

Definiciones:

ACK:

Un paquete que confirma que se recibió hasta cierto byte (además hay un bit en el header de TCP llamado ACK que indica si con el paquete se está enviando una confirmación)

Tamaño de ventana:

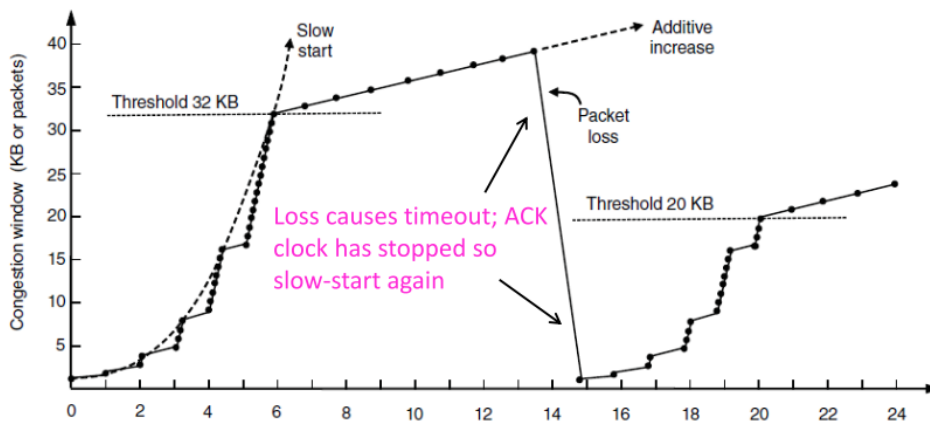
Máxima cantidad de paquetes enviados sin un ACK de confirmación

RTT:

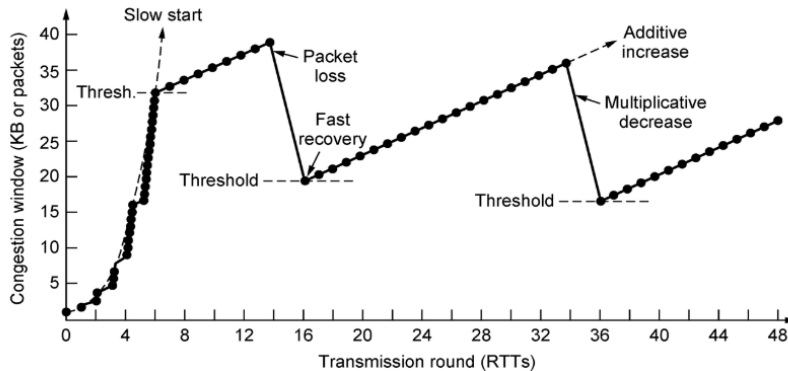
Tiempo de ida y vuelta de 1 bit

Métodos de control de congestión TCP

TCP Tahoe:



TCP reno:



TCP

i)

Ejercicio i: Responder:

- ¿Hasta cuántas palabras de 32 b se pueden tener en un encabezado TCP?
- ¿Hasta cuántas palabras de 32 b puede ocupar el campo de opciones?

El TCP header length es un número de 4 bits, así que el encabezado puede tener a lo sumo $2^4 - 1 = 15$ palabras de 32 b

Hay 5 palabras de 32 b que se ocupan por campos obligatorios, por ende queda espacio para hasta 10 palabras de 32 b para las opciones

ii)

Ejercicio ii: En el encabezado de TCP vimos que además de un campo de confirmación de 32 bits hay un bit ACK. ¿Este campo agrega realmente algo? ¿Por qué o por qué no?

El bit ACK indica si en el paquete va un número ACK, el cuál sirve para el reenvío de paquetes perdidos. Permite al servidor ir sabiendo hasta qué número de byte recibió el receptor

Direccionamiento

1)

Ejercicio 1: Un criterio para decidir si tener un servidor activo todo el tiempo o hacer que comience en demanda usando un servidor de procesos es cuán frecuentemente los servicios provistos son usados. ¿Puede pensar en algún otro criterio para tomar esta decisión?

Otros criterio:

- Cuantos recursos consume cuando no está respondiendo nada
- Cuanto tarda en iniciarse
- Qué tanto importa que responda rapido

2)

Ejercicio 2: ¿Para qué situación se necesita la solución *servidor de procesos*? ¿Cuándo se necesita además un servidor de nombres? Justifique su respuesta.

Servidor de nombres = servidor de directorio

El servidor de procesos se encarga de activar el proceso correspondiente a cierto servicio cuando el servicio es necesitado. Hace falta cuando se ofrecen múltiples servicios en un mismo servidor

El servidor de directorio se usa para que los clientes puedan preguntar qué tipos de servicios ofrece un servidor, y cuál es el número de puerto de cada uno. Se usa cuando un servidor ofrece múltiples servicios, pero los clientes no saben en qué puerto está cada uno

3)

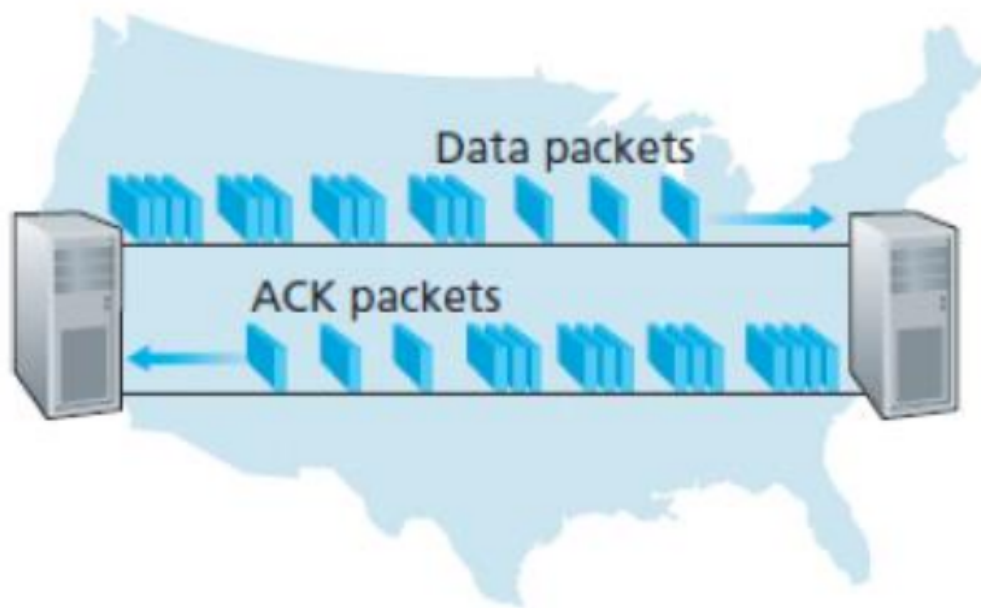
Ejercicio 3: ¿Qué diferencias hay entre protocolo inicial de conexión y direccionamiento en TCP?

El protocolo inicial de conexión es el protocolo que TCP usa para hablar con el servidor de procesos. El servidor de directorio se encarga de direccionar los clientes

Transmisión de Datos Confiable

A)

Ejercicio A. (P15) Considerar el ejemplo que atraviesa Estados Unidos de la **Figura de abajo** ¿Cuán grande tendría que ser el tamaño de ventana para que la utilización del canal sea mayor a 98%? Suponer que el tamaño de un paquete es de 1500 bytes, incluyendo tanto campos de encabezado como datos. Considere RTT de demora de propagación de 30 msec y 1 Gbps de velocidad de transferencia.



RTT = 30 ms

R = velocidad de transferencia = 1 Gb/s

L = tamaño paquete = 1500 B

k = tamaño de ventana

U = utilización del canal = $k * L/R / (RTT + L/R)$

$$\begin{aligned} L/R &= 1500 \text{ B} / (1 \text{ Gb/s}) \\ &= 1500 * 8 \text{ b} / (10^9 \text{ b/s}) \\ &= 0.012 \text{ ms} \end{aligned}$$

U > 0.98

$$\begin{aligned} k * L/R / (RTT + L/R) &> 0.98 \\ k * 0.012 \text{ ms} / (30 \text{ ms} + 0.012 \text{ ms}) &> 0.98 \\ k &> 0.98 * 30.012 \text{ ms} / (0.012 \text{ ms}) \\ k &> 2450.98 \\ k &\geq 2451 \end{aligned}$$

Con un tamaño de ventana de 2451 alcanza

B)

Ejercicio B: (P22) Considerar el protocolo retroceso N con una ventana emisora de tamaño 4 y rango de números de secuencia de 1024. Suponer que en el tiempo t , el siguiente paquete en orden que el receptor está esperando tiene un número de secuencia de k . Asumir que el medio no reordena los mensajes. Contestar las siguientes preguntas:

- ¿Cuáles son los posibles conjuntos de números de secuencia dentro de la ventana del emisor en el tiempo t ?
- ¿Cuáles son los posibles valores del campo ACK en todos los mensajes posibles corrientemente propagándose hacia el emisor en el tiempo t ? Justifique su respuesta.

a.

(sumas y restas módulo 1024)

$\{x, x+1, x+2, x+3\} : x \in \{k-4, \dots, k\}$

b.

(sumas y restas módulo 1024)

$\{k-4, k-3, k-2, k-1\}$

Pueden haber llegado hace poco paquetes con cualquiera de esos 4 números de secuencia, por ende, las confirmaciones ACK pueden todavía no haber llegado

C)

Ejercicio C: Responder las siguientes preguntas sobre protocolos de tubería:

- ¿Qué representa/significa la ventana corrediza emisora para retroceso N? ¿Y para repetición selectiva?
- ¿Por qué en el protocolo de repetición selectiva se tiene que pedir que tamaño de ventana receptora = $(MAX_SEQ + 1)/2$? (o sea, qué situación se quiere evitar).
- ¿Por qué motivo se usa un temporizador auxiliar en el protocolo de repetición selectiva?

1. 🤔

Tanto para retroceso N como para repetición selectiva es la máxima diferencia entre el último paquete enviado y el paquete mas viejo aun no confirmado

2.

Para evitar la situación de que llegue un paquete nuevo y que el receptor lo interpreta como una viejo

3. 🤔

Se usa para esperar un ratito antes de enviar la confirmación ACK. Sirve para que si justo se manda un paquete de vuelta con datos, se lo aprovecha para mandar la confirmación ACK.

¿Y también para confirmar varios paquetes al mismo tiempo?

D) ~

Ejercicio D: Un cable conecta un host emisor con un host receptor; se tiene una tasa de bits de 4 Mbps y un retardo de propagación de 0,2 mseg. ¿Para cuál rango de tamaños de segmentos de parada y espera una eficiencia de al menos 50%?

parada y espera = mandar un paquete y esperar a la confirmación de recepción

$R = \text{velocidad} = 4 \text{ Mb/s} = 4 \text{ Kb/ms}$

$RTT = 2 * 0.2 \text{ ms} = 0.4 \text{ ms}$

$L = \text{tamaño de paquete}$

$U = \text{eficiencia} = (L/R)/(L/R + RTT)$

$$U \geq 0.5$$

$$(L/R)/(L/R + RTT) \geq 0.5$$

$$L/R \geq 0.5 * (L/R + RTT)$$

$$L/R \geq 0.5 * L/R + 0.5 * RTT$$

$$L \geq 0.5 * L + 0.5 * RTT * R$$

$$0.5 * L \geq 0.5 * RTT * R$$

$$L \geq RTT * R$$

$$L \geq 0.4 \text{ ms} * 4 \text{ Kb/ms}$$

$$L \geq 1.6 \text{ Kb}$$

$$L \geq 1600 \text{ b}$$

E)

Ejercicio E: Un cable de 3000 Km de largo une dos hosts y es usado para transmitir segmentos de 1500-bytes usando protocolo retroceso N. La velocidad de transmisión es de 20 Mbps. Si la velocidad de propagación es de 6 µsec/km. ¿cuántos bits deberían tener los números de secuencia?

$R = \text{velocidad} = 20 \text{ Mb/s} = 20000 \text{ Kb/s}$

$RTT = 2 * 3000 \text{ Km} * 6 \text{ µs/km} = 36000 \text{ µs} = 36 \text{ ms}$

$L = \text{tamaño del paquete} = 1500 \text{ B} = 12 \text{ Kb}$

La máxima eficiencia es 1, así que uso $U = 1$

$T_{\text{TransiciónPaquete}} = R/L = 12 \text{ Kb} / (20000 \text{ Kb/s}) = 0.0006 \text{ s} = 0.6 \text{ ms}$

La máxima cantidad de paquetes mandados en un RTT es:

$$36 \text{ ms} / (0.6 \text{ ms}) = 60$$

Esto significa que hacen falta como máximo 61 números de secuencia (uno extra para el primero), o sea que con $\log_2(61) \approx 5.93 \approx 6$ bits alcanza

F) ~

Ejercicio F: Segmentos de 10000 bits son enviados por canal que opera a 10 Mbps usando un satélite geoestacionario cuyo tiempo de propagación desde la tierra es 270 msec. Las confirmaciones de recepción son siempre enviadas a caballito en los segmentos, los encabezados son muy cortos.

Números de secuencia de 8 bits son usados. ¿Cuál es la utilización máxima del canal para

1. parada y espera?
2. retroceso N?
3. repetición selectiva?

L = tamaño de paquete = 10000 b = 10 Kb

RTT = 270 ms * 4 = 1080 ms (Los bits tienen que hacer el viaje host1 - satélite - host2 - satélite - host1)

R = velocidad = 10 Mb/s

max_seq = $2^8 - 1 = 255$

$$L/R = 10 \text{ Kb} / (10 \text{ Mb/s}) = 0.0001 \text{ s} = 1 \text{ ms}$$

1.

k = 1

$$\begin{aligned} U &= k * L/R / (L/R + RTT) \\ &= 1 * 1 \text{ ms} / (1 \text{ ms} + 1080 \text{ ms}) \\ &= 1 \text{ ms} / (1081 \text{ ms}) \\ &= 1/1081 \\ &\approx 0.000925069 \\ &= 0.092069\% \end{aligned}$$

2.

k puede ser como máximo max_seq

k = max_seq = 255

$$\begin{aligned} U &= k * L/R / (L/R + RTT) \\ &= 255 * 1 \text{ ms} / (1 \text{ ms} + 1080 \text{ ms}) \\ &= 255 / 1081 \\ &\approx 0.23589 \end{aligned}$$

3.

k puede ser como máximo (max_seq + 1)/2

k = (max_seq + 1)/2 = (255 + 1)/2 = 128

$$\begin{aligned} U &= k * L/R / (L/R + RTT) \\ &= 128 * 1 \text{ ms} / (1 \text{ ms} + 1080 \text{ ms}) \end{aligned}$$

$$= 128 / 1081$$

$$\approx 0.118409$$

G)

Ejercicio G: Computar la fracción del ancho de banda que es desperdiciado en sobrecarga (encabezados y retransmisiones) por el protocolo de repetición selectiva en un canal de 50 kbps usando segmentos de 8000 bits de datos. Asumir que los encabezados son del tamaño como en IP, TCP y 16B para capa de enlace de datos (terminadores de tramas son de 4B). Asumir que la propagación de la señal desde la Tierra al satélite es de 270 msec. Segmentos de solo ACK nunca ocurren, segmentos NAK ocupan 512 bits. La tasa de errores para segmentos es del 1%, y la tasa de errores de segmentos NAK se puede ignorar (es demasiado chica para considerarla). Los números de secuencia son de 8 bits.

$$L = \text{tamaño de paquete} = 8000 \text{ b} = 8 \text{ Kb}$$

$$\text{headerTCP} = 5 \times 32 \text{ b} = 160 \text{ b}$$

(valor sacado de los PDF)

$$\text{headerIP} = 20 \text{ B} = 160 \text{ b}$$

(valor de wikipedia)

$$\text{header enlace} = 16 \text{ B} = 128 \text{ b}$$

$$\text{terminadores capa enlace} = 4 \text{ B} = 32 \text{ b}$$

$$L_{\text{enviado_paquete}} = \text{tamaño enviado por cada paquete} = 8 \text{ Kb} + 160 \text{ b} + 160 \text{ b} + 128 \text{ b} + 32 \text{ b} = 8380 \text{ b}$$

$$L_{\text{segmentoNAK}} = 512 \text{ b}$$

$$L_{\text{total_cuando_no_falla}}$$

$$= L_{\text{enviado_paquete}} + L_{\text{segmentoNAK}}$$

$$= 8380 \text{ b} + 512 \text{ b}$$

$$= 8892 \text{ b}$$

$$L_{\text{total_con_fallos}}$$

$$= (\sum_{i=0}^{\infty}) 0.01^i * L_{\text{total_cuando_no_falla}}$$

$$= 8892 \text{ b} / (1 - 0.01)$$

$$= 8892 \text{ b} / 0.99$$

$$\approx 8981.81818 \text{ b}$$

$$\text{sobrecargaTotal}$$

$$= L_{\text{total_con_fallos}} - L_{\text{enviado_paquete}}$$

$$\approx 8981.81818 \text{ b} - 8348 \text{ b}$$

$$= 633.81818 \text{ b}$$

$$\text{sobrecargaProporción}$$

$$= \text{sobrecargaTotal} / L_{\text{total_con_fallos}}$$

$$\approx 633.81818 \text{ b} / 8981.81818 \text{ b}$$

$$\approx$$

$$= 7.05668 \%$$

l) 🤔

Ejercicio I: Suponer que se tiene una conexión entre un emisor y un receptor, que los números de secuencia son de 4 bits (o sea van de 0 a 15). Asumir que el receptor tiene 4 búferes en total, todos de igual tamaño. Suponer que se usa la solución donde el emisor solicita espacio de búfer en el otro extremo. Mostrar la comunicación entre emisor y receptor de acuerdo a los siguientes eventos:

1. El emisor pide 8 búferes.
 2. El receptor otorga 4 búferes y espera el segmento de número de secuencia 0.
 3. El Emisor envía 3 segmentos de datos , los dos primeros llegan y el tercero se pierde.
 4. El receptor confirma los 2 primeros segmentos de datos y otorga 3 búferes.
 5. El emisor envía dos segmentos de datos nuevos que llegan y luego reenvía el segmento de datos que se perdió.
 6. El receptor confirma todos los segmentos de datos y otorga 0 búferes.
 7. El receptor otorga un búfer
 8. El receptor otorga 2 búferes
 9. El emisor manda 2 segmentos de datos
 10. El receptor otorga 0 búferes
 11. El receptor otorga 4 búferes pero este mensaje se pierde
- Para segmentos de datos enviados indicar número de secuencia
 - Para segmentos de respuesta indicar cantidad de búferes otorgados y segmentos confirmados, asumir que no se envían datos en estos segmentos.
 - Mostrar asignación de números de secuencia de segmentos recibidos a búferes del receptor

1. → pide 8 búffers
2. ← ACK 15, 4 búffer, [0,1,2,3]
3. → segmento 0
→ segmento 1
→ segmento 2 (perdido)
4. ← ACK 1, 3 búffers, [2,3,4]
5. → segmento 3
→ segmento 4
→ segmento 2
6. ← ACK 4, 0 búffers, []
7. ← ACK 4, 1 búffers, [5]
8. ← ACK 4, 2 búffers, [5,6]
9. → segmento 5

→ segmento 6

10. ¿El receptor los recibió a esos? Supongamos que si

← ACK 6, []

11.

[7,8,9,10]

J)

Ejercicio J: suponer que hay una conexión TCP entre un emisor y un receptor. El receptor tiene un buffer circular de 4 KB. Mostrar los segmentos enviados en ambas direcciones suponiendo los siguientes cambios de estado en el búfer del receptor:

1. El búfer del receptor está vacío.
 2. El búfer del receptor tiene 2KB
 3. El búfer del receptor tiene 4KB (lleno)
 4. La aplicación del receptor lee 2KB
 5. El búfer del receptor tiene 3KB
- Mostrar tamaños y números de secuencia para segmentos enviados.
 - Mostrar tamaño de ventana y número de confirmación de recepción para segmentos recibidos.
 - Mostrar cómo varía el uso del búfer circular.

¿Cuándo los números de secuencia son por byte y cuando por segmento?

Supongo que acá son por byte:

1→2

→Se mandan 2 KB de datos (números de secuencia 0,2K)

2→3

→Se mandan 2 KB de datos (números de secuencia 2K,4K)

3→4

←Se asignan 2KB de buffer

4→5

←Se asigna 3KB de buffer

K)

Ejercicio K: Suponer que hay una conexión TCP entre un emisor y un receptor. Asumir que en un momento dado el receptor anuncia un tamaño de ventana de 816 KB. Explicar cómo expresa TCP esta situación con los campos en su encabezado. ¿Qué campos se usan y qué valores tendrían?

El campo tamaño de ventana tiene 16 bits, por lo que puede expresar hasta 64 KB.

La estrategia que se usa para tamaños más grandes es en las opciones definir un escalado para el tamaño de ventana. El escalado siempre es una potencia de 2, por lo que es como hacer un shift lógico a la izquierda. El máximo escalado 2^{14} (o sea hacer un shift de 14 bits), o sea que con eso el máximo tamaño de ventana es $2^{(14+16)} \text{ B} = 2^{30} \text{ B} = 1 \text{ GB}$

Hay varios factores de escalado de la ventana con los que sería posible, una forma sería:

Escala de ventana = 1024

L)

Ejercicio L: Un emisor en una conexión TCP que recibe un 0 como tamaño de ventana periódicamente prueba al receptor para descubrir cuándo la ventana pasa a ser distinta de 0. ¿Por qué podría el receptor necesitar un temporizador extra si fuera responsable de reportar que su ventana pasó a ser distinta de cero (es decir, si el emisor no mandó un segmento de prueba)?

Porque si no se produciría un deadlock, ya que ambos se quedarían esperando un mensaje del otro

Ejercicios de control de congestión

1)

Ejercicio 1: considere el efecto de usar arranque lento en una línea con 10 msec de tiempo de ronda y no hay congestión. La ventana receptora es de 24KB y el tamaño de segmento máximo es de 2 KB. ¿Cuánto toma antes que la primera ventana llena pueda ser enviada?

1° ronda de envíos: 2 KB

2° ronda de envíos: 4 KB

3° ronda de envíos: 8 KB

4° ronda de envíos: 16 KB

5° ronda de envíos: 24 KB (sería 32, pero no lo es por el límite de la ventana receptora)

En la quinta ronda ya se manda una ventana llena. En la quinta ronda se empieza a mandar a los $4 * 10$ ms = 40 ms, que es el tiempo que toma hasta que la primera ventana llena es enviada

2)

Ejercicio 2: suponga que la ventana de congestión de TCP es fijada en 18 KB y que ocurre una expiración de temporizador. ¿Cuán grande va a ser la ventana si las siguientes 4 ráfagas de transmisiones son todas exitosas? Asumir un tamaño de segmento máximo de 1KB.

Con TCP Tahoe: empieza desde 0 un arranque lento hasta $18/2$ y sigue con aumento lineal, por lo que los tamaños de ventana después de las ráfagas son:

1°: 1 KB

2°: 2 KB

3°: 4 KB

4°: 8 KB

Por lo que el tamaño de ventana se vuelve 8 KB

Con TCP Reno: se empieza desde $18/2$ con aumento lineal, por lo que los tamaños de ventana después de las ráfagas son:

1°: 9 KB

2°: 10 KB

3°: 11 KB

4°: 12 KB

Por lo que el tamaño de ventana se vuelve 12 KB

3)

Ejercicio 3: Una entidad TCP abre una conexión y usa arranque lento. ¿Aproximadamente cuántos RTT son requeridos antes de que TCP pueda enviar N segmentos?

La cantidad de segmentos enviados en x RTT (asumiendo que no se congestiona la red) son:

$$\sum_{i=0}^x 2^i \text{ segmentos} \\ = 2^{x+1} - 1 \text{ segmentos}$$

Para enviar N segmentos (los últimos no necesariamente llegaron):

$$2^{x+1} - 1 = N \\ 2^{x+1} = N + 1 \\ x + 1 = \log_2(N + 1) \\ x = \log_2(N + 1) - 1$$

4)

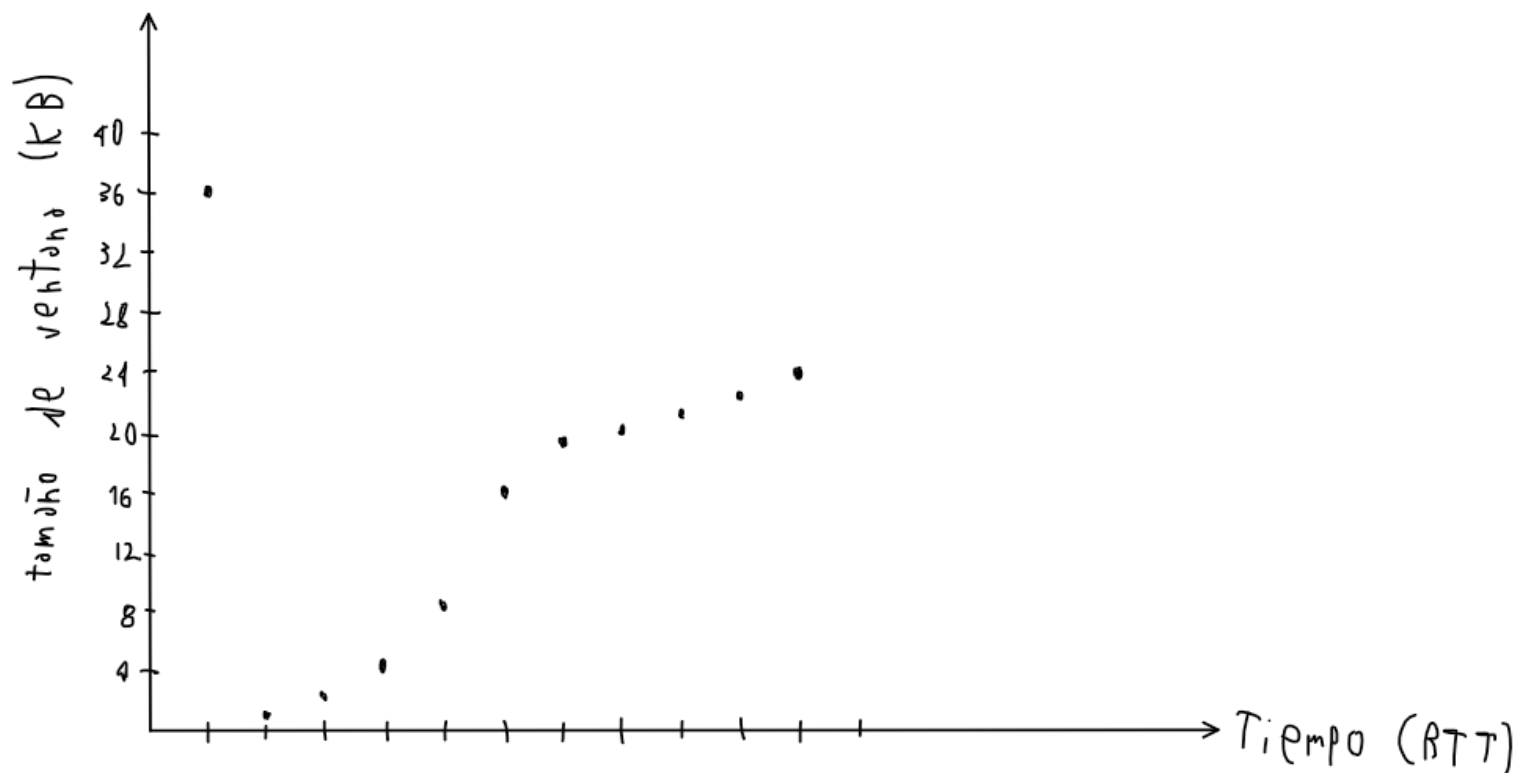
Ejercicio 4: Asumir que se usa algoritmo TCP Tahoe, la ventana de congestión es fijada a 36 KiB y luego ocurre un timeout; luego de esto el algoritmo hace lo que tiene que hacer y la ventana de congestión llega hasta los 24 KB con éxito sin que ocurran nuevos timeouts. Asumir que el segmento máximo usado por la conexión es de 1KB de tamaño. Responder:

1. ¿Si tuviera que hacer un diagrama cartesiano del comportamiento del algoritmo TCP Tahoe qué representa cada uno de los ejes cartesianos?
2. Hacer un diagrama cartesiano mostrando el comportamiento del algoritmo TCP Tahoe desde que ocurre el timeout mencionado (luego de los 36 KB) hasta que la ventana de congestión llega a 24 KB.

1.

En el eje horizontal el tiempo en cantidad de RTT, y en el eje vertical el tamaño de ventana en KB

2.



5)

Ejercicio 5: Supongamos que se usa el algoritmo de control de congestión TCP Reno. Inicialmente el umbral está fijado a 32KiB. Inicia la conexión y el algoritmo TCP Reno comienza a operar. Ocurren 10 rondas de transmisión antes de un timeout. Se pide:

- Mostrar el desempeño del algoritmo de TCP Reno desde el inicio (una vez iniciada la conexión) hasta 6 rondas de transmisión exitosas luego del timeout señalado. Asumir que el segmento máximo usado por la conexión es de 1 KiB de tamaño.

ventana de congestión: 32 KiB

ronda 1: 1 KiB

ronda 2: 2 KiB

ronda 3: 4 KiB

ronda 4: 8 KiB

ronda 5: 16 KiB

ronda 6: 32 KiB

ronda 7: 33 KiB

ronda 8: 34 KiB

ronda 9: 35 KiB

ronda 10: 36 KiB Timeout: ventana de congestión: 18 KiB

ronda 11: 18 KiB

ronda 12: 19 KiB

ronda 13: 20 KiB

ronda 14: 21 KiB

ronda 15: 22 KiB

ronda 16: 23 KiB

- **Ejercicio:** Considerar el efecto de usar arranque lento en una línea con un RTT de 10-msec y sin congestión. La ventana del receptor es de 24 KB y el tamaño de segmento máximo es de 2 KB. ¿Cuánto tiempo lleva antes de que una ventana completa pueda ser enviada?

En cada ronda se manda:

ronda 1: 2 KB

ronda 2: 4 KB

ronda 3: 8 KB

ronda 4: 16 KB

ronda 5: 24 KB

En la ronda 5 ya se manda una ventana completa, por ende, tienen que pasar 4 rondas (4 RTTs) hasta que se manda una ventana completa, o sea:

$$4 * RTT = 4 * 10 \text{ ms} = 40 \text{ ms}$$

Parte 2

Comparación de segmentos y establecimiento de conexiones

A)

Ejercicio A: el campo de números de secuencia en el encabezado TCP es de 32 bits de largo, lo cual es suficientemente largo para cubrir 4 billones de bytes de datos. Incluso si tantos bytes nunca fueran transferidos por una conexión única, ¿por qué puede el número de secuencia pasar de $2^{32} - 1$ a 0?

Porque cuando se inicia una conexión, por lo general no se empiezan los números de secuencia desde 0, si no que se suele elegir un número al azar, o un número dependiendo en un reloj, entonces, se puede empezar en un número cercano a $2^{32} - 1$

34)

Ejercicio 34: ¿Cuál es la velocidad más rápida de una línea en la cual un host puede enviar cargas útiles de TCP de 1500 B con un tiempo de vida de paquete de 120 seg sin que los números de secuencia den vuelta? Tomar en cuenta la sobrecarga de TCP, IP y Ethernet. Asumir que las tramas de Ethernet se pueden mandar continuamente.

$$T_{\text{vidaSegmento}} = 120 \text{ s}$$

$$\text{encabezado TCP} = 20 \text{ B}$$

$$\text{encabezado IP} = 20 \text{ B}$$

$$\text{encabezado Ethernet} = 42 \text{ B}$$

$$\text{tamaño sobrecarga} = 20 \text{ B} + 20 \text{ B} + 42 \text{ B} = 82 \text{ B}$$

$$U_{\text{sobrecarga}} = \text{tamaño sobrecarga} / \text{tamaño carga útil} \\ = 82 \text{ B} / 1500 \text{ B}$$

$$\text{bits números de secuencia} = 32 \text{ b}$$

Eso significa que en 120 s se pueden mandar como máximo 2^{32} B de carga útil, o sea, la velocidad máxima es:

$$2^{32} \text{ B} / (120 \text{ s}) = 4096 \text{ MB} / (120 \text{ s}) \approx 34.133 \text{ MB/s}$$

36)

Ejercicio 36: En una red cuyo segmento máximo es de 128 B, tiempo de vida máximo de segmento es 30 seg y tiene números de secuencia de 8 bits. ¿Cuál es la tasa de datos máxima por conexión?

$$\text{MAQ_SEQ} = 2^8 - 1$$

$$T_{\text{vidaSegmento}} = 30 \text{ s}$$

$$\text{bits números de secuencia} = 8 \text{ b}$$

$$\text{tamañoSegmento} = 128 \text{ B}$$

Asumiendo que los números de secuencia son por byte;

$$\text{tasa máxima de envío de datos}$$

$$= 2^8 \text{ B} / (30 \text{ s})$$

$$= 8 * 256 \text{ b} / (30 \text{ s})$$

$$\approx 68.266 \text{ B/s}$$

Asumiendo que los números de secuencia son por segmento:

$$\text{tasa máxima de envío de datos}$$

$$= 128 * 2^8 \text{ B} / (30 \text{ s})$$

$$= 8 * 128 * 256 \text{ b} / (30 \text{ s})$$

$$\approx 8738.133 \text{ b/s}$$

39)

Ejercicio 39: Para resolver el problema de que los números de secuencia dan vuelta mientras que los paquetes anteriores aún existen se podrían usar números de secuencia de 64 bits. Sin embargo, teóricamente una fibra óptica puede correr a 75 Tbps. ¿Qué tiempo de vida máximo de paquete es requerido para asegurarse que redes futuras de 75 Tbps no tienen problemas de números de secuencia que den vuelta, incluso con números de secuencia de 64 bits? Asumir que cada byte tiene su propio número de secuencia como lo hace TCP.

$$\text{MAQ_SEQ} = 2^{64} - 1$$

bits números de secuencia = 64 b

tasa máxima de envío de datos = 75 Tb/s

$T_{\text{vidaSegmento}}$

$$= 2^{64} \text{ B} / (75 \text{ Tb/s})$$

$$= 8 \cdot 2^{64} \text{ b} / (75 \cdot 10^{12} \text{ b/s})$$

$$\approx 1967652.7012 \text{ s}$$

B)

Ejercicio B: lo han contratado para diseñar un protocolo que usa una ventana como TCP. Este protocolo va a correr sobre una red de 1 Gbps. El RTT de la red es 100 ms y el tiempo de vida máximo de segmento es de 30 segundos (o sea, no pueden reutilizarse números de secuencia durante este tiempo – esto es para evitar que haya dos segmentos diferentes con el mismo número de secuencia). ¿Cuántos bits incluirá para el tamaño de ventana y para el campo de número de secuencia en el encabezado de su protocolo? Justifique su respuesta.

Asumo que el tamaño de ventana y los números de secuencia son por byte

RTT = 100 ms

R = velocidad = 1 Gb/s

Bits campo de tamaño de ventana:

$$T_{\text{trasmisiónByte}} = 1 / (1 \text{ Gb/s}) = 8 \cdot 10^{-9} \text{ s/B} = 0.008 \text{ } \mu\text{s/B}$$

La cantidad máxima de bytes mandados en un RTT es:

$$\text{RTT} / T_{\text{trasmisiónByte}} = 100 \text{ ms} / (0.008 \text{ } \mu\text{s/B}) = 12\,500\,000 \text{ B}$$

Por ende, el tamaño de ventana tiene que ser de hasta 12 500 000 B, y por ende, el campo tiene que tener por lo menos $\lceil \log_2(12500000) \rceil \text{ b} = \lceil 23.575 \rceil \text{ b} = \mathbf{24 \text{ b}}$

Bits número de secuencia:

En 30 s se pueden mandar:

$$30 \text{ s} \cdot 1 \text{ Gb/s} = 30 \text{ Gb} = 30\,000\,000\,000 \text{ b} = 3\,750\,000\,000 \text{ B}$$

Cada byte tiene que tener su propio número de secuencia, por ende, hacen falta $\lceil \log_2(3\,750\,000\,000) \rceil \text{ b} = \lceil 31.80 \rceil \text{ b} = \mathbf{32 \text{ b}}$

Seteo de temporizador de retransmisiones

Algoritmo de Jacobson:

Algoritmo de estimación del RTT.

Tiene en cada momento un SRTT como estimación del RTT

Cada vez que llega una confirmación de recepción, se calcula R = tiempo desde que se mandó el paquete hasta que llegó la confirmación y se actualiza:

$$SRTT = \alpha * SRTT + (1 - \alpha) * R$$

Con un $\alpha \in (0, 1)$ llamado factor de suavizado

También se estima la desviación estándar como:

$$D = \beta * D + (1 - \beta) * |SRTT - R|$$

Y se suele establecer el tiempo de transmisión en:

$$RTO = SRTT + 4 * D$$

32)

Ejercicio 32: Si el RTT de TCP es actualmente de 30 msec y las siguientes confirmaciones de recepción vienen luego de 26, 32 y 24 msec respectivamente. ¿Cuál es la nueva estimación del RTT usando el algoritmo de Jacobson? Usar $\alpha = 0,9$.

$$SRTT = 30 \text{ ms}$$

$$SRTT = 0.9 * 30 \text{ ms} + 0.1 * 26 \text{ ms} = 29.6 \text{ ms}$$

$$SRTT = 0.9 * 29.6 \text{ ms} + 0.1 * 32 \text{ ms} = 29.84 \text{ ms}$$

$$SRTT = 0.9 * 29.84 \text{ ms} + 0.1 * 24 \text{ ms} = 29.256 \text{ ms}$$

F)

Ejercicio F: Si el RTT de TCP es actualmente de 30 msec, la desviación media es actualmente 7 msec y las siguientes confirmaciones de recepción llegan después de 26, 32, y 24 msec respectivamente,

- ¿Cuál es la nueva estimación de la desviación media usando el algoritmo de Jacobson?
- ¿Cuál es el nuevo valor de la expiración del temporizador de retransmisiones?
- Nota: usar los resultados del ejercicio 32.

$$\text{Asumiendo } RTTVAR_i = \beta * RTTVAR_{i-1} + (1 - \beta) * |SRTT_{i-1} - R|$$

$$SRTT = 30 \text{ ms}$$

$$RTTVAR = 7 \text{ ms}$$

$$RTTVAR = 0.9 * 7 \text{ ms} + 0.1 * |30 \text{ ms} - 26 \text{ ms}| = 6.7 \text{ ms}$$

$$RTTVAR = 0.9 * 6.7 \text{ ms} + 0.1 * |29.6 \text{ ms} - 32 \text{ ms}| = 6.27 \text{ ms}$$

$$RTTVAR = 0.9 * 6.27 \text{ ms} + 0.1 * |29.84 \text{ ms} - 24 \text{ ms}| = 6.227 \text{ ms}$$

$$SRTT = 29.256 \text{ ms}$$

$$RTTVAR = 6.227 \text{ ms}$$

$$RTO = SRTT + 4 * RTTVAR = 29.256 \text{ ms} + 4 * 6.227 \text{ ms} = 54.164 \text{ ms}$$

G)

Ejercicio G: considerando el algoritmo de Karn (filmina 50 de parte 4 de capa de transporte) contestar: ¿cómo se setea el temporizador de retransmisiones de un paquete nuevo? ¿Cómo se setea el temporizador de retransmisiones de paquete retransmitido por tercera vez?

La primera pregunta es: ¿Cómo se setea el temporizador de retransmisiones cuando llega un ack de un paquete no transmitido

1. $SRTT + 4 D$
2. $\text{expiración del temporizador} := 2 * \text{expiración del temporizador}$