

Esame di **Progettazione di servizi web e reti di calcolatori (01NBE)**

Corso di Laurea in Ing. Gestionale e dell'Organizzazione d'Impresa

Prova scritta di teoria (26/6/2014)

NOTA

Le tracce delle soluzioni fornite in questo testo sono da considerarsi solo come un aiuto per comprendere i principali punti da toccare nel risolvere gli esercizi proposti ma non sono né esaustive né presentate in forma adeguata per l'elaborato da consegnarsi in sede d'esame.

In particolare per molti esercizi la soluzione è volutamente schematica e ci si attende che il candidato spieghi adeguatamente i singoli punti, per dimostrare reale comprensione dell'argomento invece che semplice capacità mnemonica di ricordare i punti elencati nelle slide (o in queste tracce di soluzione).

Esercizio 1 (punti: 5)

Spiegare cosa è il protocollo POP ed illustrarne il funzionamento con un esempio.

Traccia di una possibile risposta

POP è un protocollo client-server usato in modalità Pull da un MUA per gestire una casella di posta su un MS. Usa comandi ASCII ma è 8-bit clean” per poter trasportare anche dati binari (nel body di un messaggio di posta).

La versione oggi più usata è POP-3, che autentica l'utente verso il MS (tramite i comandi USER e PASS) e quindi permette di elencare, scaricare ed eventualmente cancellare i messaggi presenti nella casella di posta (tramite i comandi LIST, RETR, DELE).

Un esempio di interazione ...

Esercizio 2 (punti: 3)

Spiegare il concetto di *multiplexing* in una comunicazione di rete ed indicare in quale modo viene implementato in TCP e UDP.

Traccia di una possibile risposta

Multiplexing è l'abilità di usare una risorsa in modo condiviso da parte di più entità.

Nel caso di TCP e UDP la risorsa condivisa è la trasmissione di pacchetti IP verso un nodo di rete. Poiché il nodo può ospitare più processi che necessitano di comunicare in rete, il flusso destinato a ciascun processo viene distinto in base alla quintupla (IP.src, IP.dst, L4.proto, port.src, port.dest) che identifica in modo univoco una comunicazione a livello trasporto in una rete TCP/IP.

Quindi il concetto di multiplexing in TCP e UDP è associato al concetto di porta.

Esercizio 3 (punti: 3)

Spiegare che cosa è il *selective acknowledgment* in TCP, quale benefici apporta ed in quale modo.

Traccia di una possibile risposta

Il Selective Acknowledgment (SACK) è una opzione di TCP utile per ridurre le ritrasmissioni e quindi la congestione di rete.

Con l'Acknowledgment semplice, se viene perso un segmento, viene chiesta la ritrasmissione non solo del segmento perso ma anche di tutti quelli successivi (nonostante siano stati già ricevuti). Invece con SACK vengono indicati i segmenti successivi già ricevuti; in questo modo vengono ritrasmessi solo i segmenti persi.

Esercizio 4 (punti: 5)

Spiegare quali sono le regole di buona progettazione di una pagina web con riferimento all'uso dei font e dei colori.

Traccia di una possibile risposta

Distinguiamo il caso di lettura a video (V) o stampa (S) di una pagina web.

Usare font senza grazia (V) o con grazia (S) ...

Esprimere la dimensione del font come percentuale del default dell'utente (V) o come valore assoluto 10-12 pt (S) ...

Non usare troppi font diversi nella stessa pagina e soprattutto non cambiarli all'interno di uno stesso paragrafo.

Usare per sfondo e testo colori con alto contrasto (posizioni opposte sul diagramma del parametro Hue): nero su bianco (S) o bianco su nero (V) anche se spesso si usa nero su bianco anche per V perché più gradevole.

Non usare troppe sfumature di colore (potrebbero non essere disponibili) e cercare di limitarsi ai colori base disponibili su qualunque browser ...

Esercizio 5 (punti: 5)

Un ufficio è dotato di uno switch Ethernet a 1 Gbit/s, al quale sono collegati 20 PC “client” (con scheda di rete a 100 Mbit/s) ed un PC “server” (con scheda di rete a 1 Gbit/s). Ogni mattina, alle 8:30, tutti i client vengono accesi e devono scaricare gli aggiornamenti dal server. L'aggiornamento quotidiano è composto da 10 file di 15 MB ciascuno. Sapendo che il server è di tipo concorrente, dotato di 4 CPU, con disco SATA da 1 TB 3 Gbit/s ed il tempo di attivazione di un processo è di 10 ms, calcolare il tempo minimo necessario affinché tutti i client siano operativi dopo aver scaricato gli aggiornamenti.

Traccia di una possibile risposta

La banda aggregata di tutti i client operanti simultaneamente è:

$$20 \cdot 100 \text{ Mbps} = 2 \text{ Gbps}$$

quindi il collo di bottiglia è lo switch e la scheda di rete del server, visto che entrambi operano a 1 Gbps.

Poiché non ci sono calcoli da effettuare e la velocità di lettura da disco è superiore a quella di rete del server, non occorre considerare il tempo di lettura da disco (che avviene in parallelo all'invio in rete) e neanche il tempo di attivazione dei processi (trascurabile rispetto agli altri tempi). L'unico elemento da considerare è il trasferimento in rete:

$$20 \text{ client} \cdot 10 \text{ file/client} \cdot 15 \text{ MB/file} = 3000 \text{ MB} = 24000 \text{ Mbit}$$

Quindi il tempo minimo necessario ad effettuare tutti i download è:

$$24000 \text{ Mbit} / 1024 \text{ Mbps} = 23.4 \text{ s}$$

Risultato analogo si ottiene se si divide la banda disponibile per i client attivi simultaneamente:

$$1024 \text{ Mbps} / 20 \text{ client} = 51.2 \text{ Mbps (per ciascun client)}$$

e quindi calcolando il tempo per effettuare un singolo download (visto che tutti i client operano in parallelo):

$$10 \text{ file} \cdot 15 \text{ MB/file} \cdot 8 \text{ bit/B} / 51.2 \text{ Mbps} = 23.4 \text{ s}$$

Esercizio 6 (punti: 5)

In HTTP/1.1 è stata introdotta la codifica *chunked*. Spiegare di cosa si tratta, come funziona e quale problema di HTTP/1.0 risolve.

Traccia di una possibile risposta

La codifica chunked serve a trasferire dati di cui non è nota a priori la quantità (caso tipico: pagine web create dinamicamente sul server).

La codifica viene indicata nell'header HTTP Transfer-encoding e realizzata nel body inviando i dati in tanti pezzi (chunk) indicando per ciascuno di essi la dimensione in byte. La fine della trasmissione dei dati è indicata da un chunk con dimensione zero.

Questa codifica risolve il problema del possibile troncamento dei dati in HTTP/1.0: se veniva trasmessa una pagina dinamica non se ne poteva indicare la dimensione (tramite l'header Content-length) perché non nota e quindi il client assumeva che i dati fossero finiti quando veniva chiuso il canale TCP. Ma se ciò avveniva non per volontà del server ma per un problema di rete, ecco che il client riceveva solo una parte dei dati ma non aveva modo di saperlo.

Esercizio 7 (punti: 6)

Descrivere le soluzioni tecniche suggerite per proteggere un server HTTP che deve offrire il proprio servizio verso Internet, tenendo conto che il server è anche collegato ad un database di back-end tramite una rete locale.

Traccia di una possibile risposta

Posizionare il server sulla DMZ di un firewall che applichi al traffico in ingresso non solo controlli di tipo packet-filter (per limitare le porte e/o protocolli accessibili) ma anche applicativi (es. tramite un reverse proxy).

Collegare il server al DB tramite un canale sicuro (es. IPsec).

Usare TLS per i collegamenti dei client al server, se possibile attivando la TLS client authentication o almeno una HTTP authentication (basic o meglio digest), così da ridurre la superficie d'attacco.