Redes de Datos 1
1er parcial – 2019
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****.*

Pregunta 1 (8 puntos)

Entre dos equipos A y B distantes en la red, se mide el retardo de ida y vuelta enviando paquetes de eco y esperando la respuesta a los mismos.

Se considera que:

- los paquetes a la ida y a la vuelta pasan por los mismos enlaces y equipos.
- entre A y B hay 1.000 km de fibra óptica y se considera que la velocidad de propagación de la luz en ese medio es de 200.000 km/s.
- en el camino entre A y B hay 25 equipos que conmutan paquetes y cada uno impone un retardo de procesamiento de 1 µs (micro segundo).
- en el camino entre A y B hay 10 enlaces de 10 Gbps y 16 de 100 Gbps.
- cuando un paquete debe salir por cada uno de los enlaces de 10 Gbps se encuentra en promedio 10 paquetes de 10.000 bits en cola antes que él. En el resto de los enlaces se asume despreciable el encolamiento.
- el tiempo de procesamiento en los equipos de los extremos es despreciable.
- los paquetes de eco y su respuesta son de 1.000 bits.

Justifique todos los cálculos realizados (se recomienda hacer los cálculos en mili o micro segundos).

En base a los datos indicados:

a) Calcule el tiempo de propagación en la fibra óptica.

De acuerdo a la información brindada, hay 1000 Km de fibra óptica entre A y B. Si contamos el camino de ida y vuelta, son 2000 Km, que de acuerdo a la velocidad de propagación brindada tomarán:

$T_p = L/V = 2 \times 1000 \text{ km}/(200.000 \text{ km/s}) = 10000 \text{ µs}$ *(5000 µs si se considera solo el camino de ida)*

b) Calcule el tiempo de serialización de un paquete de 1.000 bits en un enlace de 10 Gbps y en un enlace de 100 Gbps.

El tiempo de serialización es el tiempo que toma "inyectar" todos los bits de una trama de largo L bits en un enlace y se calcula a partir de la velocidad de transmisión R como:

Para el enlace de 10 Gbps, $T_{s10} = L/R = 1000 \text{ bits} / (10 \times 10^9 \text{ bits/s}) = 10^{-7} \text{s} = 0{,}1 \text{ µs}$

Para el enlace de 100 Gbps, $T_{s100} = L/R = 1000 \text{ bits} / (100 \times 10^9 \text{ bits/s}) = 10^{-8} \text{s} = 0{,}01 \text{ µs}$

c) Calcule el tiempo que debe esperar un paquete para ser enviado por un enlace de 10 Gbps si hay 10 paquetes de 10.000 bits encolados delante de él. Asuma que el único tiempo que debe esperar es el tiempo de serialización de los paquetes que deben ser enviados antes.

Utilizando el mismo cálculo de la parte anterior para el tiempo de serialización de los 10 paquetes de 10.000 bits, obtenemos:

$T_{e10} = 10 \times 10.000 \text{ bits} / (10 \times 10^9 \text{ bits/s}) = 10^{-5} \text{ s} = 10 \text{ µs}$

d) ¿Cuál es el tiempo total de ida y vuelta esperado en la medida? ¿Cuál es la principal componente del retardo? Indique el valor y la forma de cálculo de cada uno de los componentes del retardo.

El tiempo de ida y vuelta, tomando en consideración solamente los retardos indicados en la letra, incluye 4 componentes:

- El retardo de propagación en la suma de los enlaces del camino de ida y vuelta entre A y B (T_p)
$T_p = 10000\ \mu s$
- El retardo de serialización del paquete de eco y su respuesta en cada enlace (10 enlaces de 10 Gbps y 16 de 100 Gbps en cada sentido):
$T_s = 20 \times T_{s10} + 32 \times T_{s100} = 2{,}32\ \mu s$
- El retardo de procesamiento (1 μs en cada equipo por cada paquete). Son 25 equipos en cada sentido:
$T_{pr} = 2 \times 25 \times 1\ \mu s = 50\ \mu s$
- El retardo de encolamiento en los enlaces indicados en la letra (10 enlaces de 10 Gbps en cada sentido)
$T_e = 2 \times 10 \times T_{e10} = 200\ \mu s$

El retardo total esperado es la suma de los retardos calculados anteriormente
$T = T_p + T_s + T_{pr} + T_e = 10.252{,}32\ \mu s \approx 10{,}25\ ms$

Claramente la principal componente del retardo está dada por el tiempo de propagación en la fibra óptica. Aunque la velocidad de propagación es una fracción importante de la velocidad de la luz en el vacío, la gran distancia entre A y B hace que las demás componentes sean órdenes de magnitud menores que el retardo de propagación. En este caso una buena aproximación del tiempo de ida y vuelta es el tiempo de propagación

e) ¿Cómo cambia su respuesta si, con los mismos parámetros, la distancia de la fibra fuera de 20 km?

Si se mantienen todos los parámetros excepto la distancia entre A y B, solamente cambia la componente T_p del tiempo total.
En este caso tendremos que:
$T_{p20k} = 2 \times 20\ km/(200.000\ km/s) = 200\ \mu s$

Por lo tanto el retardo de ida y vuelta esperado es:
$T = T_{p20k} + T_s + T_{pr} + T_e = 452{,}32\ 50\ \mu s$

En este caso la componente debida al tiempo de propagación es del mismo orden que el retardo de encolamiento (iguales) y que el retardo de procesamiento (4 veces mayor). Es interesante observar cómo a medida que la longitud de los enlaces se hace más corta, el peso relativo de las otras componentes crece de forma importante. Podemos darnos cuenta que en distancias muy cortas la velocidad de los enlaces, la velocidad de procesamiento, y la existencia o no de congestión en los enlaces ganan una importancia relativa mayor respecto a la velocidad de propagación.

Pregunta 2 (5 puntos)

Cuando un servidor de correo recibe un mensaje desde otro servidor (utilizando el protocolo SMTP) normalmente realiza una serie de verificaciones para alejar la probabilidad de que se trate de un correo no deseado o SPAM. Algunos de los chequeos se realizan usando el sistema de nombres de dominio, DNS.

a) Un chequeo posible es verificar que la dirección IP desde la que llega la conexión tenga asociada en el DNS un nombre y que a su vez ese nombre tenga asociada la misma dirección IP. Supongamos que un servidor recibe un mensaje desde la dirección IP 190.32.1.5. ¿Cuál o cuáles consultas DNS son necesarias para realizar la verificación descripta? Indique el tipo de registro DNS involucrado en cada caso.

Dado que el mensaje llega desde la IP 190.32.1.5, la primer consulta que debemos realizar es una búsqueda DNS inversa para saber si dicha IP tiene un nombre de dominio asociado. Se debe consultar al servidor DNS local por el registro PTR asociado a la etiqueta 5.1.32.190.in-addr.arpa. construida a partir de la dirección IP 190.32.1.5.

Asumiendo que el servidor DNS local es un servidor DNS recursivo, saldrá a preguntar por el registro PTR comenzando la búsqueda por uno de los root servers, siguiendo la búsqueda por el TLD .arpa de la rama reversa del árbol de nombres de dominios. Asumiendo que el registro PTR asociado a esta dirección existe, podría tener los siguientes campos:

5.1.32.190.in-addr.arpa. 3600 IN PTR nosoy.spam.com.uy.

Para verificar si ese nombre de dominio efectivamente está asociado a la dirección IP, debemos consultar por el registro A asociado a nosoy.spam.com.uy. Asumiendo que existe, este podría tener la siguiente forma:

nosoy.spam.com.uy 800 IN A 190.32.1.5

Si la IP obtenida en la segunda consulta coincide con la IP desde la que se recibió la conexión, se aumentaría la reputación de la misma al momento de clasificarla como potencial SPAM. Si son diferentes, entonces disminuiría su reputación.

b) Otro chequeo, consiste en verificar si es posible responder al originador del correo. Supongamos que el originador del correo es spammer@spamgate.com.au, ¿Qué consulta o consultas DNS serían necesarias para verificar si es posible responder a ese originador?

En este caso se debe consultar por los servidores de correo asociados al dominio spamgate.com.au, es decir, por el registro MX asociado a este nombre de dominio. Asumiendo que existe, el registro podría tener la siguiente forma:

spamgate.com.au 86400 IN MX 1 mail.spamgate.com.au

Una vez que obtenemos el nombre de dominio del servidor, debemos consultar por el registro A asociado a mail.spamgate.com.au y si este registro existe obtendríamos la dirección IP del servidor de correos. Si no existe el registro MX o existe el MX pero no el A, podremos asumir que se trata de un correo SPAM.

Adicionalmente se podría verificar que en el puerto 25 de esa IP hay un programa de correo escuchando y que ese servidor acepta correos para ese usuario.

Pregunta 3 (6 puntos)

a) En un protocolo simple (stop and wait) de capa de transporte en una red con ruido, explique los escenarios que pueden dar lugar a segmentos duplicados.

En un protocolo simple como stop and wait, el procedimiento es el siguiente: envío un segmento e inicio un temporizador. Si me llega el reconocimiento (Ack) correspondiente antes de que el temporizador expire, cancelo el temporizador, y estoy en condiciones de enviar el segmento siguiente. Si el temporizador expira (el Ack no llegó a tiempo) reenvío el segmento.

En este caso, los duplicados pueden aparecer por expiración del temporizador. Es decir: envío un segmento, este segmento o su reconocimiento demoran en la red (Tida + Tvuelta > Timeout de retransmisión), mientras tanto expira el temporizador y retransmito. Al receptor le llegarán dos segmentos iguales (a esto le llamamos duplicados).

Otra opción es que se pierda el reconocimiento: envío un segmento, llega correctamente, pero se pierde el Ack. El temporizador del transmisor expira y reenvío el segmento, que llega (duplicado) al receptor.

b) ¿Cómo se resuelve el problema de los segmentos duplicados?

Separaremos la respuesta en dos aspectos: 1) cómo detecto la existencia de duplicados, y 2) qué se hace en este protocolo simple al respecto.

Para detectar duplicados (entre otras funciones), es que tenemos números de secuencia. El número de secuencia consiste en asociar al segmento enviado con un identificador que permita confirmar la recepción del mismo, anunciar su ausencia o pérdida (en protocolos más complejos), y detectar duplicados. En este caso sencillo alcanza con tener dos números de secuencia, dado que nunca tengo más de un segmento en tránsito.

En este protocolo simple de parada y espera el receptor verifica el número de secuencia del segmento recibido, y si no coincide con el que está esperando asume que es un duplicado, descartándolo y enviando un reconocimiento al transmisor reconociendo el último número de secuencia correctamente recibido.

c) ¿Cuál es la eficiencia del protocolo stop and wait? Mencione posibles alternativas que permitan mejorar esa eficiencia.

Dado que envío un segmento, espero el Ack, y ahí vuelvo a enviar un segmento, este protocolo tiene una eficiencia muy baja. En particular, imaginemos que no se pierden segmentos, que la capacidad del canal es R, la velocidad máxima a la que podemos enviar datos sería: (LargoSegmento/R)/(RTT+ (LargoSegmento/R)) [unidades en bps].

*Hay mecanismos más sofisticados de envío de datos, manteniendo la premisa de entregar la totalidad de TPDUs en orden a la capa superior. Dos de ellos son Go Back N y Selective Repeat. Ambos usan el mismo concepto detrás de la mejora en eficiencia: ventanas. La idea central es que tengo la posibilidad de enviar una determinada cantidad de segmentos sin haber recibido confirmación de estos. Para ver la eficiencia en un caso ideal, podríamos pensar que la cantidad de segmentos sin confirmación que puedo tener en tránsito al mismo tiempo (ventana de transmisión) es igual a lo que puedo enviar durante un RTT utilizando toda la capacidad del canal: capacidadCanal*RTT/LargoSegmento.*

Pregunta 4 (9 puntos)

a) ¿Qué es la congestión en las redes de datos? ¿Cómo se manifiesta?

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. En esos casos, existen más paquetes de los que se puede procesar o que quieren utilizar el mismo enlace, por lo que se utiliza una cola donde almacenarlos hasta poder procesarlos.

Los efectos observados incluyen aumento de retardo de ida y vuelta (RTT), dependiendo de en qué lugar de la cola de atención se encuentra el paquete, o pérdidas por descartes cuando no hay más espacio en la cola. A nivel de capa de transporte se observa un incremento en el tiempo en que se envía un segmento y llega su reconocimiento, o se generan retransmisiones en los protocolos, con una disminución de la tasa efectiva de intercambio de datos.

b) ¿Cuáles son las hipótesis del mecanismo de control de congestión implementado por TCP?

La hipótesis "fundacional" es que los enlaces poseen una tasa de error muy baja, por lo que si ocurre un descarte de un paquete (TCP lo percibe como la ausencia de un ACK antes de que expire el temporizador de retransmisión) es debido a congestión (por ejemplo por cola de atención llena).

Esta hipótesis debe ser complementada con otra que acote el tiempo de espera para asumir si un segmento fue descartado (porque el paquete en el que viajaba fue descartado). TCP define un temporizador que utiliza para asumir si un segmento fue descartado, cada vez que envía un segmento empieza a correr este temporizador, la llegada de un reconocimiento (ACK) antes de que expire el temporizador es el funcionamiento esperado. En caso que el ACK no llegue antes del vencimiento del temporizador, se asume que fue descartado y TCP debe re-transmitir el segmento.

En versiones mas modernas de TCP, se interpreta de forma diferente la llegada de ACK triplicados y se acelera la retransmisión del segmento que se identifica como faltante. Pero en definitiva es un refinamiento de las dos primeras hipótesis.

c) ¿Cómo reacciona el mecanismo de control de congestión de TCP ante la no llegada de los reconocimientos de los datos enviados? ¿De qué forma esta reacción contribuye a disminuir la congestión?

*El mecanismo de control de congestión en TCP utiliza un valor de ventana de congestión (**cwnd**) para estimar cuántos datos puede enviar evitando la congestión. Cuando los ACK llegan antes de que expiren los temporizadores, se incrementa el valor de **cwnd** (la magnitud depende de en que fase se encuentre, pero se incrementa de todas formas). Cuando no llegan los ACK, como mencionamos antes, asume que hubo congestión e intenta reaccionar para disminuir la tasa de envío de datos por segundo, el mecanismo de control de congestión disminuye el valor de **cwnd,** la magnitud depende de la variante de TCP. En sus primeras versiones disminuye al tamaño de 1 MSS cuando se detecta la congestión. También se ajusta el valor de Congestion Avoidance (ssthresh), que se utiliza para definir en que fase se trabaja, a la mitad del valor de **cwnd** donde se ocurrió congestión.*

*El momento de enviar datos, la ventana del transmisor W_{TX} (la cantidad de bytes que puede enviar sin esperar los ACK) es el mínimo entre el valor de Windows Size (WS) y la ventana de congestión estimada (**cwnd**). Con esto en mente, un transmisor nunca puede superar la tasa de envío de:*

$$\frac{W_{TX}}{RTT}$$

Ajustando el valor de W_{TX} se ajusta la tasa de envío de datos, por lo cual disminuyendo el valor de ***cwnd*** *estoy logrando disminuir el envío de datos (cantidad de paquetes que llevan segmentos), de forma de que se libere capacidad en el elemento o los elementos que estaba produciendo congestión.*

Nota: *El incremento del valor de RTT es una consecuencia de la existencia de encolamientos, pero no por ello evita que se mantenga la congestión. La congestión se debe al exceso de paquetes, por lo cual debo disminuir la cantidad de paquetes en la red, la forma de controlarlo es ajustando el tamaño de ventana de congestión.*

d) ¿Cómo interpreta TCP la llegada de reconocimientos repetidos?

Un receptor TCP cada vez que le llega un segmento envía un ACK, como los reconocimientos son acumulativos, si los segmentos están en orden, incrementa el número de reconocimiento, pero cuando los segmentos no se encuentran en orden, envía un ACK de todas formas, pero con el último número de secuencia "dado por bueno". Este comportamiento puede generar que un transmisor TCP reciba ACK duplicados o triplicados, por ejemplo, cuando se pierde el primer segmento de una secuencia de cuatro segmentos enviados, y llegan correctamente los tres restantes.

Recibir ACK duplicados es un indicador de que hay un segmento faltante en el receptor, en el caso de ACK triplicados la variante Reno de TCP asume que hubo un evento de congestión puntual que hizo descartar un segmento, pero fue puntual porque siguieron llegando segmentos a destino y se siguen recibiendo ACK duplicados. TCP puede decidir re-enviar el segmento identificado como faltante antes de que expire el temporizador, respecto a ***cwnd*** *no disminuye el valor a 1 MSS sino al valor de umbral de comienzo de funcionamiento de Congestion Avoidance (ssthresh). A su vez, se fija el umbral a la mitad del valor de la ventana actual.*

e) ¿Por qué es importante la correcta estimación del tiempo de ida y vuelta (RTT) entre transmisor y receptor? ¿Cómo estima TCP el valor de RTT?

La importancia de una correcta estimación del RTT radica en como definir el valor del temporizador que se utiliza para asumir que se perdió un segmento. No es posible definir un valor fijo de RTT ya que este depende de los dos interlocutores, podrían estar en la misma LAN o a kilómetros de distancia, por cual los valores de temporizadores deben ser diferentes y ajustarse en cada escenario.

Además el valor de RTT puede variar durante una conexión TCP. En una red con tráfico alto, donde los paquetes deben esperar en una fila para ser procesados, el valor de RTT será mayor que si la red no tuviera tráfico. Si recordamos que la capa IP brinda un servicio de datagramas, cada paquete puede utilizar un camino diferente, lo cual también contribuye a que el valor de RTT varíe a lo largo de una conexión.

TCP utiliza un concepto de estadística para definir el valor del temporizador equivalente al valor medio de RTT más 4 veces el valor de la varianza (indicador de dispersión respecto a la media). Por este motivo la estimación del valor de RTT es fundamental.
El valor de RTT se estima utilizando el valor de las mediciones ***M[n]*** *(cuánto demoró le llegada del ACK del segmento n) y el valor anterior estimado del RTT. Se pondera la información del pasado y la información nueva. También se debe de realizar un estimador de la desviación estándar* ***D[n]****, con la misma idea de asignar pesos diferentes a la información pasada y a la nueva. Por último se calcula el valor del temporizador de Timeout.*

$$RTT[n] = a\ RTT[n-1] + (1 - a)\ M[n] \qquad a = 7/8$$

$$D[n] = b\ D[n-1] + (1 - b)\ |RTT[n-1] - M[n]| \qquad b = 3/4$$

$$Timeout = RTT + 4 * D$$

Nota: *Observar que el timeout no es directamente el estimador de RTT sino que se necesita adicionar "una guarda" que evite que genere falsos positivos (asumir pérdidas cuando no lo son). El incremento del valor M[n], irá incrementando el estimador de RTT, y con ello el temporizador de timeout.*

Pregunta 5 (4 puntos)

a) Explique por qué las direcciones IP se asignan por rangos a las distintas subredes que componen Internet.

La asignación de IPs por rangos permite reducir el tamaño de las tablas de forwarding de los enrutadores. Si por ejemplo todos los equipos de una región estuvieran comprendidos en un mismo rango de IPs, entonces todo el tráfico destinado a estos equipos estará comprendido en una entrada de la tabla de forwarding, logrando con esto una menor necesidad de procesamiento y memoria de estos equipos intermedios así como búsquedas más rápidas.

Además, se logra una asignación jerárquica y una estructura de la distribución de direcciones escalable.

b) Explique qué es el largo de un prefijo de direcciones y cómo se relaciona con la definición de los rangos de direcciones IP. ¿Cómo se relaciona el largo del prefijo con la máscara de la red?

El largo de un prefijo de direcciones determina la cantidad de bits que serán utilizados para identificar un rango de direcciones. El prefijo determinará cuántos bits tienen en común todas las direcciones de ese rango y establece la cantidad de direcciones que se podrán asignar.

Por ejemplo, con un prefijo de largo 25 (los primeros 25 bits se mantienen iguales para todo el rango), podré asignar 126 direcciones útiles (128 contando dirección de red y broadcast), correspondientes a las que se pueden formar variando los 32-25=7 bits menos significativos del ultimo octeto.

La mascara de red estará formada por tantos 1's en sus bits más significativos como el largo del prefijo, y el resto de bits menos significativos en 0. La mascara de red puede ser escrita en notación Dotted-decimal notation o en binario y es utilizada para el encaminamiento de los paquetes en el algoritmo longest-prefix-match.

Pregunta 6 (8 puntos)

a) Describa los campos de las entradas de la tabla de forwarding en IP. ¿Cuándo se usa esta tabla?

Cada entrada en la tabla de forwarding de IP consiste en un rango de direcciones destino y la dirección IP del próximo salto hacia ese destino. Los rangos de direcciones se especificarán con número de red y largo del prefijo (o máscara correspondiente a ese largo de prefijo).

La tabla de forwarding se utilizará en cada nodo de la red ante la llegada de cada paquete que se debe encaminar. Cuando llega un paquete a un enrutador, en base a su dirección IP de destino se consulta la tabla de forwarding para decidir hacia dónde debe encaminarse el paquete. Si hay un rango en la tabla en el que la IP destino del paquete está incluida, la tabla indicará cuál es el próximo enrutador al que hay que hacerle llegar ese paquete.

b) Explique detalladamente el algoritmo de búsqueda en dicha tabla (longest-prefix match).

El algoritmo utilizado para encaminar los paquetes en Internet es el llamado "Longest Prefix Match" que cumple las siguientes reglas:

i. *La tabla de forwarding debe estar ordenada desde las redes más específicas o más chicas (representadas por las máscaras más largas o con más unos) hacia las redes más generales o más grandes (representadas por las máscaras más cortas o con menos unos). Esto permite que si hay más de una entrada en la tabla que sirve para llegar a un mismo destino, se encuentre primero en la tabla aquella que sea más específica para el destino.*

ii. *Se comienza la búsqueda en la tabla por la primer entrada y se realiza el AND bit a bit entre la máscara de esa entrada y la dirección destino del paquete a encaminar. El resultado se compara con el número de red de esa entrada. Si coinciden, el paquete se encamina hacia la IP indicada como próximo salto en la entrada. Si no coinciden, se realiza el mismo procedimiento con la siguiente entrada de la tabla.*

iii. El procedimiento anterior continúa hasta que se encuentre una coincidencia o hasta que se termine la tabla. Si se recorre toda la tabla y no se encuentran coincidencias, el paquete se descarta enviando un mensaje ICMP (Network Unreachable) al originador del paquete, para que se entere que su paquete

fue descartado por no encontrarse una ruta hacia ese destino, el mensaje ICMP es originado en el router que lo descarta, por lo que el originador sabrá en qué punto de la red fue descartado.

iv. *En la tabla puede existir una ruta por defecto que sea tomada por todos los paquetes que no hayan encontrado una ruta específica. Esta entrada se indicará con un rango de destino especificado como 0.0.0.0/0 y la IP del próximo salto asociado a esa entrada. Notoriamente esta entrada al tener máscara de largo 0 y número de red 0.0.0.0, al ser aplicada a cualquier dirección IP destino resultará en una coincidencia (por lo indicado en ii). Además por tener máscara de largo 0 estará ubicada en el último lugar de la tabla y por tanto será elegida solamente si no se encontraron coincidencias más específicas en el resto de la tabla.*