Primer Parcial – 18 de setiembre de 2023

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 15 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 (12 puntos)

a) Compare conmutación de circuitos y conmutación de paquetes.
b) Para cada tipo de conmutación, mencione una aplicación que se beneficia de la misma.
c) ¿Qué significa que un conmutador implementa el método de conmutación de almacenamiento y reenvío (*store&forward*)?
d) ¿Que retardos introduce este modo de funcionamiento?

Solución Pregunta 1

a) Entre los rasgos mas característicos en los que difieren estos dos métodos tenemos:
   - en conmutación de paquetes los datos del usuario comienzan a viajar antes, ya que no necesita establecer el circuito previamente a la transmisión de datos.
   - Conmutación de circuitos puede asegurar calidad de servicio mediante reserva de recursos a lo largo del circuito, mientras que conmutación de paquetes en generalmente “best effort”.
   - Si bien conmutación de circuitos permite garantizar niveles de servicio, tiende a ser ineficiente en el uso del canal si las aplicaciones que los usan no utilizan el mismo ya que esos recursos no son compartidos. En conmutación de paquetes en cambio, cualquier aplicación puede transmitir si el canal está libre, por lo tanto no se limita a nadie.
   - Conmutación de paquetes es mas robusto ante la caída de enlaces, y eventualmente cada paquete podría viajar por caminos diferentes uno de otro. En el caso de circuitos la caída de un enlace termina con el circuito y la comunicación.

b) Para la conmutación de circuitos es claro que las aplicaciones que utilicen un cierto nivel de servicio mas o menos constante son las mas beneficiadas. Originalmente se hablaba de telefonía, hoy podemos hablar de aplicaciones de streaming de video, música on-line y similares. Para la de paquetes tenemos aplicaciones que pueden funcionar bien sin reserva de recursos y en ráfagas, como por ejemplo HTTP.

c) Refiere a que el conmutador debe recibir todos los bits correspondientes al paquete para procesarlo y luego ponerlo en la cola de salida correspondiente para su transmisión en el enlace de salida.
d) Esto genera retardos de cola y de procesamiento.

Pregunta 2 (12 puntos)

a) Dado un protocolo RDT, mencione y justifique qué mecanismos se deben agregar para soportar un canal con errores de bit y pérdidas.
b) Explique por qué a los protocolos RDT se dice que son de parada y espera (*stop&wait*).
c) Mencione las dos alternativas vistas en el curso para mejorar el rendimiento de los protocolos de parada y espera. Explique con mayor detalle una de ellas.

Solución Pregunta 2

a) Esas son las hipótesis del RDT 3.0, así que tenemos los mecanismos de:
   - ACK para confirmar mensajes de datos correctamente recibidos usando chequeos de correctitud (checksum) para corroborar que no hay errores de bit.
   - timers para retransmisiones ante pérdidas. Se pudo perder el paquete de datos o su confirmación, de manera que el timer se dispara al no recibir ACK luego de un timeout.
   - numero de secuencia (bit alternante). Dado que hay retransmisiones, es importante poder identificar qué paquete de datos y/o ack es el que se está recibiendo. Esto evita eventuales

duplicaciones.

b) Porque envían un paquete con datos, y no envían el siguiente hasta no recibir confirmación de que el mismo fue recibido por el destino.
c) Ante los problemas de rendimiento que tienen los protocolos stop&wait es que se crean los pipelined, tratando de mejorar el uso del canal. Los vistos son Go Back N (GBN) y Selective Repeat (SR).
SR envía varios paquetes (el tamaño de la ventana) sin ser confirmados, y permite que sean confirmados de forma individual y no en orden por parte del receptor. Entonces también puede reenviar por timeout solo los que no fueron confirmados.
GBN envía varios paquetes (el tamaño de la ventana) sin ser confirmados, y el receptor los confirma de manera individual. El receptor no acepta paquetes de datos que sean de secuencias mayores a la que espera (por ejemplo porque se perdió uno anterior). También se maneja el ACK acumulativo, donde el ACK de N implica que se recibió correctamente los anteriores también. En este caso el receptor puede liberar de su buffer esos paquetes y corre la ventana para enviar nuevos datos. El emisor tiene un timer para el paquete mas viejo no confirmado, y al vencer vuelve a enviar el mensaje perdido y todos los siguientes que había transmitido.

Pregunta 3 (8 puntos)

Suponga que existe un enlace de microondas a 10 Mbps entre un satélite geoestacionario (situado a 36000 Km de altura sobre el Ecuador) y su estación base en la Tierra. El satélite toma una fotografía digital por minuto y la envía a la estación base. La velocidad de propagación es $2,4 \times 10^8$ metros/segundo.

a) ¿Cuál es el retardo de propagación del enlace?
b) ¿Cuál es el producto ancho de banda-retardo, $R \times d_{prop}$?
c) Sea **x** el tamaño de la fotografía. ¿Cuál es el valor mínimo de **x** para que el enlace de microondas esté transmitiendo continuamente?

Solución Pregunta 3

a) d_{prop} = largo del enlace / velocidad de propagación
largo del enlace = 36.000 Km = 36×10^6 m
velocidad de propagación es $2,4 \times 10^8$ m/s
$d_{prop} = 36x10^6$ m / $2,4 \times 10^8$ m/s = 36 m / $2,4x10^2$ m/s = 36 m / 240 m/s = 3,6 m / 24 m/s =0,15 s
b) $R \times d_{prop} = 10x10^6$ bps * 0,15 s = 1.500.000 bits.
c) Necesito conocer el tamaño de x para que el enlace esté siempre ocupado. Para eso necesito una foto que ocupe todo el ancho de banda durante los 60 segundos que le lleva al satélite tener la siguiente foto para transmitir. El enlace puede transmitir $10x10^6$ bps, entonces con fotos de hasta 600.000.000 bits podremos ocupar todo el enlace durante 60 segundos sin solaparse con la transmisión de la siguiente foto.

Pregunta 4 (10 puntos)

a) Explique la necesidad multiplexar/demultiplexar en capa de transporte, y relaciónelo con el concepto de proceso.
b) Detalle la demultiplexión con TCP, y diga en qué se diferencia con la de UDP.

Solución Pregunta 4

a) La capa de red entrega paquetes de un host origen a un host destino, pero en un mismo host pueden existir muchos procesos comunicándose con otros procesos en otros hosts (o incluso dentro del mismo). Esto hace insuficiente el uso únicamente de las direcciones de host (IP) para identificar los orígenes y destinos de esas comunicaciones, y es allí donde es necesario un identificador extra para identificar procesos dentro de hosts, utilizado para la multiplexación/demultiplexación. Es así que quien envía datos debe multiplexar, agregando estos identificadores de procesos (puertos) origen/destino, para que luego el receptor recibiendo estos datos sea capaz de en base a esos identificadores los entregue a los procesos que corresponde (demultiplexación).
b) Cita textual del libro de la sección "*Multiplexación y demultiplexación orientadas a la conexión*": "Para entender la demultiplexación TCP, tenemos que tener en cuenta los sockets TCP y el

establecimiento de las conexiones TCP. Una sutil diferencia entre un socket TCP y un socket UDP es que el primero queda identificado por una tupla de cuatro elementos: dirección IP de origen, número de puerto de origen, dirección IP de destino, número de puerto de destino. Por tanto, cuando un segmento TCP llega a un host procedente de la red, el host emplea los cuatro valores para dirigir (demultiplexar) el segmento al socket apropiado. En particular, y al contrario de lo que ocurre con UDP, dos segmentos TCP entrantes con direcciones IP de origen o números de puerto de origen diferentes (con la excepción de un segmento TCP que transporte la solicitud original de establecimiento de conexión) serán dirigidos a dos sockets distintos."

Pregunta 5 (8 puntos)

Considere la siguiente cadena de caracteres ASCII capturados con Wireshark a partir de una conexión TCP:

```
HTTP/1.1 200 OK<cr><lf>Date: Sun, 10 Sep 2023 19:53:29 GMT<cr><lf>Server:
Apache<cr><lf>Last-Modified: Fri, 17 Sep 2021 19:26:14 GMT<cr><lf>Accept-Ranges:
bytes<cr><lf>Vary: Accept-Encoding,User-Agent<cr><lf>Content-Encoding:
gzip<cr><lf>Content-Length: 12038<cr><lf>Keep-Alive: timeout=15,
max=93<cr><lf>Connection: Keep-Alive<cr><lf>Content-Type: text/html<cr><lf>Set-
Cookie: ABCD=abcd; expires=Mon, 11-Sep-2023 01:53:29 GMT; path=/;
Httponly<cr><lf><cr><lf>
```

Los caracteres <cr><lf> son caracteres de retorno de carro y avance de línea (es decir, la cadena de caracteres <cr> en el texto representa el único carácter de retorno de carro).

Responda las siguientes preguntas, justificando las respuestas en base a la captura:

a) ¿Qué parte del mensaje muestra la captura, y a qué protocolo corresponde?
b) ¿Quién envía el mensaje, el iniciador de la conexión o el que la aceptó?
c) ¿Es una conexión persistente o no persistente?
d) ¿Qué puede decir sobre el cuerpo del mensaje?

Solución Pregunta 5

a) La captura muestra el encabezado de un mensaje del protocolo HTTP 1.1
b) Dado que es un mensaje de respuesta (HTTP/1.1 200 OK) podemos decir que lo envía el que la aceptó, que sería el servidor respondiendo a un pedido de un cliente.
c) Persistente. Eso se explicita en los encabezados Keep-Alive y Connection.
d) El cuerpo del mensaje no está incluido en lo que se muestra, pero debería estar a continuación de <cr><lf><cr><lf> que es lo que delimita los encabezados del inicio del body. El body serían los 12038 bytes siguientes (Content-Length: 12038) que vendrán comprimidos (Content-Encoding: gzip) y que el contenido comprimido es un HTML (Content-Type: text/html). También podríamos decir que ese recurso no ha sido modificado desde el 17 Sep 2021 (Last-Modified: Fri, 17 Sep 2021 19:26:14 GMT).