

REDES DE COMPUTADORAS - PRIMER PARCIAL Es del 2014

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur - 14/5/2013

Nombre: _____	LU: _____
Cantidad de hojas entregadas: _____	Preguntas realizadas: _____

Observaciones

- ✓ Apague el celular y manténgalo guardado.
- ✓ Numere y ponga su nombre a todas las hojas. Indique cuantas hojas entrega.
- ✓ Lea todo el ejercicio antes de comenzar a desarrollarlo.

Ejercicio 1:

- a) ¿Cuáles son las cinco capas de la pila de protocolos de Internet?
- b) ¿Cuáles son las siete capas de la pila de protocolos según el modelo OSI?
- c) Sabiendo que en cada capa se manipulan diferentes mensajes, confeccione una tabla asociando a cada capa los siguientes términos: segmento, dato, trama, datagrama y *bits*.
- d) Especifique el orden de encapsulamiento de los términos de acuerdo al inciso (a).

Ejercicio 2: Defina los siguientes términos:

- a) Ancho de banda
- b) Latencia

Ejercicio 3: En base a las siguientes direcciones IP determinar los siguientes datos:

- Dirección de red
- Cantidad de *bits* de *hosts*
- Cantidad de equipos (*hosts*) de la subred
- Dirección IP del primer equipo de la subred
- Dirección IP del último equipo de la subred
- Dirección de *broadcast*

- a) 92.30.1.33/16
- b) 200.63.133.23/255.255.255.252

Ejercicio 4:

- a) ¿Cuál es la primera subred para la red 27.0.0.0 con máscara 255.192.0.0?
- b) Si se cuenta con las siguientes subredes: 140.120.4.0, 140.120.8.0, 140.120.12.0, 140.120.16.0; ¿cuál es la máscara que debe utilizarse en cada una de ellas?

Ejercicio 5:

Suponga un canal de comunicación en el cual se presentan errores a nivel de *bits* y pérdida de paquetes. Para poder comunicarse a través de este canal se requieren cuatro características:

- Implementar protocolo *stop and wait*;
- Detección de errores;
- Realimentación del emisor;
- Retransmisión luego de un límite de tiempo.

Dada una máquina de estados finita que modele el problema enunciado responda:

- a) ¿Cuántos estados debe tener el emisor?
- b) ¿Se define un rango de números de secuencia para todos los paquetes (de datos y reconocimiento)?
- c) ¿El emisor enviará datos sin recibir un paquete de reconocimiento?
- d) ¿Cuántos estados debe tener el receptor?
- e) ¿Se tiene en cuenta la corrupción de los paquetes de reconocimiento? Justifique.
- f) ¿En base a qué probabilidad se define la duración del *timer* de retransmisión?
- g) ¿Cómo se denomina el protocolo definido por el autómata?

Ejercicio 6:

Se desean transmitir 4 paquetes haciendo uso del protocolo *GBN*, para una ventana de transmisión de tamaño 4 usando de 0 a 3 como números de secuencia. Suponer que se pierde el envío del primer paquete y que se pierde el mensaje de reconocimiento del segundo paquete. Adopte un *timeout* de 1 *RTT*. Grafique el intercambio de paquetes que se produce entre el emisor y el receptor.

Ejercicio 7:

- Grafique el método (*3-way handshaking*) que se utiliza en el protocolo *TCP* para establecimiento de una conexión mediante sockets entre dos equipos A y B dentro de una red.
- Indicar en el gráfico anterior los números de secuencia de cada paquete con los mensajes *SYN* y *ACK*.
- ¿Por qué existe *UDP*? ¿No sería suficiente permitir a los procesos de usuario enviar directamente paquetes IP? Justifique su respuesta.

Ejercicio 8:

Dado el siguiente servidor "echo" (permite obtener como respuesta el dato enviado al server). Determine que error se ha cometido en la programación con *sockets* e indique por qué no es posible su correcto funcionamiento. Pista: falta sólo una sentencia en el código proporcionado.

```
int main(int argc, char **argv)
{
    int listenfd; /* listening socket */
    int connfd; /* connection socket */
    int portno; /* port to listen on */
    int clientlen; /* byte size of client's address */

    struct sockaddr_in serveraddr; /* server's addr */
    struct sockaddr_in clientaddr; /* client addr */

    struct hostent *hostp; /* client host info */
    char buf[BUFSIZE]; /* message buffer */
    char *hostaddrp; /* dotted decimal host addr string */
    int optval; /* flag value for setsockopt */
    int n; /* message byte size */

    if (argc != 2) { fprintf(stderr, "Usage: %s <port>\n", argv[0]); exit(1); }
    portno = atoi(argv[1]);

    listenfd = socket(AF_INET, SOCK_STREAM, 0);
    if (listenfd < 0) error("ERROR opening socket\n");

    optval = 1;
    setsockopt(listenfd, SOL_SOCKET, SO_REUSEADDR, (const void *)&optval, sizeof(int));
    bzero((char *) &serveraddr, sizeof(serveraddr));
    serveraddr.sin_family = AF_INET;
    serveraddr.sin_addr.s_addr = htonl(INADDR_ANY);
    serveraddr.sin_port = htons((unsigned short)portno);

    if (bind(listenfd, (struct sockaddr *) &serveraddr, sizeof(serveraddr)) < 0)
        error("ERROR on binding\n");

    //
    clientlen = sizeof(clientaddr);

    while (1) {
        connfd = accept(listenfd, (struct sockaddr *) &clientaddr, &clientlen);
        if (connfd < 0) error("ERROR on accept\n");

        hostp = gethostbyaddr((const char *)&clientaddr.sin_addr.s_addr,
            sizeof(clientaddr.sin_addr.s_addr), AF_INET);
        if (hostp == NULL) error("ERROR on gethostbyaddr()\n");

        hostaddrp = inet_ntoa(clientaddr.sin_addr);
        if (hostaddrp == NULL) error("ERROR on inet_ntoa()\n");

        printf("Server established connection with %s (%s)\n", hostp->h_name, hostaddrp);
        bzero(buf, BUFSIZE);

        n = read(connfd, buf, BUFSIZE);
        if (n < 0) error("ERROR reading from socket\n");

        printf("Server received %d bytes: %s\n", n, buf);
        n = write(connfd, buf, strlen(buf));
        if (n < 0) error("ERROR writing to socket\n");
        close(connfd);
    }
}
```


Ejercicio 9:

Dado la siguiente secuencia de comandos e instrucciones, necesaria para poder determinar a qué servidor de electrónico se debe enviar un correo electrónico, responda:

- ¿Cuál es el servidor de destino del correo electrónico?
- ¿Cuál es el usuario de destino?
- ¿Cuál es el usuario de origen?
- ¿Por qué el email fue aceptado por el servidor de destino? Justifique su respuesta.

Paso 1: Búsqueda del registro MX para el dominio: @compras.com.ar

```
[ldm@beta ~]$ nslookup -query=mx compras.com.ar
Server:      127.0.0.1
Address:     127.0.0.1#53
Non-authoritative answer:
compras.com.ar mail exchanger = 10 mx.compras.com.ar.
```

```
Authoritative answers can be found from:
com.ar nameserver = a.dns.ar.
com.ar nameserver = ns2.switch.ch.
com.ar nameserver = athea.ar.
com.ar nameserver = ctina.ar.
com.ar nameserver = relay1.mecon.gov.ar.
a.dns.ar internet address = 200.108.145.50
a.dns.ar has AAAA address 2801:140::10
athea.ar internet address = 200.16.98.2
ctina.ar internet address = 200.16.97.17
relay1.mecon.gov.ar internet address = 168.101.16.10
```

Paso 2: Conexión al servidor de correo del dominio @compras.com.ar

```
[ldm@beta ~]$ telnet mx.compras.com.ar 25
telnet: Trying 90.7.224.6...
telnet: Connected to mx.compras.com.ar.
telnet: Escape character is '^'.
server: 220 compras.com.ar ESMTS Sendmail; Tue, 13 May 2014 18:05:27 -0300
client: HELO client.example.com
server: 250 compras.com.ar Hello mail.bestbuy.com.ar [223.59.40.18], pleased to meet you
client: MAIL FROM: <carlos@bestbuy.com.ar>
server: 250 2.1.0 <carlos@bestbuy.com.ar>... Sender ok
client: RCPT TO: <mariano@compras.com.ar>
server: 250 2.1.5 <mariano@compras.com.ar>... Recipient ok
client: DATA
server: 354 Enter mail, end with "." on a line by itself
client: From: sender@example.com
client: To: recipient@example.com
client: Subject: Test message
client: This is a test message.
client:
server: 250 2.0.0 s4DL5RO7024922 Message accepted for delivery
client: QUIT
server: 221 2.0.0 compras.com.ar closing connection
server: Connection closed by foreign host.
```

Paso 3: Correo electrónico recibido y depositado en la casilla de: mariano@compras.com.ar

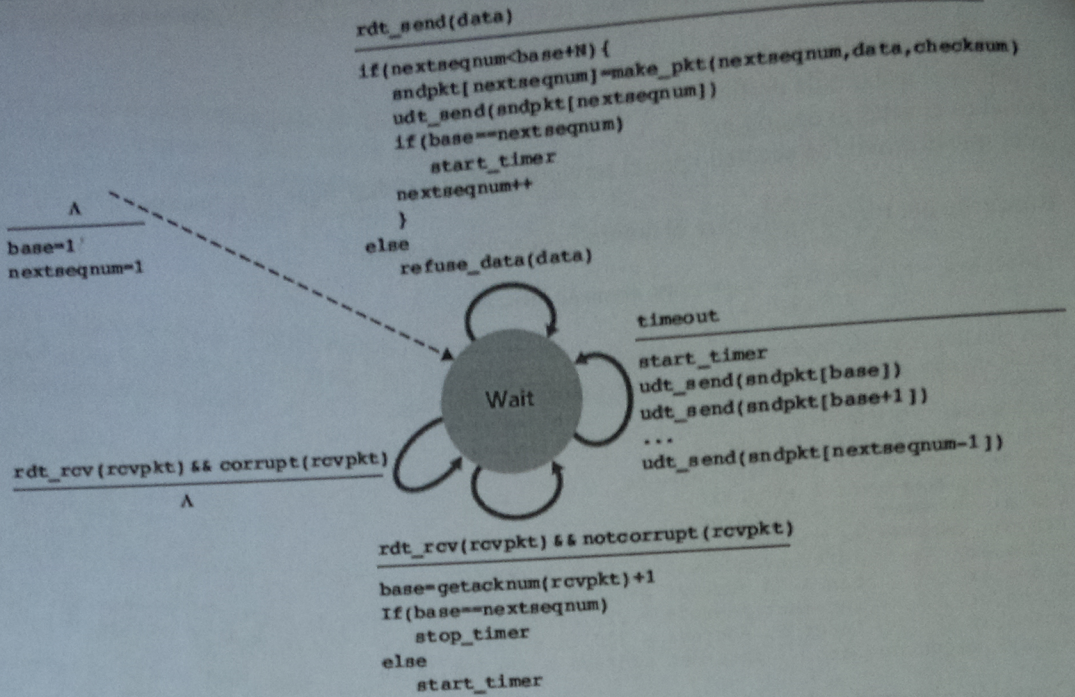
```
[ldm@mx /var/mail]$ cat mariano
From carlos@bestbuy.com.ar Tue May 13 18:05:58 2014
Return-Path: <carlos@bestbuy.com.ar>
Received: from client.example.com (mail.bestbuy.com.ar [223.59.40.18])
        by compras.com.ar (8.13.8/8.13.8) with SMTP id s4DL5RO7024922
        for <mariano@compras.com.ar>; Tue, 13 May 2014 18:05:39 -0300
Date: Tue, 13 May 2014 18:05:27 -0300
Message-Id: <201405132105.s4DL5RO7024922@compras.com.ar>
From: sender@example.com
To: recipient@example.com
Subject: Test message
X-Spam-Status: No, score=0.9 required=5.0 tests=MISSING_DATE,MISSING_MID,
        RP_MATCHES_RCVD,SPF_PASS autolearn=disabled version=3.3.1
X-Spam-Checker-Version: SpamAssassin 3.3.1 (2010-03-16) on mx.compras.com.ar
X-SPF-Scan-By: smf-spf v2.0.2 - http://smfs.sf.net/
Received-SPF: Pass (compras.com.ar: domain of carlos@bestbuy.com.ar
        designates 223.59.40.18 as permitted sender)
        receiver=compras.com.ar; client-ip=223.59.40.18;
        envelope-from=<carlos@bestbuy.com.ar>; helo=client.example.com;
```

This is a test message.

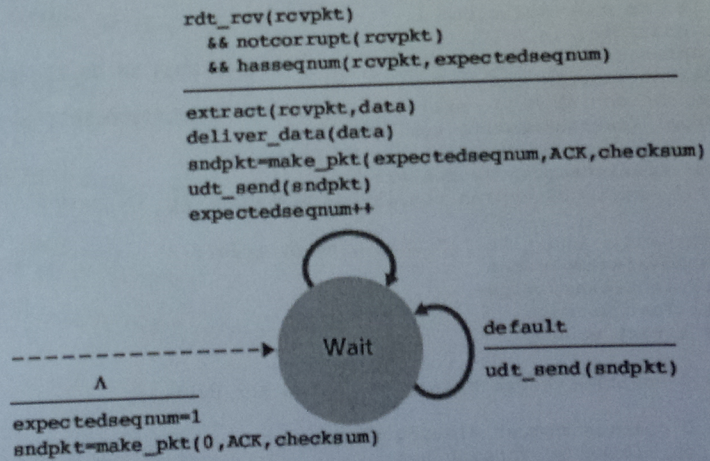
Ejercicio 6:

Se da:

GBN Emisor



GBN Receptor



del protocolo *GBN*, para una ventana de transmisión de tamaño 3 y
Suponer que se pierde el envío del primer paquete y que se pierde el
e. Adopte un *timeout* de 1 *RTT*.
luce entre el emisor y el receptor.

e se utiliza en el protocolo *TCP* para establecimiento de una conexión
entro de una red.