Esame di **Progettazione di servizi web e reti di calcolatori (01NBE)**
Corsi di Laurea in Ing. Gestionale e dell'Organizzazione d'Impresa
Prova scritta di teoria (2/7/2015)

> 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 le 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 che cosa identificano i record DNS di tipo SOA e NS.

Traccia di una possibile risposta
Record SOA (Start-Of-Authority) = informazioni sul primary nameserver di un dominio (es. suo indirizzo IP, indirizzo mail del suo amministratore, varie scadenze relative ai record forniti)
Record NS (NameServer) = indica il nome di un nameserver per un certo dominio, tipicamente ci sono tantiu record NS per un dominio (corrispodenti al primary NS ed ai vari secondary NS).

Esercizio 2 ***(punti: 5)***
Spiegare (anche con l'aiuto di un esempio) il funzionamento del protocollo SMTP per la spedizione di un messaggio RFC-822 tra un MUA ed un MSA.

Traccia di una possibile risposta
Il client (MUA) apre un canale TCP (porta 25 o 587) ed avvia un dialogo che mira a:

1. *identificare il nodo client*
2. *identificare il mittente RFC-822*
3. *identificare i destinatari RFC-822*
4. *trasmettere il messaggio RFC-822*
5. *chiudere il canale TCP*

Ad esempio:

```
(S) 220 mail.x.com
(C) HELO mynode.isp.it
(S) 250 Hello mynode.isp.it
(C) MAIL FROM: francesco@isp.it
(S) 250 francesco@isp.it ...  Sender ok
(C) RCPT TO: giuseppe@x.com
(S) 250 giuseppe@x.com ...  Recipient ok
(C) DATA
(S) 354 Enter mail, end with "." on a line by itself
From:  francesco@isp.it
To:  giuseppe@x.com
Subject:  incontro

Ci vediamo domani alle 11:30.
Frank
.
```

```
(S) 250 Ok
(C) QUIT
(S) 221 mail.x.com closing connection
```

Esercizio 3 ***(punti: 5)***
Spiegare ed illustrare cosa avviene a livello di protocollo HTTP/1.1 tra il browser ed il server quando un utente visita la seguente pagina HTML (presente alla URL `http://www.alpha.com`) e preme il pulsante Invia dopo aver introdotto come nome `Amedeo De Capitani` e come anni `34`.

```
...
<form method="POST" action="/enrol.php">
Nome? <input type=text id="nome"><br>
Anni? <input type=text id="anni"><br>
<input type="submit" value="Invia">
</form>
...
```

Traccia di una possibile risposta
Il browser apre un canale 80/tcp col server e trasmette le seguenti informazioni:

```
POST /enrol.php HTTP/1.1
Host:  www.alpha.com
Content-Type:  application/x-www-form-urlencoded
Content-Length:  31

nome=Amedeo+De+Capitani&anni=34
```

Esercizio 4 ***(punti: 5)***
Un laboratorio offre un collegamento ad Internet tramite un access point wireless a 54 Mbps, collegato ad un backbone di rete da 100 Mbps. Nel laboratorio sono presenti 12 studenti dotati tutti di un notebook con scheda wireless a 11 Mbps.
Sette studenti devono scaricare da un server Internet un archivio RAR da 100 MB per svolgere l'esercitazione mentre quattro studenti devono scaricare un archivio da 200 MB.
Calcolare il tempo minimo necessario affinché tutti gli studenti siano pronti a svolgere l'esercitazione nell'ipotesi che tutti gli studenti operino simultaneamente.

Traccia di una possibile risposta
Si noti che uno studente è inattivo. I restanti 11 studenti iniziano simultaneamente a lavorare ed occorre quinidi calcolare il collo di bottiglia tra loro ed il server:

$$\min(11 \cdot 11\,\text{Mbps}, 54\,\text{Mbps}, 100\,\text{Mbps}) = \min(121\,\text{Mbps}, 54\,\text{Mbps}, 100\,\text{Mbps}) = 54\,\text{Mbps}$$

L'access point risulta quindi il collo di bottiglia che determina la velocità per scaricare 100 MB *da parte di 11 studenti:*

$$T_{11} = \frac{11 \cdot 100\,\text{MB} \cdot 8\,\text{bit/byte}}{54\,\text{Mbps}} = 162.9\,\text{s}$$

A questo punto 7 studenti hanno terminato il loro lavoro e restano solo 4 studenti che devono scaricare altri 100 MB. *Il collo di bottiglia deve quindi essere ri-calcolato come segue:*

$$\min(4 \cdot 11\,\text{Mbps}, 54\,\text{Mbps}, 100\,\text{Mbps}) = \min(44\,\text{Mbps}, 54\,\text{Mbps}, 100\,\text{Mbps}) = 44\,\text{Mbps}$$

Questa volta il collo di bottiglia è creato dalle schede wireless dei latop e quindi il tempo per completare il lavoro da parte dei 4 studenti residui è:

$$T_4 = \frac{4 \cdot 100\,\text{MB} \cdot 8\,\text{bit/byte}}{44\,\text{Mbps}} = 72.7\,\text{s}$$

Il tempo minimo necessario affinché tutti gli studenti siano pronti a svolgere l'esercitazione è:

$$T = T_{11} + T_4 = 162.9 + 72.7 = 235.6\,\text{s}$$

Esercizio 5 ***(punti: 6)***
Identificare quali sono i tre componenti principali di una generica applicazione, illustrarne la funzionalità e dire come potrebbero essere implementati in un'architettura web 3-tier.

Traccia di una possibile risposta
I tre componenti rincipali di una generica applicazione sono Interfaccia Utente (o UI), Logica Applicativa e Gestione Dati che svolgono le seguenti funzioni . . .
In un architettura web 3-tier (ossia browser, front-end e back-end) la scelta è tra le seguenti opzioni:

- *mettere sul front-end il server HTTP assieme alla logica applicativa, lasciando nel banck-end solo i dati (es. DBMS)*
- *mettere sul front-end solo il server HTTP e posizionare sul back-end sia la logica applicativa sia la gestione dati*

Esercizio 6 ***(punti: 5)***
Si desidera proteggere un server HTTP dagli attacchi condotti via rete. Suggerire le difese da adottare e come devono essere configurate.

Traccia di una possibile risposta
Creare un firewall (di tipo packet filter) con una DMZ e posizionare il server sulla DMZ.
Miglioramento: esporre sulla DMZ non direttamente il server ma un reverse proxy (ossia un firewall applicativo) che faccia da interfaccia pubblica per il (oppure i) server.
Proteggere il collegamento degli utenti al server tramite un canale TLS (per evitare attacchi sul canale tra client e server) ed un'adeguata autenticazione degli utenti (es. HTTP basic authentication, meglio ancora TLS client authentication, o come minimo HTML form-based authentication).