Examen de Redes - ETSINF - 5 de septiembre de 2011 (1º parcial)

*Apellidos, Nombre:*_____***SOLUCIONES***_____

*Grupo de matrícula:*_____***TODOS***_____

Una empresa decide nombrar su servidor web como “avionet.com”, nombre que coincide con su dominio de correo. Además, su servidor de correo SMTP será “smtp.avionet.com” y “dns1.avionet.com”, “dns2.avionet.com” sus servidores de nombres.

a) ***(0,75 puntos)*** Indica si la decisión anterior sería correcta. Justifica la respuesta. En caso de que no sea correcta, propón una solución alternativa.

b) ***(0,75 puntos)*** Indica los registros DNS que sería necesario incluir en los servidores dns1 y dns2 para poder acceder a los servidores web y SMTP de la empresa. Para cada registro indica el contenido de los campos tipo de registro, nombre y valor.

a) La decisión es correcta. No hay ningún problema de ambigüedad si el servidor web y el dominio tienen el mismo nombre, ya que la información se almacena en distintos tipo de registros DNS, y al realizar una consulta se indica el tipo de registro que almacena la información solicitada.

b)

	Tipo registro	Nombre	Valor
1	A	avionet.com	Dir. IP del servidor web
2	A	smtp.avionet.com	Dir. IP del servidor SMTP
3	MX	avionet.com	smtp.avionet.com

(0,25 puntos)¿Qué orden usamos en el programa smbclient para descargar un archivo del servidor? ¿Y para enviar un archivo al servidor?

Para enviar el archivo: put nombre_archivo

Para descargar el archivo: get nombre_archivo

(2 puntos) Un cliente HTTP se conecta a un servidor web para obtener una página HTML de 800 bytes que contiene un gráfico de 4.500 bytes (dos objetos en total). El diálogo HTTP se hará empleando una conexión persistente que permanecerá abierta después de enviar todos los objetos por si el cliente desea realizar nuevas peticiones. Todas las peticiones HTTP tienen un tamaño de 50 bytes.

El tamaño total de las cabeceras añadidas por todos los niveles (transporte, red y enlace) es de 100 bytes. La conexión emplea los mecanismos de control de la congestión TCP y ACKs retrasados. El MSS establecido al inicio de la conexión es de 1000 bytes. A lo largo de toda la conexión, los tamaños de la ventana del cliente y el servidor permanecerán constantes con los valores siguientes: WIN(C) = WIN(S) = 16 segmentos.

Suponiendo que no se pierden paquetes ni se producen retransmisiones, describe el intercambio de segmentos entre el cliente y el servidor hasta que todos los datos enviados hayan quedado reconocidos. Se supone que los números de secuencia iniciales son NSI(C) = 6.000 y NSI(S) = 7.500.

NOTA: El formato de los segmentos a representar tendrá en cuenta el número de secuencia, los flags de la cabecera TCP, el reconocimiento (si procede) y el campo de datos (si procede). Así como los valores de las ventanas de congestión y de transmisión (expresados en segmentos) tras enviar el segmento. Así por ejemplo:

Origen	Nº secuencia	Flags	Nº ACK	Datos	VCong	VTrans
C	51	ACK	200	51..100	2	1

representa un segmento emitido por el cliente C que lleva 50 bytes de datos, con números de secuencia del 51 al 100, un reconocimiento hasta el octeto 199, con el flag ACK activo. Una ventana de congestión que permite enviar un máximo de dos segmentos y una ventana de transmisión que indica que hay un único segmento de datos enviado y pendiente de reconocimiento.

Origen	Nº sec.	Flags	Nº ACK	Datos	VCong	VTrans
C	6000	SYN	-	-	2	0
S	7500	SYN, ACK	6001	-	2	0
C	6001	ACK	7501	-	2	0
C	6001	ACK, PUSH	7501	6001…6050	2	1
S	7501	ACK, PUSH	6051	7501…8300	2	1
C	6051	ACK, PUSH	8301	6051…6100	3	1
S	8301	ACK, PUSH	6101	8301…9300	3	1
S	9301	ACK, PUSH	6101	9301…10300	3	2
S	10301	ACK, PUSH	6101	10301…11300	3	3
C	6101	ACK	10301	-	4	0
C	6101	ACK	11301	-	4	0
S	11301	ACK, PUSH	6101	11301…12300	4(5)	1
S	12301	ACK, PUSH	6101	12301…12800	4(5)	2
C	6101	ACK	13301	-	4	0

(1 punto) a) Cuando se emplean conexiones http persistentes y tras enviar una petición GET, si el servidor responde enviando el objeto solicitado, ¿cómo sabe el cliente cuándo ha recibido todos los bytes del objeto?

b) ¿Y cuando la conexión es no persistente?

a) En la respuesta del servidor la cabecera "Content-Length" indica la longitud en bytes del objeto enviado en el cuerpo del mensaje. El cliente no tiene más que contar el número de bytes que va recibiendo del objeto.

b) Cuando la conexión HTTP es no persistente el servidor envía el objeto y, a continuación, cierra la conexión. El cliente no tiene más que leer del socket hasta detectar el cierre de la conexión.

(1 punto)¿Por qué no se aplica la técnica de reconocimientos retrasados cuando se envían reconocimientos duplicados?

Para que el otro extremo detecte cuanto antes la pérdida del segmento y lo pueda retransmitir.

(0,25 puntos)Suponiendo que el programa servidor web que has realizado en la práctica 5 lee del socket mediante un objeto de tipo Scanner que hemos llamado **lee**. Indica el código necesario para comprobar si la petición enviada por el cliente es un GET.

```
Scanner lee=new Scanner(cliente.getInputStream());
if ((lee.next()).equals("GET"))
```

(2 puntos) Un servidor TCP ofrece diversos servicios (implementados en la clase **ServidorPrivado**) a través de los puertos comprendidos entre el 4000 y el 4010. Inicialmente estos servicios están inactivos y sólo se activan a petición del cliente. Para activar un servicio, el cliente debe conectarse al puerto 5000 (donde está permanentemente escuchando un servidor TCP iterativo, implementado en la clase **ServidorAcceso**) y enviar el número de puerto correspondiente al servicio que solicita (`port`). Si el puerto solicitado está fuera del rango permitido, el servidor de acceso enviará al cliente el mensaje `"prohibido" + port.` El cliente no podrá realizar la conexión solicitada.

Si el puerto solicitado está dentro del rango permitido [4000, 4010], el servidor de acceso comprobará si el puerto solicitado está libre (haciendo uso las excepciones de la clase ServerSocket). Si lo está, enviará al cliente el mensaje `"adelante" + port` y realizará la llamada `ServidorPrivado(port).start()` para que un nuevo *thread* atienda al cliente en el puerto solicitado. A continuación, el cliente podrá realizar la conexión al puerto `port`. Si ServidorAcceso encuentra el puerto ocupado, le enviará al cliente el mensaje `"ocupado" + port` y el cliente no podrá realizar la conexión solicitada.

Implementa en Java la clase **ServidorAcceso**.

```
import java.net.*;
import java.util.Scanner;
import java.io.*;

public class ServidorAcceso {
  public static void main(String args[]) throws Exception {
    ServerSocket sa = new ServerSocket(5000);
    while(true) {
      Socket s = sa.accept();
      Scanner entrada = new Scanner(s.getInputStream());
      PrintWriter eixida = new PrintWriter(s.getOutputStream(),true);
      int port = entrada.nextInt();
      if (port < 4000 || port >4010) {
        eixida.println("prohibido " + port);
      }
      else {
        try{
          ServerSocket server = new ServerSocket(port);
          server.close();
          eixida.println("adelante " + port);
          ServidorPrivado sp = new ServidorPrivado(port);
          sp.start();
        }
        catch(IOException e){
          eixida.println("ocupado " + port);
        }
      }
      s.close();
    }
  }
}
```

(1 punto) Considera dos computadores, A y B, conectados por un enlace de 1.5 Mbps. El tiempo de propagación es t_{prop}=514 msec. El computador A envía un paquete de 150 bytes al computador B.

a) Calcula el tiempo que tarda en recibirse en B el paquete completo.

b) Si A comienza a transmitir el paquete en t = 0, ¿dónde se encuentra el último bit del paquete cuando t=t_{trans}?

c) ¿Qué tamaño tendría que tener el paquete para que t_{trans}=t_{prop}?

A) $T_{total} = T_{trans} + T_{prop} = 150x8/1.5x10^{6} + 514x10^{-3} = 514.8x10^{-3}$ s.

B) Cuando t=T_{trans}, el ultimo bit acaba de salir de A. Está al principio de la línea.

C) Igualando $L/1.5x10^{6} = 514x10^{-3}$

$L = 514*1.5*10^{3} = 771*10^{3}$ bits

(1 punto) Si arrancamos dos clientes web (navegador) en el mismo computador y enviamos una petición HTTP por cada uno de los clientes hacia el mismo servidor y el mismo puerto (80):

(a) ¿Cómo diferencia el servidor a qué navegador debe entregar cada respuesta?

(b) Si en lugar de la aplicación web se tratase de una aplicación que utilizara el servicio UDP para el transporte de los mensajes, ¿qué diferencias de funcionamiento existirían con el caso anterior?

a) Para cada segmento recibido se analiza a qué Conexión pertenece. Una conexión se caracteriza, no solo por IP y puerto destino, sino también por origen. Puesto que los puertos origen son distintos, no hay ambigüedad con el socket destino.

b) En UDP no hay conexiones, y por tanto todos los datagramas se entregan al mismo socket independientemente de su origen.