Primer Certamen (Tiempo: 90 min.)

1.- Responda en forma breve y clara:

a) ¿Qué es control de flujo? Dé un ejemplo donde se requiera control de flujo.
Control de flujo es el mecanismo que permite adaptar la trasmisión de datos a lo que el receptor puede recibir. Ejemplo: cuando enviamos datos a un receptor lento (impresora), debemos hacer pausas para no sobrecargarlo. Otro: Cuando alimentamos un bebé, debemos adaptarlos a lo que éste puede comer.

b) ¿Qué significa que una aplicación siga el modelo cliente servidor? Mencione una aplicación que siga este modelo.
Significa que la aplicación posee dos tipos de procesos. Un proceso servidor que debe ser ejecutado en un puerto específico y queda a la espera de peticiones, y procesos clientes que deben saber dónde está el servidor y el puerto donde corre para solicitar servicios. Los clientes pueden correr desde cualquier parte y no requieren estar funcionando permanentemente.
Ejemplos son: la web, ssh, ftp, el correo electrónico.

c) Mencione una ventaja de la conmutación de circuitos y una ventaja de la conmutación de paquetes.
Conmutación por circuito: permite ofrecer garantías de ancho de banda y retardo.
Conmutación de paquetes: permite compartir enlaces cuando el tráfico es en ráfagas. No requiere establecimiento de llamada.

d) Explique qué mecanismo usa el sitio de Amazon para que www.amazon.com pueda identificar a un usuario cuando éste accede a esa página con posterioridad a haber hecho su primera compra allí.
Amazon hace uso de cookies. Al comprar, Amazon deja en el computador del usuario un identificador que es enviado a Amazon la siguiente vez que el usuario accede al sitio. Con ese identificador Amazon accede a una base de datos y reconoce al mismo usuario que compró antes.

e) Explique cuándo y cómo se usa el GET condicional.
GET condicional es usado cuando un cache desea saber si un contenido que él tiene ha sido actualizado en el servidor.
Cuando un navegador pide un mismo contenido al cache, éste envía un GET condicional indicando además la fecha del objeto que él posee. Si no hay una nueva versión en el servidor, éste lo informa en el encabezado y no envía los datos. Si hay actualización, la envía y el cache actualiza su objeto.

2.-

a) Suponga que usted está escribiendo un correo electrónico donde da instrucciones a alguien para bajar la página ubicada en http://www.elo322.cl/index.html haciendo uso de la aplicación **telnet**. ¿Cuáles serían sus instrucciones? Complete el correo adjunto. Asuma que la página contiene sólo el archivo de texto index.html.

Estimado Amigo:
Para bajar la página puedes usar el comando telnet. Escribe en una consola lo que indico:
$ telnet ..... / use estos comentarios para explicar cada paso */*
.....

Estimado Amigo:
Para bajar la página puedes usar el comando telnet. Escribe en una consola lo que indico:
$ telnet www.elo322.cl 80 / para conectarse al servidor web, puerto 80*/*
GET /index.html HTTP/1.1 / enviamos un GET y nombre de archivo más protocolo a usar */*
*HOST: www.elo322.cl /*como mínimo el encabezado debe tener el campo host*/*
/ dejar una línea en blanco*/*
/ aquí verás la respuesta desde el servidor */*
Control-C / para terminar el telnet presiona control-C o*
espera a que el servidor cierre la conexión/*

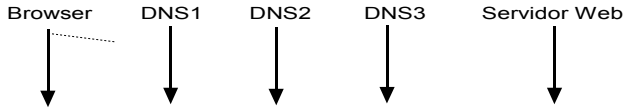
b) Usted dispone del programa TCPserver #puerto que espera por conexiones en el puerto #puerto. Ante una conexión de un cliente, TCPserver escribe a pantalla todo lo que reciba en su socket.

Indique las instrucciones a seguir para que un mensaje HTTP de su navegador aparezca en la pantalla donde corre TCPserver. Usted dispone de dos computadores en su red: servidor.elo.utfsm.cl y cliente.elo.utfsm.cl

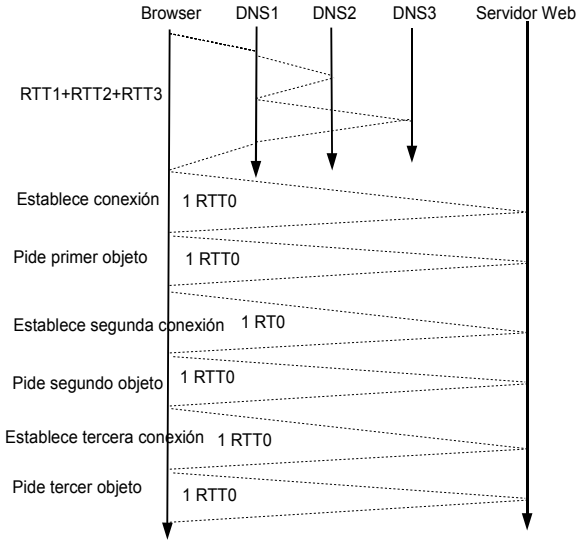
Supongo sistema operativo: Qué hacer en servidor.elo.utfsm.cl:	Supongo sistema operativo: Qué hacer en cliente.elo.utfsm.cl: .....

Sistema Operativo: linux *Qué hacer en servidor.elo.utfsm.cl:* *$ TCPserver 1234 /* correr servidor en puerto > 1024 */* */* Aquí aparecerá el mensaje HTTP*/* */* si se trata de un servidor escrito en java, sería:*/* */*$ java TCPserver 1234 */*	*Sistema Operativo: linux* *Usa lo que corresponda para correr un navegador y pon en el URL* http://servidor.elo.utfsm.cl:1234/

3.- Suponga que usted hace click en una liga que lo lleva hacia un sitio que su computador nunca ha visitado. Suponga que 3 servidores DNS son visitados para obtener la IP (DNS Iterativo). Asuma que los Round-Trip Time para cada consulta a los servidores DNS son RTT1, RTT2, RTT3.
La página visitada resulta tener 3 objetos muy pequeños (despreciar tiempo de transmisión), todos en el mismo servidor siendo el primero un texto HTML. Sea RTT0 el RTT entre su computador y el servidor web. Para cada caso haga un diagrama que muestre la secuencia de tiempos y calcule el tiempo desde el click hasta recibir los ~~4~~3 objetos:

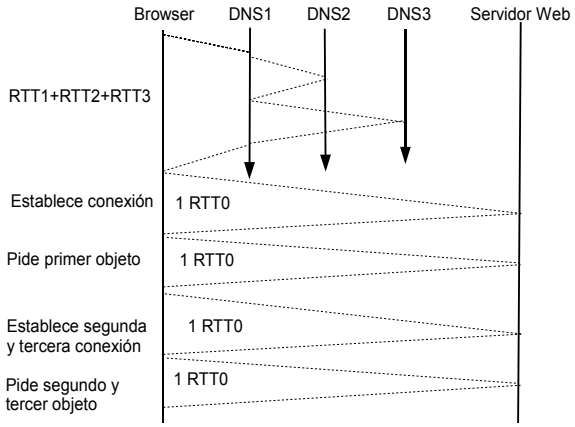
a) HTTP no persistente y sin conexiones paralelas. b) HTTP no persistente con conexiones paralelas. c) HTTP persistente con pipeline.	

a)



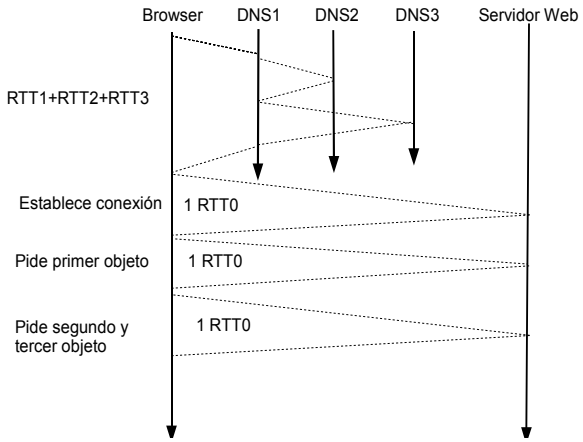
Total 6*RTT0+RTT1+RTT2+RTT3

b)

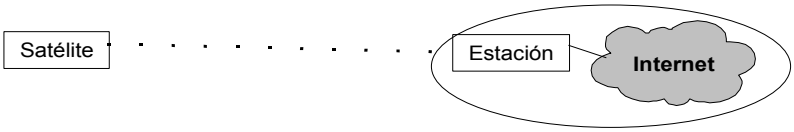


Total 4*RTT0+RTT1+RTT2+RTT3

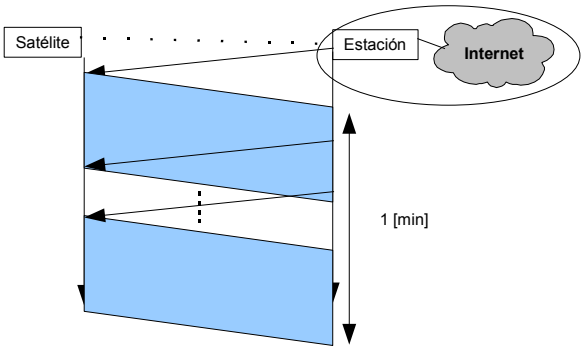
c)



4.- Cada 1 minuto tomamos una foto en un satélite a 35.800 km sobre la superficie de la tierra. El satélite está conectado a Internet a través de una estación en la tierra. El satélite tiene un servidor UDP que envía la última foto tomada cada vez que recibe un pequeño mensaje UDP de $t_{transmisión}$ despreciable desde la tierra. El mensaje de respuesta con la foto es de 20 [Mbytes], el enlace a tierra es de 10Mbps y rapidez de propagación de 2,4 x10^8 [m/s].
a) ¿Cuál es el número mayor de consultas desde Internet por minuto que puede atender el ~~equipo~~ satélite?
b) ¿Cómo sugiere usted aumentar el número de consultas atendibles por minuto? Cambiar el satélite no es una opción.



a)



En enlace es de 35.800 [km], lo cual genera un $t_{propagación}$ para todos los paquetes de:

$$t_{propagación} = \frac{35.8000 * 10^3 [m]}{2{,}4 * 10^8 [m/s]} = 0{,}149 [s]$$

El tiempo de transmisión $t_{transmisión}$ del mensaje UDP pequeño desde la tierra es despreciable, pero el de la foto es:

$$t_{transmisión} = \frac{20 * 1024^2 * 8 [bits]}{10 * 10^6 [bits/s]} = 16{,}8 [s]$$ *si usted uso 1K=1000 en medición de tamaños de archivo, es OK.*

$$t_{transmisión} = \frac{20 * 1000^2 * 8 [bits]}{10 * 10^6 [bits/s]} = 16 [s]$$

Luego se pueden atender en promedio $\frac{60[s]}{16{,}8[s]} = 3{,}6[requerimientos/min]$

(Alternativamente 3,75 [requerimientos/min])
Cabe destacar que no influye el tiempo de propagación.

b) Para aumentar el número de consultas atendibles por minuto, sugiero poner un cache en la estación terrestre. Éste bajará la foto ante una petición y la dejará en cache con tiempo de expiración de 1 minuto. Cualquier petición en ese intervalo sería atendida directamente. Después de un minuto el requerimiento demoraría lo que toma bajar una nueva foto.

Alternativamente puedo sugerir desarrollar una aplicación en la estación terrestre que baje la foto cada minuto y sea ésta quien la envíe a todas las peticiones desde Internet. En este caso la latencia o demora en enviar la foto sería menor. Claro que estaría bajando fotos que posiblemente nadie pedirá.

Basta que usted sugiera usar un cache en la la estación terrestre.