

**Solución del Primer Parcial – 24 de setiembre de 2016**  
(ref: solprc20160924.odt)

**Instrucciones**

- Indique su nombre completo y número de cédula en cada hoja.
- Numere todas las hojas e indique en la primera la cantidad total de hojas que entrega.
- Escriba las hojas de un solo lado y utilice una caligrafía claramente legible.
- Comience cada pregunta en una hoja nueva.
- Sólo se responderán dudas de letra. No se responderán dudas de ningún tipo durante los últimos 30 minutos de la prueba.
- La prueba es individual y sin material. Apague su teléfono celular mientras esté en el salón de la prueba.
- Duración: 2 horas. Culminadas las 2 horas, el alumno no podrá modificar de ninguna forma las hojas.
- Justifique todas sus respuestas.

**Pregunta 1 (4 puntos)**

Mencione las principales características del protocolo FTP (File Transfer Protocol). En particular indique a qué capa pertenece, sobre qué protocolo de la capa inmediata inferior se implementa, y qué tipo de conexión de control utiliza.

**Solución**

Se trata de un protocolo de Capa de Aplicación que utiliza a TCP como protocolo de Capa de Transporte y que permite la transferencia de archivos desde y hacia un equipo remoto basado en la arquitectura cliente/servidor.

En una sesión FTP se establecen 2 conexiones TCP: una de control (del tipo "out-of-band"), desde el cliente al puerto 21 del servidor a través de la cual se envían los diferentes comandos (listado de directorio, cambio de directorio, solicitud de transferencia de archivo, ...). La segunda conexión, de datos, se puede establecer tanto desde el servidor (puerto origen 20) como desde el cliente, según el modo de trabajo.

**Pregunta 2 (2 puntos)**

Compare las estrategias "conmutación de circuitos" y "conmutación de paquetes" respecto al uso de los recursos de la red.

**Solución**

Como la conmutación de circuitos implica la reserva previa -antes de enviar los datos- de los recursos de la red, en los momentos ociosos de esos recursos -cuando no se utilizan- ellos no pueden ser utilizados por otros usuario de la red. Como en la conmutación de paquetes no hay reserva de recursos, este desaprovechamiento no ocurre.

**Pregunta 3 (4 puntos)**

Desde la dirección `guri@asado.uy` se quiere enviar un correo electrónico a `vaina@arepas.ve`, utilizando comandos del protocolo SMTP desde una terminal del cliente denominado `localhost`.

- a) Cómo debe hacer el cliente para saber con qué servidor debe comunicarse?
- b) Muestre cuál sería la secuencia esperable de comandos y salidas a observar en la terminal

## Redes de Computadoras

del cliente en su conexión con el servidor para enviar el correo.

### Solución

a) En primer se debe consultar al DNS por el servidor de mail del dominio asado.uy. Supongamos que devuelve mail.asado.uy. También es válido consultar al DNS por el servidor de mail del dominio arepas.ve y establecer el telnet a él.

b)

```
$ telnet mail.asado.uy 25
```

```
S: 220 mail.asado.uy
C: HELO localhost
S: 250 Hello localhost, pleased to meet you
C: MAIL FROM: <guri@asado.uy>
S: 250 guri@asado.uy ... Sender ok
C: RCPT TO: <vaina@arepas.ve>
S: 250 vaina@arepas.ve ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: En Uruguay nos encanta el asado
C: Qué vaina con las arepas?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 mail.asado.uy closing connection
```

Notación:

S - Servidor

C - Cliente

### Problema 1 (10 puntos)

Se desea adicionar al rdt 3.0 un mecanismo que permita al emisor conocer que el receptor está activo y funcionando. El mecanismo no debe agregar nuevos tipos de mensajes a los ya utilizados por rdt 3.0. Si transcurrido un tiempo X se detecta que el receptor no está activo, se debe finalizar el emisor con un mensaje a la capa superior.

Suponga que el canal pierde y daña mensajes, pero mantiene el orden de transferencia de los mismos.

#### Se pide:

- Proponga un valor para el tiempo X en función de un RTT conocido.
- Implemente la máquina de estados del emisor y receptor del protocolo solicitado.

### Solución

a) La duración del timer debe permitir como mínimo el tiempo requerido para el envío del paquete y de la respuesta de aceptación. Por lo tanto debe ser como mínimo RTT, y podría considerarse  $X=RTT$ .

Cuanto mayor sea el timer, más tiempo tardará en reconocer la caída del receptor.

Sin embargo, como puede haber pérdida de mensajes, el receptor debería enviar un mensaje cada menos tiempo, por ejemplo  $RTT/2$ . Esto asegura que ante la pérdida de un mensaje, el receptor no será desconectado.

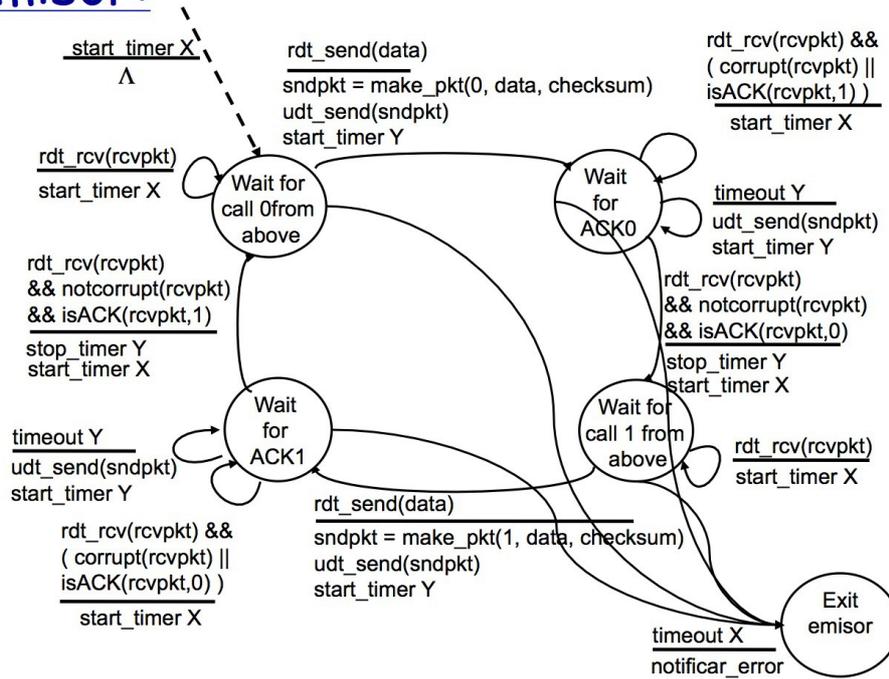
# Redes de Computadoras

b)

## Emisor:

El emisor utiliza dos timers, uno (Y) definido en el rdt 3.0 para generar retransmisiones e implementar el protocolo stop & wait, y otro (X) que cuenta cuanto tiempo hace que no recibe nada del receptor. Cuando se supere el tiempo X especificado irá a un estado de Error, indicando que se ha perdido la conexión.

## Emisor:



## Redes de Computadoras

Receptor:

El receptor cuenta el tiempo que pasa sin enviar nada, y si se supera  $X/2$  (valor configurado para el timer Z), re-envía el último ACK enviado. En la solución propuesta, se puede perder un envío sin que se desconecte al receptor.

Es importante notar que no se agregan mensajes nuevos, simplemente una re-transmisión que permite alimentar la máquina del emisor y evitar la desconexión por inactividad.

