

## Solución Primer Parcial – 04 de octubre de 2021

### Nota

El parcial se tomó en modalidad virtual a través de la plataforma EVA.

Consistió en 5 preguntas de un set de preguntas que se sortearon de forma aleatoria para cada estudiante.

Tuvo una duración de 90 minutos.

### Pregunta 1 (8 puntos)

V1

Considere un paquete de longitud 1.500 bytes que tiene su origen en un sistema terminal origen y que viaja a través de tres enlaces hasta un sistema terminal de destino. Estos tres enlaces están conectados mediante dos dispositivos de conmutación de paquetes. Los dispositivos de conmutación de paquetes demoran 2 milisegundos en procesar cada paquete. La velocidad de propagación en ambos enlaces es igual a  $2,5 \times 10^8$  m/s, la velocidad de transmisión en los tres enlaces es de 4 Mbps, la longitud del primer enlace es de 8000km, la del segundo de 4000km y la del último enlace es de 2000 km.

**Se pide:** plantee, calcule y justifique cuál es el retardo terminal a terminal.

Siendo  $L$  la longitud del paquete,  $R_i$  las velocidades de transmisión,  $d_i$  las distancias de los enlaces y si las velocidades de propagación (con  $i=1,2,3$ ) tenemos:

el primer sistema terminal requiere  $L/R_1$  para transmitir el paquete al primer enlace, dicho paquete se propagará por el primer enlace en  $d_1/s_1$ ; el primer conmutador de paquetes agrega un retardo de procesamiento de  $d_{proc}$ ; luego de recibir el paquete completo el conmutador requiere  $L/R_2$  para transmitirlo al segundo enlace el que agregará  $d_2/s_2$  para propagarlo. Del mismo modo actúa el segundo conmutador, obteniendo:  $d_{terminal-terminal} = L/R_1 + d_1/s_1 + d_{proc} + L/R_2 + d_2/s_2 + d_{proc} + L/R_3 + d_3/s_3$ .

Sustituyendo por los valores tenemos:

$$d_{terminal-terminal} = 1500 \cdot 8 / 4 \cdot 10^6 + 8000 \cdot 10^3 / 2,5 \cdot 10^8 + 2 + 1500 \cdot 8 / 4 \cdot 10^6 + 4000 \cdot 10^3 / 2,5 \cdot 10^8 + 2 + 1500 \cdot 8 / 4 \cdot 10^6 + 2000 \cdot 10^3 / 2,5 \cdot 10^8 = 3 + 32 + 2 + 3 + 16 + 2 + 3 + 8 = 69ms$$

Conversiones:

$L$  bytes  $\rightarrow$  bits

$R_i$  Mbps  $\rightarrow$  bps

$d_i$  Km  $\rightarrow$  m

si m/s  $\rightarrow$  m/s

$d_{proc}$  ms  $\rightarrow$  ms

resultados en ms

V2

a) Enumere tres protocolos de capa de aplicación, explique brevemente su principal función y que protocolo de capa de transporte utilizan.

b) En una sesión de comunicación entre dos procesos, ¿qué proceso es el cliente y qué proceso es el servidor? Justifique.

## Redes de Computadoras

c) En una aplicación P2P o protocolo entre pares, cuando dos pares se conectan utilizando TCP, ¿es posible identificar el cliente y el servidor? Justifique.

a)

SMTP → TCP, su principal función es intercambiar correos electrónicos entre servidores de correo.

POP → TCP, su principal función es que un agente de correo obtenga de un servidor los correos de su casilla.

HTTP → TCP, su principal función es permitir el pedido y la transferencia de recursos a un servidor web como por ejemplo documentos HTML.

se acepta también DNS, aunque no es estrictamente un protocolo → UDP, su principal función es la asociación entre nombres de dominios y direcciones IP.

b) El servidor es quién atiende los pedidos del cliente, iniciando primero y quedando a la espera de las conexiones establecidas desde el o los clientes.

c) Desde el punto de vista de la conexión TCP, es claro quién está esperando por la llegada de una conexión y quién la inicia, por lo que, a nivel de capa de transporte hay un cliente y un servidor. A nivel de capa de aplicación y protocolo P2P cualquiera de las dos entidades puede enviar los mismos mensajes a su contraparte. Es a nivel de capa de aplicación donde no se distingue cliente y servidor.

V3

a) Defina retardo de propagación y retardo de transmisión. Explique donde se dan y de que factores dependen.

b) Analice la frase: "en una red donde todos los enlaces tienen una velocidad de transmisión cercana a infinito, los *hosts* interconectados mediante conmutadores de paquetes en esta red nunca podrán percibir un retardo mayor a la suma de los retardos de propagación". Justifique.

a) Una vez que un bit ha entrado en el enlace, tiene que propagarse hasta el siguiente salto. El tiempo necesario para propagarse desde el principio del enlace hasta dicho salto es el retardo de propagación. El bit se propaga a la velocidad de propagación del enlace, que depende del medio físico del enlace (es decir, de que el medio sea fibra óptica, cable de cobre de par trenzado, etc.) y está comprendido en un rango que es igual, o ligeramente inferior, a la velocidad de la luz. El retardo de propagación es igual a  $d/s$ , donde  $d$  es la distancia entre el router emisor y el siguiente salto y  $s$  es la velocidad de propagación del enlace.

Suponiendo que los paquetes se transmiten de manera que el primero que llega es el primero que sale, como suele ser común en las redes de conmutación de paquetes, nuestro paquete solo puede ser transmitido después de que todos los paquetes que hayan llegado antes que él hayan sido transmitidos. Sea la longitud del paquete igual a  $L$  bits y la velocidad de transmisión del enlace del router emisor hasta el siguiente salto igual a  $R$  bits/segundo. El retardo de transmisión será igual a  $L/R$ . Este es el tiempo necesario para introducir (es decir, transmitir) todos los bits del paquete en el enlace. Normalmente, en la práctica, los retardos de transmisión son del orden de microsegundos a milisegundos.

b) No puede ser cierta, pues los delays percibidos pueden ser debido a los retardos de procesamiento y de cola que también puedan existir.

V4

a) Defina retardo de cola y de procesamiento. Explique donde se dan y de que factores dependen.

b) ¿De qué depende que en algunos casos el retardo de cola sea grande y que en otros sea insignificante?

## Redes de Computadoras

a) En la cola, el paquete experimenta un retardo de cola mientras espera para ser transmitido a través del enlace. La duración del retardo de cola para un determinado paquete dependerá del número de paquetes que hayan llegado antes a la cola y que estén esperando para ser transmitidos por el enlace. Si la cola está vacía y no se está transmitiendo ningún paquete actualmente, entonces el retardo de cola de nuestro paquete será cero. Por el contrario, si hay mucho tráfico y muchos paquetes están esperando también para ser transmitidos, el retardo de cola será grande. En la práctica, los retardos de cola pueden ser del orden de microsegundos a milisegundos.

El tiempo requerido para examinar la cabecera del paquete y determinar dónde hay que enviarlo es parte del retardo de procesamiento. El retardo de procesamiento puede también incluir otros factores, como el tiempo necesario para comprobar los errores de nivel de bit del paquete que se hayan producido al transmitir los bits del paquete desde el nodo situado aguas arriba hasta el router A. Los retardos de procesamiento en los routers de alta velocidad suelen ser del orden de los microsegundos o menores.

b) La respuesta a esta pregunta depende de la velocidad a la que llega el tráfico a la cola, de la velocidad de transmisión del enlace y de la naturaleza del tráfico entrante, es decir, de si el tráfico llega periódicamente o en ráfagas.

V5

Considere las capas del stack de protocolos de Internet (o TCP/IP).

- a) ¿Qué capas procesa un router? Justifique brevemente.
- b) ¿Qué capas procesa un switch de la capa de enlace? Justifique brevemente.
- c) ¿Qué capas procesa un host? Justifique brevemente.
- d) ¿Qué capas procesa un servidor web (HTTP)? Justifique brevemente.

a) física, enlace, red

b) física, enlace

c) física, enlace, red, transporte, aplicación

d) física, enlace, red, transporte, aplicación (si considera que con servidor web se refiere a la aplicación servidor y lo justifica adecuadamente, es válida la respuesta de que solo procesa la capa de aplicación)

### **Pregunta 2 (8 puntos)**

V1

Un usuario decide utilizar un servicio de correo electrónico vía web desde su computadora.

- a) ¿Qué protocolo/s de Capa de Aplicación y de Capa de Transporte utilizará su computadora para ello? Indique en cada caso el uso específico de cada protocolo en este contexto así como también los puertos involucrados.
- b) ¿Cómo cambia ello si, para el mismo servicio de correo electrónico, utiliza un programa cliente como por ejemplo Thunderbird? Indique en cada caso el uso específico de cada protocolo en este contexto así como también los puertos involucrados.

a) En Capa de Aplicación hará uso del protocolo HTTP y en Capa de Transporte, de TCP. Mediante HTTP (más formalmente, HTTP utilizando TLS o sea, HTTPS), a través del navegador oficiando como cliente de correo electrónico, podrá visualizar/organizar los mensajes de correo y, enviar y recibir. TCP mantendrá la conexión entre el cliente (navegador) y el servidor de correo

## Redes de Computadoras

(cuyo front end en este caso será un servidor web).

El puerto en el servidor será el 443 para HTTPS (el puerto 80 -HTTP- para estos usos se encuentra completamente desaconsejado).

b) En este caso, los protocolos involucrados son:

SMTP para el envío de correos

POP3 y/o IMAP para la recepción de correos

Los puertos “inseguros” involucrados son:

SMTP – 25

POP3 – 110

IMAP – 143

En todos los casos, existen sabores “seguros” de los protocolos mencionados, como ser:

SMTP “seguro” – 465, 587

POP3 “seguro” (POP3 sobre TLS) - 995

IMAP “seguro” (IMAP sobre TLS)– 993

### V2

Considere que un cliente desea conectarse a un servidor web y obtener una página HTML que incluye dos imágenes. Discuta la cantidad de solicitudes HTTP y consultas DNS necesarias para que la página HTML junto con las imágenes pueda ser desplegada en un navegador.

Primero se realiza una consulta DNS para resolver la dirección del servidor web el cual tiene el HTML deseado. Para obtener el HTML, establecida la conexión TCP a partir de la dirección obtenida con la consulta DNS, se realiza un pedido HTTP utilizando método GET por el recurso de la página HTML.

Una vez que se tiene el HTML, se debe realizar una solicitud HTTP por cada una de las imágenes. Estas imágenes están definidas en el HTML, y se debe resolver en caso de que el dominio donde estas imágenes están almacenadas no sean en el mismo dominio que el sitio original. Por lo que, si cada una de las imágenes se encuentran hospedadas en dominios diferentes se deben resolver estas direcciones (Consulta DNS) para luego realizar la solicitud GET correspondiente.

Por lo que se necesitan al menos 3 requests HTTP, y dependiendo de donde se encuentran las imágenes de 1 a 3 consultas DNS para poder mostrar todo el contenido de la página.

### V3

El administrador de un servidor web decide exponer el portal de su empresa en los puertos 225 (para http) y 241 (para https).

Analice técnicamente la decisión y los posibles impactos (positivos y negativos).

Técnicamente es viable pues la implementación del servicio no depende del número de puerto dónde escucha.

Impactos positivos:

Permite “ocultar” el puerto donde está escuchando el servicio, ya que no son los puertos reservados para ello: 80 y 443.

Impactos negativos:

En la letra se mencionan puertos “bien conocidos”. No es un diseño “prolijo” utilizar los reservados para dichos protocolos. Si se quiere “ocultar” (impacto positivo), habría que utilizar puertos altos (> 1024)

Si se seleccionan puertos  $\leq 1023$ , es posible que se termine seleccionando algún puerto que esté destinado para otro servicio, lo que puede determinar que se reciban mensajes “no esperados” e incluso, conflictos asociados al uso de un mismo puerto por parte de 2 servicios.

## Redes de Computadoras

Es una pésima decisión si es un portal web “tradicional”, pues los usuarios no podrán visitarlo, salvo que conozcan los puertos utilizados y explicitarlos en la URL.

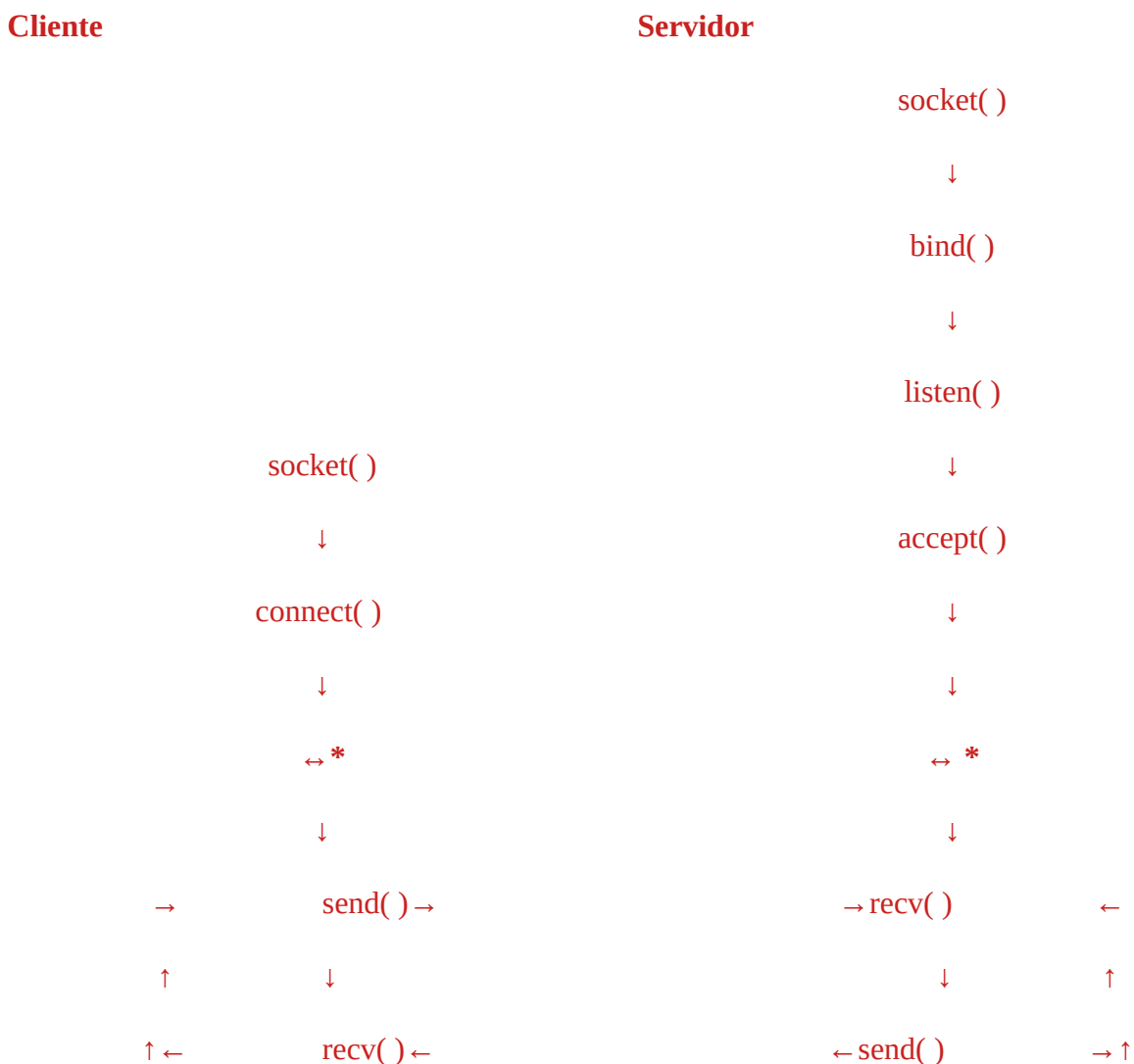
V4

a) A los efectos de establecer y utilizar por parte de una aplicación una conexión TCP, ordene en el tiempo, tanto del lado del servidor como del cliente, una secuencia típica esperada en cuanto a la invocación de los siguientes métodos y cómo se relacionan entre sí:

listen( )  
 socket( )  
 recv( )  
 connect( )  
 close( )  
 send( )  
 bind( )  
 accept( )

b) Tomando como referencia la respuesta a la parte a), indique en que momento debe ocurrir el establecimiento de conexión de 3 vías de TCP.

a)





## Redes de Computadoras

Ya que http es stateless, se utilizan las cookies para almacenar el historial de utilización de un cliente en un determinado sitio. Es una forma de modelar un estado entre el cliente y el servidor.

La tecnología de las cookies utiliza cuatro componentes:

(1) una línea de cabecera de la cookie en el mensaje de respuesta HTTP; (2) una línea de cabecera de la cookie en el mensaje de solicitud HTTP; (3) el archivo de cookies almacenado en el sistema terminal del usuario y gestionado por el navegador del usuario; y (4) una base de datos back-end en el sitio web.

en el cual puede ser utilizado para:

- Autorización
- Recomendaciones
- Carrito de compras
- Mantener la sesión del usuario

V4

2) Explique brevemente el principal uso de los siguientes métodos HTTP:

- a) GET
- b) POST
- c) HEAD
- d) PUT
- e) DELETE

a) La inmensa mayoría de los mensajes de solicitud HTTP utilizan el método GET. Este método se emplea cuando el navegador solicita un objeto, identificándose dicho objeto en el campo URL.

b) A menudo, un cliente HTTP utiliza el método POST cuando el usuario completa un formulario; por ejemplo, cuando especifica términos para realizar una búsqueda utilizando un motor de búsqueda. Con un mensaje POST, el usuario solicita también una página web al servidor, pero el contenido concreto de la misma dependerá de lo que el usuario haya escrito en los campos del formulario. Si el valor del campo de método es POST, entonces el cuerpo de la entidad contendrá lo que el usuario haya introducido en los campos del formulario. No podemos dejar de mencionar que una solicitud generada con un formulario no necesariamente utiliza el método POST. En su lugar, a menudo los formularios HTML emplean el método GET e incluyen los datos de entrada (especificados en los campos del formulario) en el URL solicitado.

c) El método HEAD es similar al método GET. Cuando un servidor recibe una solicitud con el método HEAD, responde con un mensaje HTTP, pero excluye el objeto solicitado. Los desarrolladores de aplicaciones a menudo utilizan el método HEAD para labores de depuración.

d) El método PUT suele utilizarse junto con herramientas de publicación web. Esto permite a un usuario cargar un objeto en una ruta específica (directorio) en un servidor web determinado. Las aplicaciones que necesitan cargar objetos en servidores web también emplean el método PUT.

e) El método DELETE permite a un usuario o a una aplicación borrar un objeto de un servidor web.

### **Pregunta 4 (8 puntos)**

V1

Explique los mecanismos que utiliza el protocolo RDT 3.0 para soportar transferencia fiable sobre un canal con pérdidas y errores.

El protocolo RDT3.0 utiliza los mecanismos de RDT2.2 y además de esto agrega un temporizador y retransmisiones.

## Redes de Computadoras

Por lo tanto, para el control de errores de bit en los datos utiliza la suma de comprobación la cual es agregada a los mensajes enviados por el emisor. El receptor utiliza esto para comprobar que el mensaje no haya sido modificado por la red.

El receptor utiliza mensajes de confirmación ACK para avisar al emisor de la correcta recepción de un mensaje. Los mensajes de confirmación incluyen un número de secuencia. Para avisar de un error en la recepción del mensaje se envía un ACK pero con el número de secuencia del mensaje anterior.

Para el caso de pérdida de paquetes el emisor utiliza un temporizador el cual se inicia luego del envío de cada mensaje. Si el temporizador se vence y no ha llegado un mensaje ACK para el paquete enviado, se reenvía el mensaje anterior.

Para evitar problemas con mensajes duplicados en el receptor, se utilizan números de secuencia 0 y 1 para diferenciar los mensajes de datos.

### V2

Considere los protocolos RDT2.2 (transferencia confiable en canal con errores de bit) y RDT3.0 (transferencia confiable en canal con errores y pérdida de paquetes).

- Explique en que casos el emisor decide realizar una retransmisión de datos.
- Explique que cuidados se deben tener en cuenta para que estas retransmisiones no generen inconvenientes en el receptor.
- Explique que cambios son necesarios con respecto a las retransmisiones si se pasa de un protocolo *stop-and-wait* a un protocolo con *pipelining*.

En el protocolo RDT2.2 la retransmisión se realiza cuando se recibe un ACK distinto del esperado. Esto indica que el mensaje llegó corrupto al emisor y por lo tanto debe ser reenviado. También se realiza un reenvío en el caso de recibir un mensaje corrupto.

En el protocolo RDT3.0 las retransmisión se realiza cuando se vence el temporizador del último mensaje enviado. Esto sugiere que el mensaje de datos o el ACK correspondiente se perdieron.

En las retransmisiones se debe tener cuidado de no generar duplicados del lado del receptor que no puedan ser detectados. Para esto se utilizan los números de secuencia para diferenciar los mensajes.

En un protocolo stop and wait con dos números de secuencia es suficiente ya que nunca se va a enviar un mismo numero de secuencia sin antes asegurarse que el otro llegó correctamente. Sin embargo, cuando se utiliza pipelining es necesario aumentar el rango de números de secuencia ya que van a existir varios mensajes enviados sin ser reconocidos. En este caso es importante que el rango de número de secuencias sea mayor que la cantidad de mensajes que se pueden enviar “a la vez” sin ser reconocidos para poder detectar los duplicados correctamente.

### V3

Describa los conceptos de multiplexación y demultiplexación y explique la diferencia en la multiplexación y demultiplexación entre un protocolo orientado a conexión y uno no orientado a conexión.

La multiplexación y demultiplexación consiste en poder diferenciar los mensajes salientes/entrantes a los distintos procesos ejecutando en un mismo host.

Para esto se utilizan los puertos origen y destino de los mensajes de capa de transporte. Sin embargo, existe una diferencia en como se utilizan estos puertos dependiendo si el protocolo es orientado o no a conexión.



## Redes de Computadoras

En el caso de un protocolo no orientado a conexión, la demultiplexación se realiza solo a partir del puerto destino (además de la IP destino que identifica al host). Es decir cada socket dentro de un mismo host es identificado por el puerto en donde está esperando los mensajes.

En el caso de un protocolo orientado a conexión, además de la IP y puerto destino, se utiliza también la IP y puerto origen para realizar la demultiplexación. Es decir, dos mensajes con igual IP y puerto destino, pueden estar dirigidos a distintos sockets si la IP y puerto origen son distintos.

V4

¿Es el protocolo UDP capaz de reaccionar a una red congestionada y tratar de adaptarse?

Si su respuesta fue negativa, explique que sería necesario agregar al protocolo UDP para que pueda implementar un mecanismo de control de congestión.

Si su respuesta fue positiva, explique como lo hace.

No, no es capaz.

Para que UDP pueda implementar control de congestión sería necesario contar con algún mecanismo que informe o infiera la existencia de congestión en la red.

Por ejemplo, TCP utiliza la pérdida de mensajes de confirmación ACK para inferir la congestión.

Por lo tanto, si se desea que UDP realice control de congestión una opción sería agregarle mensajes de confirmación o algún tipo de retroalimentación del receptor. También se podría implementar algún tipo de retroalimentación de parte de los nodos intermedios de la red.

Luego, pudiendo detectar la congestión, el siguiente paso sería agregar a UDP algún mecanismo que controle los datos que se envían pudiendo reducir o aumentar la velocidad a la que se envían los datos.

V5

Enumere y explique dos ventajas y dos desventajas que observaría una aplicación al usar UDP como protocolo de transporte en comparación con utilizar TCP.

Ventajas:

\* La aplicación tendría mayor control sobre los datos que se envían y cuando lo hacen dado que UDP simplemente empaqueta los datos y los envía por la red sin ningún proceso intermedio como la transferencia fiable o el control de congestión que pueden hacer que los datos se retrasen en el emisor antes de ser enviados. Por lo tanto, una ventaja concreta podría ser que el retardo sería menor.

\* No hay retardo en el establecimiento de conexión. Dado que UDP no realiza ningún establecimiento de conexión, no hay ningún proceso previo al envío de los datos que pueda agregar retardos.

\* Poca sobrecarga del host y overhead en los paquetes. Dado que UDP no debe mantener ningún tipo de estado en los sistemas terminales, no hay casi sobrecarga en los hosts por el uso de UDP. Además, el cabezal de los segmentos UDP es bastante pequeño en comparación con TCP, por lo que el overhead en el envío de los datos es poco.

\* Se puede enviar a la mayor tasa posible permitida por la red. Dado que UDP no implementa control de congestión, UDP no agrega ninguna limitante a la velocidad a la que se pueden transmitir los datos.

Desventajas:

\* No hay ninguna garantía de que los datos lleguen al receptor. Dado que UDP no implementa transferencia fiable, no es posible asegurar que los datos enviados hayan sido recibidos por el receptor.

\* Es posible saturar al receptor o incluso saturar la red y no darse cuenta del problema. Dado que UDP no implementa control de flujo ni control de congestión, es posible que se generen problemas

## Redes de Computadoras

como ser la pérdida de paquetes debido a que el receptor no puede procesar los datos tan rápido como son recibidos o que incluso se sature la capacidad de la red. En este caso, la aplicación experimentaría una pérdida considerable de paquetes.

**Pregunta 5 (8 puntos)**

V1

Considere los protocolos go-back-N (GBN) y selective-repeat (SR). Suponga que el tamaño del espacio de números de secuencia es S. ¿Cuál es la máxima ventana de emisor permitida para evitar confusiones del lado del receptor?

Ilustre con un ejemplo.

En el caso de GBN es suficiente con que la ventana tenga tamaño S-1. Esto se debe a que el receptor no tiene buffer y siempre espera por un único segmento.

En el caso de SR, el tamaño de la ventana tiene que ser menor o igual que la mitad del tamaño del espacio de números de secuencia. Esto se debe a que la ventana del emisor, de tamaño N, puede empezar, según los ACKs recibidos, en cualquier valor entre [k-N, k-1], siendo k el siguiente paquete a ser enviado. Si hubiera recibido ACKs hasta k-1, entonces la ventana sería [k, k+N-1]; por lo tanto "los números de secuencia ocupados" por el emisor son potencialmente 2N (k-N hasta k+N-1). Luego 2N debe ser menor o igual que S, el espacio de números de secuencia disponibles.

V2

Para un emisor TCP, describa la relación entre el UltimoByteEnviado, el UltimoByteReconocido, y las ventanas de control de congestión y flujo (VentCongestion, VentRecepcion).

TCP proporciona un servicio de control de flujo a sus aplicaciones para eliminar la posibilidad de que el emisor desborde el buffer del receptor.

Esto implica que la diferencia entre enviados y reconocidos debe ser menor o igual a la ventana de recepción.

Para el control de congestión, TCP agrega una variable adicional, la ventana de congestión VentCongestion, que impone una restricción adicional sobre la velocidad a la que el emisor TCP puede enviar tráfico a la red. Dado que los valores de bytes enviados y reconocidos son únicos, se debe imponer que:

$$\text{UltimoByteEnviado} - \text{UltimoByteReconocido} \leq \min\{\text{VentCongestion}, \text{VentRecepcion}\}$$

V3

Describa la utilidad del campo "Ventana de Recepción" en el cabezal del segmento TCP.

El campo "Ventana de Recepción" (de 16 bits) se utiliza para el control de flujo, e indica el número de bytes que un receptor está dispuesto a aceptar.

TCP proporciona un servicio de control de flujo a sus aplicaciones para eliminar la posibilidad de que el emisor desborde el buffer del receptor. El control de flujo es por tanto un servicio de adaptación de velocidades (adapta la velocidad a la que el emisor está transmitiendo frente a la velocidad a la que la aplicación receptora está leyendo).

El receptor define una variable llamada BufferRecepcion que es el tamaño del buffer de recepción y no debe ser desbordado. Luego, se define la variable VentRecepcion que representa el espacio disponible en el buffer:

$$\text{VentRecepcion} = \text{BufferRecepcion} - [\text{UltimoByteRecibido} - \text{UltimoByteLeido}]$$

## Redes de Computadoras

(inicialmente,  $VentRecepcion = BufferRecepcion$ )

Este valor, dinámico, es enviado por el receptor en cada segmento que envía al emisor.

Mientras tanto, el emisor controla dos variables,  $UltimoByteEnviado$  y  $UltimoByteReconocido$ , y por lo tanto se debe cumplir, para el emisor:

$UltimoByteEnviado - UltimoByteReconocido \leq VentRecepcion$

V4

Considere un protocolo de transporte confiable que utiliza la técnica go-back-N (GBN):

- ¿qué parámetro define cuántos segmentos pueden ser enviados y no reconocidos?
- Suponga que el valor del parámetro anterior es 5, y el emisor envió 5 segmentos aún no reconocidos, ¿el emisor puede enviar otro segmento?
- Ahora el emisor recibe confirmación de todos menos de el primero de esos 5, ¿ahora sí podría enviar un nuevo segmento?
- Todavía no fue recibida la confirmación del primero de los 5 segmentos y vence el timeout, ¿que sucede del lado del emisor?
- Ahora se reciben las confirmaciones de los segmentos 1, 2 y 5, ¿esto habilita a enviar datos nuevos? ¿cuantos segmentos podría enviar? ¿por qué?

- El parámetro es lo que esté definido como la ventana de transmisión del lado del emisor.
- No, dado que ya tiene el máximo de segmentos en viaje no reconocidos.
- Como GBN usa acks acumulativos, recibir un ack del 5 significa que todos los anteriores llegaron, de manera que la ventana se desliza 5 posiciones y en consecuencia se pueden enviar 5 nuevos segmentos.
- Esta situación no es posible, ya que la ventana se avanza y el timer se reseteó cuando se recibieron los ack de cualquiera de los segmentos siguientes.
- Esta situación tampoco es posible, porque el emisor ya deslizó la ventana y la letra no dice que se hayan enviado mas segmentos.

V5

- ¿Qué se intenta detectar y que acciones asociadas dispara el uso del RTT en TCP?
- Muestre y explique cómo se calcula el RTT para TCP
- Suponga que existiera una versión de TCP en la cual el RTT fuera una constante. Describa que inconvenientes presentaría tener una constante con un valor muy bajo, y en otro caso, una muy alta.

- El objetivo es medir el umbral actual aceptable para asumir que un paquete se ha perdido, lo que implicaría hacer una retransmisión (al menos del segmento perdido). También esto podría tener consecuencias en lo que respecta al control de congestión, ya que la detección de timeouts es un signo de congestión fuerte, lo que lleva a modificar otras variables de ese mecanismo (ventana,  $ssthreshhold$ , etc).
- Con el fin de estimar un RTT típico, es natural por tanto calcular algún tipo de promedio de los valores de  $RTTMuestra$ :  $RTTEstimado = (1-a) * RTTEstimado + a * RTTMuestra$ .  $RTTEstimado$  es una media ponderada de los valores de  $RTTMuestra$ . Esta media ponderada asigna un mayor peso a las muestras recientes que a las más antiguas. Esto es lógico, ya que las muestras más recientes reflejan mejor la congestión que existe actualmente en la red.
- Si fuera muy pequeño, entonces se generarían retransmisiones innecesarias, ocupando el canal y agregando overhead. Por el contrario si fuera muy grande, se generarían grandes demoras por ejemplo en redes hostiles en donde presentan tasas de pérdida grandes.

V6

## Redes de Computadoras

Explique cómo se comporta el receptor de un protocolo que utiliza go-back-N (GBN) con los paquetes fuera de orden.

Discuta ventajas y desventajas.

El receptor descarta los paquetes fuera de orden.

Si bien parece absurdo y un desperdicio de recursos descartar un paquete recibido correctamente (pero desordenado), hay que tener en cuenta que el receptor debe entregar los datos en orden a la capa superior. Si se espera el paquete  $n$ , pero llega el paquete  $n+1$ , el receptor podría guardarlo en el buffer de recepción, para entregarlo a la capa superior después de haber recibido y entregado el paquete  $n$ . Sin embargo, si se pierde el paquete  $n$ , tanto este como el paquete  $n+1$  serán retransmitidos como resultado de la regla de retransmisión del emisor. Por tanto, el receptor puede simplemente descartar el paquete  $n+1$ .

La ventaja de este método es la simplicidad del almacenamiento en el buffer del receptor (el receptor no necesita almacenar en el buffer ninguno de los paquetes entregados desordenados), y solo debe mantener es el número de secuencia del siguiente paquete en orden.