ad un valore pari almeno a MSS prima di trasmettere oppure la trasmissione avviene quando si riceve un acknowledgement per il segmento preedente.

21) A cosa serve il meccanismo di PUSH dei dati?

Serve per "intimare" al tcp di inviare i dati presenti nel buffer di uscita. L'inoltro dei dati avviene quindi immediatamente.Inoltre il ricevente viene configurato allo stesso modo ponendo a 1 il flag push. Questo tipo di funzionamento viene utilizzato in applicazioni come telnet.

22) Come vengono gestiti i dati URGENT?

Come alla domanda 16: in particolare non sno soggetti al normale controllo di flusso.

23) Perché il TCP ha la necessità di stimare il valore più appropriato per il TIMEOUT di ritrasmissione?

Perché se fosse troppo piccolo si avrebbero continue ritrasmissioni non necessarie e se fose troppo alto, passerebbero quantità di tempo troppo alte per rilevare la perdita dei pacchetti.

24) Come opera l'algoritmo che modifica dinamicamente il TIMEOUT del TCP?

Pesante questa!!!Sono formulacce : scusate ma non ho la pazienza di scriverle!!!!!

25) Cos'è il timer di persistenza?

Quando il ricevitore segnala al trasmettitore di avere una finestra nulla la trasmissione cessa.Riprende solo quando il ricevitore manda un segnale di ack al mittente. In questo caso se viene perso questp segnale di ack sono guai e la connessione rimarrebbe bloccata. Perciò si usa un timer di persistenza quando arriva un messaggio che indica finestra nulla. Allo scadere di questo timer si manda un messaggio di sonda probe e se viene ricevuto un ack si esce dallo stato critico altrimenti riprende a conteggiare il timer di persistenza.

26) Che cos'è la RCVWND? Quando viene modificata?

La finestra di ricezione o RCVWND è lo spazio di dati del biffer di ricezione disponibile a ricevere nuovi dati. Essa viene modificata quando arrivano nuovi dati o quando l'applicazione svuota il buffer di ricezione dai dati che sono arrivati.

27) Che cos'è la CWND?

E' la finestra di congestione che tiene il trasmettitore. Essa è volta a limitare il ritmo di trasmissione in seguito al riscontro di eventi come la perdita di pacchetti che sono interpretati dal tcp come eventi di congestione sulla rete. Il trasmettitore non può trasmettere piu dati del minimo tra RCVWND e CWND.

28) Qual è la finestra che limita effettivamente la velocità di trasmissione?

Solitamente la CWND perché è piu piccola.Dico solitamente!!!!!

29)Descrivere l'algoritmo di gestione della CWND.
Come si distingue tra le fasi di Slow Start e Congestion Avoidance?
Come viene variata la CWND in Slow Start?
Come viene variata la CWND in Congestion Avoidance?

Come viene fissata la CWND e la SSTHRESH allo scadere di un timeout?

a) Le fasi di Slow Start e di congestion avoidance vengono distinte dal valore della variabile SSTHRESH.In particolare se SSTHRESH > CWND allora si è in fasse di slow start altrimenti ci si trova in fase di congestion avoidance.

b)In slow start la variabile SSTRESH viene impostata ad un valore molto alto dal trasmettitore mentre la CWNd viene posta a 1 MSS. In quiesto modo si parte sempre in slow start. Successivamente la CWND viene incrementata di 1 fattore per ogni ack ricevuto. Ricordiamo che un fattore pari a 1 significa che la finestra viene aumentata di un fattore pari a 2 .

c)In questa fase, ossia quando la variabile SSTHRESH < CWND si è in fase di congestion avoidance.La CWND in particolare viene incrementata di 1/CWND ad ogni ack ricevuto.

d)Alloo scadere di un timeout (che viene interpretato come un evento di congestione) la CWND viene posta a 1 mentre la SSTHRESH viene posta al massimo tra il valore di 2MSS e la FlightSize/2. Ricordiamo che la flight size contiene il numero di byute trasmessi che non hanno avuto un riscontro.

30)A cosa serve e come funzionano i meccanismi di Fast Retransmit e Fast recovery?

Questi due meccanismi servono per recuperare velocemente le perdite di pacchetti che possono avvenire sulla rete.Essi funzionano nel seguente modo:

- Al terzo ack consecutivo duplicato con lo stesso An si pone SSTHRESH uguale al massimo tra la Filght Size/2 e 2MSS.
- Viene ritrasmesso il pacchetto indicato dall'AN
- Si pone la CWND uguale alla somma di SSTHRESH e 3MSS.
- Per ogni ulteriore Ack ricevuto la CWND viene incrementata di 1.
- Vengono trasmessi nuovi segmanti se consentito dai valori delle finestre CWND e RWND.
- Appena arriva un ack che riscontra nuovi dati si esce dalla fase di fast recovery e si pone di nuovo CWND ugale al massimo tra FlightSize/2 e 2MSS.

Ciò che si vuole fare è quindi aumentare la CWND perché un ack duplicato dimostra che i pacchetti inviati successivamente a quello richiesto sono arrivati correttamente e quindi non si hanno eventi di congestione per i pacchetti successivi.sfortunatamente non funziona per perdite multiple di pacchetti.