Redes de Datos 1
1er parcial – 2024
Solución

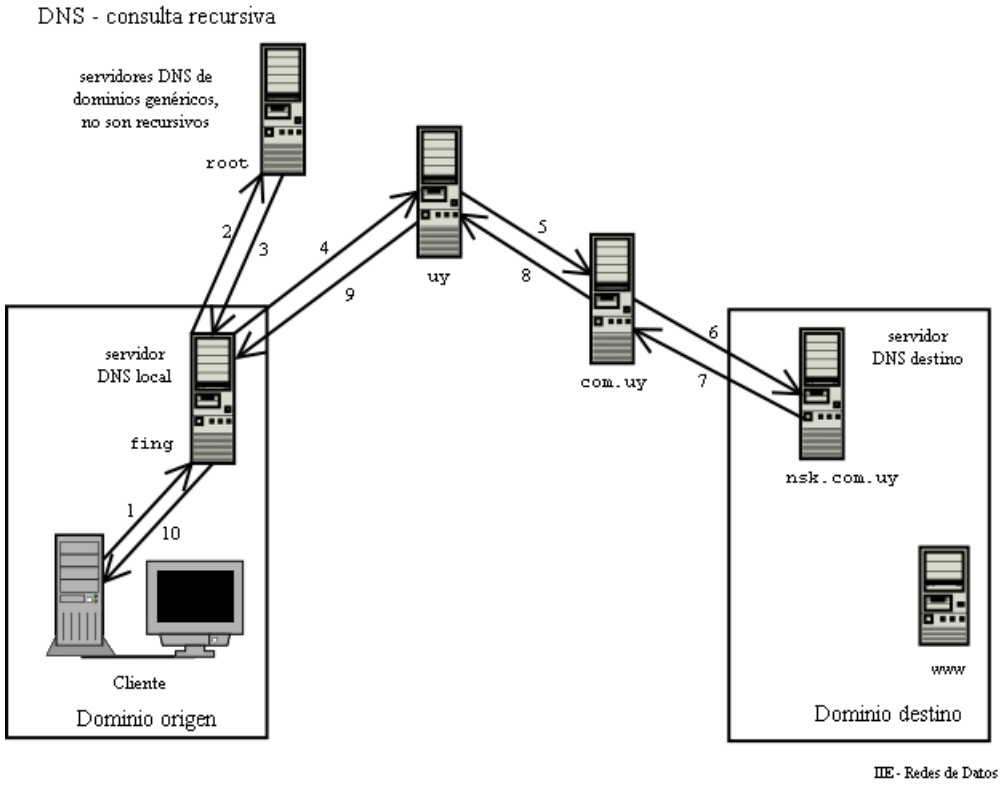
Esta es una posible solución a las preguntas planteadas. Por razones didácticas normalmente contiene bastante más información que la mínima necesaria para responder la pregunta de forma suficiente.

Pregunta 1 (11 puntos)

En el sistema de nombres de dominio (DNS):

a) ¿Cuál es la diferencia entre un servidor DNS recursivo y uno no recursivo? Ejemplifique para ambos casos.

Un servidor recursivo es aquel que, ante una consulta por un dominio del que no es autoritativo (ver parte "b"), realizará las consultas necesarias hasta llegar al registro solicitado. En el caso de ser no recursivo el servidor responderá con la información que tiene en caché si aplica, o con una indicación de cómo seguir la búsqueda en caso de corresponder (si el dominio del que buscamos el registro se encuentra en su rama y "aguas abajo", deberá indicar cómo seguir bajando por el árbol, de lo contrario responderá indicando que se debe comenzar por los servidores raíz).



Ejemplo: *En este caso el servidor DNS local (fing) y los servidores de los dominios "uy" y "com.uy" están permitiendo consultas recursivas, mientras el servidor "root" no está permitiendo la consulta recursiva. En caso de no responder recursivamente, la respuesta incluirá el registro NS del próximo servidor a consultar para encontrar la información deseada. Los servidores DNS locales en general son recursivos, mientras que los "root" no lo son.*

b) ¿Cuándo un servidor DNS es autoritativo para un dominio?

Decimos que un servidor es autoritativo para un dominio cuando es el encargado de gestionar los registros asociados a este dominio. El servidor autoritativo tiene la información "oficial" del dominio y sus administradores se encargan de crear y actualizar los registros necesarios. También se encarga de responder consultas sobre los registros del dominio (lo que se señaliza mediante banderas). Finalmente, tendrá la información de las subzonas delegadas a otros servidores (registros NS del nuevo dominio y los A y/o AAAA asociados a estos NS). Para que la información almacenada en el servidor autoritativo de un dominio pueda ser consultada, se debe poder llegar a dicho servidor siguiendo la cadena de delegaciones a partir de los root servers.

c) La Facultad de Ingeniería tiene un servidor web con dirección IP 164.73.32.20 que atiende el sitio www.fing.edu.uy. El servidor de nombres autoritativo para el dominio fing.edu.uy es 164.73.32.2.
Dado que un servidor web puede responder diferentes contenidos para diferentes nombres de sitios, se desea usar el nombre www.idm.uy para el sitio web de Ingeniería de Muestra (IdM) y alojar el sitio en el mismo servidor que el de www.fing.edu.uy. Para implementar esa configuración hay varias soluciones.

¿Cuál o cuáles de las siguientes acciones realizaría Ud.?

- Crearía el registro:

www.idm.uy	3600	IN	CNAME	www.fing.edu.uy

- Crearía los registros:

idm.uy	3600	IN	NS	ns.fing.edu.uy
ns.fing.edu.uy	3600	IN	A	164.73.32.2

- Crearía el registro:

www.idm.uy	3600	IN	A	164.73.32.20

Justifique qué acción o acciones entiende que resolverían el problema, indicando en qué servidor o servidores se configurarían.

Hay varias soluciones:

- *sin delegar la zona idm.uy*
  - *i. definir en el servidor autoritativo de la zona uy:*

www.idm.uy	*3600*	*IN*	*CNAME*	www.fing.edu.uy

  - *ii. o definir en el servidor autoritativo de la zona uy:*

www.idm.uy	*3600*	*IN*	*A*	*164.73.32.20*

- *delegando la zona idm.uy al servidor de fing.*
  - *i. definir en el servidor autoritativo de la zona uy:*

idm.uy	3600	IN	NS	ns.fing.edu.uy
ns.fing.edu.uy	*3600*	*IN*	*A*	*164.73.32.2*

  - *ii. y en el servidor autoritativo de idm.uy (ns.fing.edu.uy, 164.73.32.2) definir:*

www.idm.uy	*3600*	*IN*	*CNAME*	*www.fing.edu.uy*
o				
www.idm.uy	*3600*	*IN*	*A*	*164.73.32.20*

d) Desde una computadora de la Facultad de Química (pc1.fq.edu.uy) se desea acceder al sitio www.idm.uy. El equipo pc1.fq.edu.uy tiene configurado como servidor DNS recursivo la IP 164.73.160.194.
Explique la secuencia de consultas DNS requeridas para que pc1.fq.edu.uy pueda acceder a www.idm.uy. Asuma que no hay información relevante en ningún caché y que ninguno de los servidores intermedios acepta consultas recursivas. Suponga valores para las direcciones y nombres de los equipos intermedios que necesite.

Origen de la consulta/respuesta	Destino de la consulta/respuesta	Consulta realizada/respuesta recibida	Comentarios
pc1.fq.edu.uy	*164.73.160.194*	*Cuál es el registro A asociado a www.idm.uy?*	*Para conectarse con el sitio web necesitamos la dirección IP (registro A).* *La consulta se hace a la IP del DNS recursivo que pc1 tiene configurado*
164.73.160.194	*IP_root_server*	*Cuál es el registro A asociado a www.idm.uy?*	*El DNS local debe conocer las direcciones IP de los root servers. Elije uno para consultar.* *Siempre se consulta por el registro final que se necesita porque no se sabe quién lo puede tener (por autoritativo o por tenerlo en el caché)*
IP_root_server	*164.73.160.194*	*Registro NS de uy es ns.uy* *Registro A de ns.uy es IP_NSUY*	*Los root servers no responden consultas recursivas, por lo que brindará la información para poder continuar la búsqueda.* *En este caso además del registro NS es necesario el registro A del NS de uy (glue record)*
164.73.160.194	*IP_NSUY*	*Cuál es el registro A asociado a www.idm.uy?*	

A partir de este punto las consultas/respuestas dependen de la solución propuesta en la parte c).

*Si el administrador de uy **no delegó** el dominio idm.uy y creó un **registro A**:*

Origen de la consulta/respuesta	Destino de la consulta/respuesta	Consulta realizada/respuesta recibida	Comentarios
IP_NSUY	*164.73.160.194*	*Registro A de www.idm.uy es 164.73.32.20*	*El NSUY devuelve la respuesta solicitada.*
164.73.160.194	*pc1.fq.edu.uy*	*Registro A de www.idm.uy es 164.73.32.20*	*pc1 recibe el registro solicitado*

*Si el administrador de uy **no delegó** el dominio idm.uy y creó un **registro CNAME**:*

Origen de la consulta/respuesta	Destino de la consulta/respuesta	Consulta realizada/respuesta recibida	Comentarios
IP_NSUY	*164.73.160.194*	*Registro CNAME de www.idm.uy es www.fing.edu.uy*	*Con esta respuesta, el DNS recursivo, necesitará averiguar el registro A asociado a www.fing.edu.uy.* *Como ya conoce el NS de uy, puede comenzar esa nueva búsqueda a partir de él. (Nota 1)*
164.73.160.194	*IP_NSUY*	*Cuál es el registro A asociado a www.fing.edu.uy?*	
IP_NSUY	*164.73.160.194*	*Registro NS asociado a fing.edu.uy es ns.fing.edu.uy* *Registro A asociado a ns.fing.edu.uy es 164.73.32.2*	*NSUY devuelve la información para continuar la búsqueda (glue record)*
164.73.160.194	*164.73.32.2*	*Cuál es el registro A asociado a www.fing.edu.uy?*	
164.73.32.2	*164.73.160.194*	*Registro A asociado a www.fing.edu.uy es 164.73.32.20*	*El servidor autoritativo devuelve el registro consultado.*
164.73.160.194	*pc1.fq.edu.uy*	*Registro A de www.idm.uy es 164.73.32.20*	*pc1 recibe el registro solicitado*

Nota 1: El NS de uy podría enviar como información adicional el NS de edu.uy (y su IP) para evitar la doble consulta. En el caso concreto, además a uy y edu.uy los atiende el mismo servidor autoritativo, por lo que podría responder el NS de fing.edu.uy y su registro A y se ahorrarían los dos siguientes mensajes.

*Si el administrador de uy **delegó** el dominio idm.uy:*

Origen de la consulta/respuesta	***Destino de la consulta/respuesta***	***Consulta realizada/respuesta recibida***	***Comentarios***
IP_NSUY	*164.73.160.194*	*Registro NS asociado a idm.uy es ns.fing.edu.uy* *Registro A asociado a ns.fing.edu.uy es 164.73.32.2*	*Devuelve el NS y el A (glue record) para poder continuar la búsqueda.*
164.73.160.194	*164.73.32.2*	*Cuál es el registro A asociado a www.idm.uy?*	
164.73.32.2	*164.73.160.194*	*Registro A asociado a www.idm.uy es 164.73.32.20* *o* *Registro CNAME asociado a www.idm.uy es www.fing.edu.uy y registro A asociado a www.fing.edu.uy es 164.73.32.20*	*El servidor autoritativo devuelve el registro consultado.*
164.73.160.194	*pc1.fq.edu.uy*	*Registro A de www.idm.uy es 164.73.32.20* *o* *Registro CNAME asociado a www.idm.uy es www.fing.edu.uy y registro A asociado a www.fing.edu.uy es 164.73.32.20*	*pc1 recibe el registro solicitado*

Pregunta 2 (6 puntos)

Para un proyecto vinculado al agro se requiere tomar medidas de humedad del suelo en varios puntos de un campo. Para esto se diseñan pequeños equipos alimentados por baterías que se “siembran” en el campo y que mediante una conexión inalámbrica permiten recoger medidas de humedad y transmitirlas a un servidor para su procesamiento.
Los equipos son accesibles remotamente para actualizar su programación o su configuración.
El sistema de recolección de datos de humedad, requiere que los equipos envíen cada una hora el valor de la humedad registrado (representado con un número de 16 bits).

a) ¿Cuál de los protocolos de transporte vistos en el curso (TCP y UDP) utilizaría para la función de actualización de software y configuración? Justifique su respuesta indicando qué características del protocolo elegido le parecen relevantes para esta función.

Sería mejor TCP, en primer lugar porque garantiza el envío de la totalidad de los bytes, sin repetidos ni duplicados ni en desorden, de modo que se entregue a la capa de aplicación toda la información tal cual fue enviada. Esto es importante por la tarea que se busca realizar, un error (ejemplo falta de un segmento) podría por el contrario no permitir que el software se actualice o configure correctamente. En segundo lugar, TCP realiza control de congestión y control de flujo, ambos útiles en este contexto, para no sobrecargar la red y no enviar al receptor más datos de los que puede recibir, respectivamente. Tratándose de un sensor, es probable que tenga poco buffer de recepción, siendo importante por lo tanto el control de flujo.
TCP además es orientado a conexión, durante la conexión se intercambian valores importantes para poder asumir las funciones comentadas, tanto relativos a parámetros para control de congestión y de flujo como respecto a la garantía de recepción en orden y sin duplicados, gracias a los números de secuencia y reconocimiento. Por último, el establecimiento de conexión parece interesante también para este caso, ya que la actualización de software probablemente involucre el envío de un volumen importante de datos, es razonable por lo tanto asegurarse que el receptor está escuchando antes de enviar los datos a la red.

b) ¿Cuál de los protocolos de transporte vistos en el curso (TCP y UDP) utilizaría para la función de envío de datos de humedad? Justifique su respuesta indicando qué características del protocolo elegido le parecen relevantes para esta función.

Al ser un envío de datos periódico y de solamente 2 bytes convendría usar UDP, teniendo en cuenta además que si se pierde una medida no sería demasiado relevante (las medidas se realizan recurrentemente, hay varios sensores enviando datos que serán similares, se podría además incorporar

otras soluciones como enviar datos cada 10 o 15 minutos por si alguno se pierde). Si se usara TCP se estaría agregando un overhead innecesario debido al establecimiento y fin de conexión, y a los encabezados de TCP que son más grandes que los de UDP. Como consecuencia, con TCP demoraría más la transmisión, se consumiría más energía y por lo tanto batería, que asumimos puede ser una limitante en este tipo de dispositivos. Por otro lado, los datos son únicamente 2 bytes, no es de esperar que haya problemas de congestión ni de atorar al servidor, ni de tener segmentos que lleguen en desorden (los 2 bytes típicamente serán enviados en un solo segmento). Por lo tanto, el hecho de que UDP no ofrezca estas garantías no debería ser un problema. Todas estas razones justifican la elección de UDP frente a TCP.

Pregunta 3 (6 puntos)

a) Explique las diferencias entre la arquitectura de red de circuitos virtuales y la de datagramas en cuanto a la implementación del plano de datos. ¿Qué tablas se usan en cada caso? ¿Cómo se usan? ¿Qué información contienen? ¿Cómo se obtiene dicha información? ¿Qué ventajas tiene cada arquitectura?
Nota: No es necesario explicar detalladamente el mecanismo de búsqueda en IP.

El plano de datos en capa de red es el encargado de encaminar los paquetes de origen a destino. Para ello utiliza la información obtenida por el plano de control, representado por las tablas de encaminamiento (forwarding).
En el caso de la capa de red basada en circuitos virtuales (CV) los paquetes siguen un circuito virtual, que es una conexión lógica entre el origen y el destino. Los circuitos virtuales son establecidos por el plano de control PREVIO AL ENVIO DE LOS DATOS, y se identifican mediante identificadores de circuito virtual. Dichos identificadores son locales a cada "tramo" de la red, para simplificar la elección de los mismos, por lo que en cada enrutador precisaremos relacionar el identificador de CV con el que se recibe el paquete (acordado con el enrutador anterior), con el identificador correspondiente al tramo "de salida" (el que se debe utilizar para enviar el paquete al siguiente enrutador). La relación entre los identificadores de CV de entrada y los de salida para los circuitos establecidos se resume en la tabla de circuitos virtuales (es decir la tabla de encaminamiento en CV), y es obtenida por el plano de control al momento de establecer cada circuito, ya sea de forma estática (alguien o alguna entidad configura los CV), o mediante la utilización de protocolos de enrutamiento dinámico. Los detalles variarán de protocolo a protocolo, pero en esencia para cada identificador de CV de entrada en uso (incluyendo posiblemente la interfaz de entrada), la tabla de CV nos indica la relación con el identificador de salida e interfaz de salida (o posiblemente próximo salto). Cada paquete llegará identificado por un identificador de CV, y el encaminamiento se realizará buscando en la tabla este identificador de CV de entrada y utilizando los datos de salida para modificar el paquete y reenviarlo al próximo salto. Un posible ejemplo de tabla de CV se observa en la siguiente tabla:

Entrada		***Salida***	
Interfaz	*Id CV*	*Interfaz*	*Id CV*
i0	*20*	*i1*	*53*
i1	*14*	*i2*	*12*
i2	*17*	*i4*	*33*

Por otro lado en las redes de datagramas, como por ejemplo la red IP, el plano de datos no requiere la configuración previa de un camino. Cada paquete, o datagrama, es independiente y lleva la dirección completa del destino, por lo que se encamina utilizando el mejor camino a ese destino de forma independiente a los otros paquetes. La tabla de encaminamiento o forwarding en el plano de datos contiene la información de los destinos alcanzables (en el caso de IP como pares Red/Máscara), y el próximo salto, es decir la IP (o en algunos casos la interfaz) a la que debe ser enviado el paquete para continuar su camino al destino. Para realizar el encaminamiento se busca para CADA paquete la entrada más específica de la tabla que incluya la dirección de destino, y se utilizan los datos de próximo salto (e interfaz de salida) para reenviar el paquete. La tabla se encuentra ordenada con las redes más específicas (con máscara más larga) primero.
La información de la tabla de encaminamiento la obtiene el plano de control, utilizando ya sea información estática (configurada), información de redes directamente conectadas, y/o información obtenida dinámicamente mediante la utilización de protocolos y algoritmos de enrutamiento dinámico. El detalle de las tablas de encaminamiento para IP y el algoritmo de búsqueda lo pueden ver en el material del teórico.

Algunas ventajas de la arquitectura de circuitos virtuales son:

- *Es más sencillo implementar calidad de servicio e ingeniería de tráfico en una arquitectura de circuitos virtuales, ya que el camino de los paquetes está establecido, lo cual hace que los retardos*

sean más predecibles y además pueden reservarse recursos que garanticen el tratamiento de los paquetes que circulan por ese circuito
- *La búsqueda en la tabla de encaminamiento es más sencilla, llevando a un en encaminamiento más rápido y barato.*

Algunas desventajas de la arquitectura de circuitos virtuales son:
- *Los caminos establecidos se almacenan en las tablas de circuitos virtuales de cada enrutador, por lo que si el enrutador se cae, perderá esas tablas y por tanto todos los circuitos que pasaban por él se caen. Para volver a cursar tráfico será necesario re-establecer los circuitos nuevamente.*
- *Es necesario establecer el circuito antes de la transferencia de los datos, por lo que para transferir pocos datos a un destino el overhead puede ser considerable*

Algunas ventajas de la arquitectura basada en datagramas:
- *Sencillez en la red, los enrutadores no precisan mantener estado de los flujos establecidos. Esto además las hace más escalables.*
- *Facilidad de recuperación ante falla de enrutadores y enlaces*

Algunas desventajas de la arquitectura basada en datagramas:
- *Es más difícil ofrecer calidad de servicio y realizar ingeniería de tráfico*
- *Es más difícil hacer control de congestión y por eso en TCP/IP, el control de congestión se hace en capa de transporte (TCP).*

Pregunta 4 (10 puntos)

a) ¿Qué se entiende por congestión en una red? Explique cuáles son los síntomas que se observan en la red cuando existe congestión y cuáles pueden ser las causas principales.

La congestión ocurre cuando se sobrepasa la capacidad de algún elemento en la red, típicamente de algún enlace o la CPU/memoria de algún enrutador de la red por donde transitan los paquetes IP que llevan los segmentos TCP. En esos casos, existen más paquetes de los que se puede procesar o que quieren utilizar un mismo enlace, por lo que se utiliza una cola donde almacenarlos temporalmente hasta poder procesarlos. Cuando los paquetes tienen que esperar en esas colas, tardarán más tiempo en llegar a destino e incluso podrían ser descartados en algún enrutador de la red si se supera la capacidad de la cola de atención.

Los paquetes terminan acumulando las demoras ocurridas en cada salto de red a lo largo se su trayecto y los extremos no conocen exactamente dónde ocurre la congestión, solo se percibe el efecto acumulado. La noción de demora acumulada de los paquetes es más extensa, comprende, por ejemplo, demoras en dar ACK del receptor, demora de análisis en Firewall u otros fenómenos que puede darse en las capas inferiores.

A continuación se presenta un cuadro a modo de resumen de los síntomas de congestión que percibe TCP, la capa de Aplicación y cuales son las causas "usuales" o principales.

Síntoma en TCP	***Síntoma en Aplicación***	***Causas***
Demoras en la llegada de ACK	*Pequeña variación de velocidad o tasa de intercambio y refresco de pantalla. Podría ser imperceptible (o poco perceptible) desde la Aplicación.*	*Los paquetes IP (que llevan segmentos) se encolan en alguno de los enrutadores intermedios. La demora adicional se debe a la espera hasta llegar al primer lugar.*
Re-envío de algunos segmentos	*Se percibe una variación en la velocidad de descarga o lentitud de refresco de pantalla.*	*Expira el temporizador de retransmisión de algunos segmentos, el paquete que lleva el segmento original acumula mucha demora o directamente no encontró espacio en alguna de las colas de procesamiento y se descartó.*
Re-envío de los mismos segmentos varias veces	*Se percibe como problemas de conexión, pantallas que nunca terminan de refrescar, baja o nula tasa de intercambio de datos.*	*La mayoría de los paquetes enviados no encuentra espacio en las colas de procesamiento de algunos de los enrutadores intermedios, se descartan. Los re-envíos de TCP vuelven a generar paquetes que empeoran la congestión y son descartados nuevamente.*

b) ¿Qué se entiende por control de flujo? Explique cuáles son los síntomas que se observan cuando no se realiza un control de flujo adecuado.

El control de flujo se entiende como la capacidad de un transmisor de adaptar el envío de datos a la capacidad del receptor. Esta definición es muy generalista, pero contempla el escenario por el cual lo utiliza TCP. Se requiere un almacenamiento temporal donde guardar los datos que han sido reconocidos por TCP, pero aún no fueron leídos por la aplicación. La forma que tiene de adaptar la velocidad de transmisión TCP es conociendo la cantidad de bytes disponible en el receptor para este almacenamiento intermedio.

Cuando no se realiza control de flujo el síntoma por excelencia es que se envía información e independientemente de que haya o no congestión, el receptor no termina recibiendo la información y esta se debe volver a enviar. Se asocia a una mala utilización de los recursos de red, la capacidad que se utilizó en un envío ***inútil****, nadie la pudo utilizar. Desde el punto de vista de la aplicación los síntomas pueden ser similares a los que mencionamos en el caso de congestión. Además, estos envíos de información que se descartan en el destino, contribuyen a aumentar la cantidad de paquetes que circulan por la red y aumentan por tanto el "piso" de congestión.*

c) ¿Qué información utiliza TCP y cómo la utiliza para realizar el control de congestión? No es necesario explicar detalladamente el mecanismo de control de congestión de TCP.

En pocas palabras TCP (formalmente el transmisor TCP) obtiene la información del estado de la red observando la llegada de reconocimientos (ACK) a los segmentos enviados.

Información	***Intuición***	***Construye o Acciones***
Medida de tiempo de envío de un segmento y llegada de su ACK (RTT)	*Captura la demora que hay en el trayecto, y en particular las demoras por avanzar en las colas de servicio en los enrutadores.*	*Construyo un estimador de* ***RTT*** *y* ***desviación estándar de RTT****, para luego tener un estimador de* ***timeout*** *o timer de retransmisión.*
Llegada de ACK antes de que expire el timer de retransmisión.	*No hay congestión puedo incrementar cwnd, lo cual permite incrementar la tasa de envío. (*)*	*Actualizo el estimador de capacidad de la red* ***cwnd*** *(llamado ventana de congestión) y el estimador de* ***RTT, desviación estándar de RTT y timeout****.*
No llegada del ACK antes que expire el timer de retransmisión	*Congestión en la red, debo disminuir cwnd, lo cual disminuye la tasa de transmisión en la conexión TCP.*	*Disminuyo cwnd y retransmito los bytes para los cuales haya expirado el timer y estén dentro de la nueva ventana. Modifico a la baja el valor de* ***ssthresh*** *que refleja el umbral a partir del cual el crecimiento de cwnd es lineal. En este caso, el estimador de RTT deja de ser una referencia para decidir el timeout, se utiliza una estrategia para incrementar el timer de expiración, por ejemplo duplicar el último valor hasta los 120 segundos y luego repetir n veces el re-envío.*

TCP controla la tasa de envío considerando la ventana de transmisión (cantidad máxima de bytes a enviar sin esperar a los reconocimientos) W_{TX} <= cwnd, de está forma la tasa máxima es ***W_{TX}/RTT*** *(bps). Esto permite que si hay congestión y disminuyo cwnd, se envíe menos información a la red, así como si no ocurre congestión puedo incrementar la tasa máxima de envío aumentando cwnd.*

Además se requieren algunos parámetros adicionales, como por ejemplo, el valor de cwnd y ssthresh inicial, cómo incrementar la ventana cuando no hay congestión o cómo decrementar la ventana cuando ocurre congestión, cómo ajustar los temporizadores de retransmisión cuando ocurrió congestión, cuántas veces retransmitir un segmento hasta asumir pérdida de conexión, etc.

Nota(*): *Podría haber una leve congestión si las medidas puntuales de RTT aumentan, si bien TCP incrementa el valor de cwnd, en parte se compensan en que se envía W_{TX} cada RTT.*

d) ¿Qué información utiliza TCP y cómo la utiliza para realizar el control de flujo?

En pocas palabras TCP (el transmisor TCP) utiliza la información de realimentación que le envía el receptor mediante el campo del encabezado TCP ***WindowSize****. El receptor informa la capacidad disponible en bytes para almacenar nuevos datos en el buffer del receptor, que no es otra cosa que un espacio de memoria temporal donde se almacenan los datos recibidos y reconocidos pero no leídos por la aplicación del lado del receptor. De esta forma implementar el control de flujo es sencillo, alcanza que el transmisor procure que su ventana de transmisión* W_{TX} <= ***WindowSize*** *reportado por el receptor.*

Cada vez que llegan segmentos con información de aplicación al receptor, este debe de enviar un ACK lo antes posible y en ese segmento, el mismo header TCP que lleva la información de reconocimiento lleva también el nuevo valor de WindowSize. De esta forma si la aplicación lee de forma espaciada porque hay mucha utilización de la CPU por otros procesos, rápidamente se consume estos buffers y el transmisor deja de enviar, evitando saturar al receptor (y evitando re-enviar los segmentos). Esto se pudo observar durante el laboratorio de TCP mediante el ***zero Window*** *que reportaba WireShark. El transmisor TCP debe poder recuperarse del escenario de pérdida de un segmento con actualización del campo WindowSize, cuando el último valor fue cero y hay información a enviar.*

Cuando el transmisor envía datos, ajusta la capacidad remanente del receptor, por ejemplo, si el transmisor envió 3 MSS, sabe que la capacidad remanente del receptor es WindowsSize – 3 MSS (que debe ser mayor o igual a cero), es decir que puede enviar hasta dicha cantidad de bytes sin esperar los ACK y la actualización de ventana del receptor. Si el transmisor recibe un ACK antes de finalizar los envíos, actualiza el valor de W_{TX} *con la información más actualizada que proporcionó el receptor, y considerando (restando) los bytes pendientes de reconocimiento.*

No debe confundirse el control de flujo con otras políticas de control de envío que pueden implementar el transmisor o receptor, como por ejemplo el receptor puede realizar políticas adicionales para evitar el síndrome de la ventana tonta (no reportando actualización de WindowSize hasta tener una capacidad razonable) o el transmisor implementar políticas para maximizar los datos de aplicación enviados (piggybacking o Nagle). En general estás políticas intentan evitar los escenarios donde se hace una pobre utilización de recursos de la red (pocos bytes de datos de aplicación en cada envío).
Nota: *Recordar que es posible negociar un factor de escala al comienzo de la conexión TCP, de forma de poder utilizar valores mayores a 64 KB de buffer, se debe considerar multiplicar el factor de escala a lo recibido en el campo WindowSize.*

Nota2: *Un transmisor TCP debe considerar la información de espacio disponible en buffer del receptor así como la capacidad de la red, esto se contempla eligiendo que la ventana de transmisor sea* W_{TX} <= ***min { WindowSize, cwnd}*** *de forma de contemplar ambas restricciones.*

Pregunta 5 (10 puntos)

a) Explique las funciones del plano de control o plano de ruteo en la capa de red.

El plano de control de la capa de red se encarga de encontrar los mejores caminos en la red para llegar desde un origen a un destino dados.
Se considera la topología de la red como un grafo definido por los entrutadores o conmutadores y los enlaces que los interconectan. Los enlaces pueden tener diferentes pesos o costos y se trata de encontrar el camino de menor costo entre cada pareja origen/destino.
Las métricas para esos costos pueden ser de diferente naturaleza: retardos, ancho de banda, distancia física, costo de uso del enlace, etc.
Esta función se puede realizar cada vez que se detecta un cambio en la red (se cae un enlace o nodo, se conecta un nuevo enlace o nodo) y también se puede recalcular periódicamente si hay mejores caminos que los que se están usando.

b) Explique cómo se vincula la función de ruteo con la función de encaminamiento o forwarding.

Los caminos definidos por la función de ruteo, deben plasmarse en tablas de forwarding que le dicen a cada enrutador cómo debe encaminar cada paquete.
Para que los paquetes sigan el camino definido por el plano de control, cada enrutador debe contribuir encaminando cada paquete que le llega.
La función de encaminamiento se realiza entonces cada vez que llega un paquete al equipo.

c) Explique y compare la implementación del plano de control de forma estática o dinámica.

Los mejores caminos se pueden calcular una vez y realizar la configuración de las tablas de forwarding necesarias para implementarlos de forma manual. Esto es lo que se conoce como enrutamiento estático. Ese enrutamiento se puede revisar periódicamente y hacer los cambios en las tablas si corresponde.
Esta modalidad de configuración tiene la ventaja de ser muy estable, pero ante una falla de la red (y por tanto un cambio de la topología), es necesario recalcular los mejores caminos y volver a reconfigurar manualmente las tablas de forwarding. El tiempo necesario para detectar, recalcular y reconfigurar hace que la red pueda tener problemas temporales por destinos que no son alcanzables hasta que se reconfiguren las nuevas tablas.
Por otro lado la modalidad dinámica permite que se detecten automáticamente los cambios de la red (mediante protocolos de ruteo dinámico), que se dispare el recálculo de los mejores caminos (usando algoritmos de ruteo) y que se reflejen automáticamente los cambios necesarios en las tablas de forwarding de los equipos. Esta modalidad permite una adaptación automática de los caminos frente a los cambios de la red (sin intervención humana) pero requiere más capacidad de procesamiento y memoria en los enrutadores.

d) Explique para qué se usan algoritmos de ruteo. No se requiere explicar detalladamente ningún algoritmo en particular.

Como se mencionó anteriormente, dada una topología de red compuesta por nodos y enlaces (con costos), se debe aplicar algún algoritmo para encontrar los mejores caminos entre cada pareja de origen-destino.
Como ejemplos tenemos el algoritmo de Dijkstra y los algoritmos de vector distancia vistos en clase.
En la arquitectura clásica, los enrutadores recogen información de la red y ejecutan un algoritmo que determina los mejores caminos, que se traducen en entradas en sus propias tablas de forwarding.
En la arquitectura SDN (Software Defined Networking), el controlador conoce la topología de la red, corre el algoritmo y mediante un protocolo específico dialoga con los enrutadores para definirles la tabla de forwarding que deben usar.

e) Explique para qué se usan protocolos de ruteo.

Como se mencionó anteriormente para conocer el estado de la red, los equipos deben intercambiar información (qué destinos conocen, con qué otros equipos están conectados, etc.). Para intercambiar información se necesita un protocolo que defina el formato de la información y las reglas de intercambio necesarias.