

Rec 1º parcial 2015

Un cliente y un servidor se comunican mediante el protocolo TCP. La aplicación cliente envía una petición de 700 bytes al servidor. La respuesta del servidor es un mensaje de 2483 bytes, tras el que iniciará el cierre de la conexión. Sabemos que el MSS que emplean el cliente y el servidor es de 600 bytes. Los números de secuencia iniciales serán $NSI(C) = 5.000$, $NSI(S) = 8.000$ y las respectivas ventanas de recepción $WIN(C) = WIN(S) = 3048$. Para este ejercicio, el tamaño inicial de la ventana de congestión es dos segmentos ($2 * MSS$ bytes) y se aplican los algoritmos de control de congestión descritos en el temario de la asignatura. Ambos extremos emplean reconocimientos retrasados. Describe la evolución de la conexión TCP, desde el establecimiento hasta el cierre de la conexión. La respuesta ha de reflejarse en la tabla siguiente:

[illegible]

Un cliente y un servidor se comunican mediante el protocolo TCP. La aplicación cliente envía una **petición de 700 bytes al servidor**. La **respuesta del servidor** es un mensaje de **2483 bytes, tras el que iniciará el cierre de la conexión**. Sabemos que el **MSS** que emplean el cliente y el servidor es de **600 bytes**. Los números de secuencia iniciales serán **NSI(C) = 5.000**, **NSI(S) = 8.000** y las respectivas ventanas de recepción **WIN(C) = WIN(S) = 3048**. Para este ejercicio, el **tamaño inicial de la ventana de congestión es dos segmentos (2*MSS bytes)** y se aplican los algoritmos de control de congestión descritos en el temario de la asignatura. Ambos extremos emplean **reconocimientos retrasados**. Describe la evolución de la conexión TCP, desde el establecimiento hasta el cierre de la conexión. La respuesta ha de reflejarse en la tabla siguiente:

Origen (C/S)	Nº Secuencia	Flags	Nº ACK	Datos (byte inicial y final)
C	5000	SYN	-	-
S	8000	SYN,ACK	5001	-
C	5001	ACK	8001	5001-5600
C	5601	ACK	8001	5601-5700
S	8001	ACK	5701	8001-8600
S	8601	ACK	5701	8601-9200
C	5701	ACK	9201	-
S	9201	ACK	5701	9201-9800
S	9801	ACK	5701	9801-10400
S	10401	ACK	5701	10401-10483
C	5701	ACK	10401	-
C	5701	ACK	10484	-
S	10484	FIN	5701	-
C	5701	FIN,ACK	10485	-
S	10485	ACK	5702	-

Primer parcial 2018-2019

Tras establecer una conexión entre un proceso en el host A y otro en el host B, la tabla refleja la evolución de **la ventana de recepción TCP de B** en cada RTT. Suponiendo que A tiene infinitos segmentos para enviar, que en el RTT=6 recibe 3 ACK's duplicados y que en el RTT=11 se produce un *Timeout* (eventos que se detectan al final del RTT, y por tanto afectan al siguiente RTT), completa la tabla siguiente indicando los valores solicitados al comienzo de cada RTT. No se producen otros errores ni se utilizan reconocimientos retardados. Excepto los RTT, las restantes variables se expresan en segmentos.

[illegible]

Solución:

3 ACKs

TimeOut

RTT	1	2	3	4	5	6	7	8	9	10	11	12	13	14
V_rec(B)	32	30	30	32	30	30	28	20	0	20	28	30	32	32
Umbral (A)	32	32	32	32	32	32	15	15	15	15	15	9	9	9
V_cong(A)	2	4	8	16	32	33	15	16	17	17	18	1	2	4
V_trans(A)	2	4	8	16	30	30	15	16	0	17	18	1	2	4

Primer parcial curso 2019-2020

Indica la secuencia total de peticiones y respuestas DNS necesarias para que el servidor de correo smtp.upv.es obtenga la IP del servidor SMTP del dominio gmail.com. El servidor DNS de la UPV (dns.upv.es) tiene en caché los servidores TLD necesarios (y sus IPs) para las consultas que tenga que realizar. Las cachés DNS de los demás equipos que intervienen están vacías. Suponemos que los TLD's conocen todos los servidores de nombres autorizados de su dominio, y que los servidores autorizados de un dominio se nombran como dns.dominio. Por ejemplo, el servidor DNS del dominio gmail.com será dns.gmail.com.

Muestra el resultado en la siguiente tabla de tres columnas: origen, destino, registro de la consulta o respuesta (nombre, valor y tipo). Si en el registro de la consulta algún campo no contiene valor, indícalo con una línea, por ejemplo, una consulta que no contiene valor en el campo “valor”, sería: (p1.yahoo.es, ____, AAAA). En caso necesario pueden indicarse varios registros por intercambio.

[illegible]

Primer parcial curso 2019-2020

Indica la secuencia total de peticiones y respuestas DNS necesarias para que el servidor de correo smtp.upv.es obtenga la IP del servidor SMTP del dominio gmail.com. El servidor DNS de la UPV (dns.upv.es) tiene en caché los servidores TLD necesarios (y sus IPs) para las consultas que tenga que realizar. Las cachés DNS de los demás equipos que intervienen están vacías. Suponemos que los TLD's conocen todos los servidores de nombres autorizados de su dominio, y que los servidores autorizados de un dominio se nombran como dns.dominio. Por ejemplo, el servidor DNS del dominio gmail.com será dns.gmail.com.

Muestra el resultado en la siguiente tabla de tres columnas: origen, destino, registro de la consulta o respuesta (nombre, valor y tipo). Si en el registro de la consulta algún campo no contiene valor, indícalo con una línea, por ejemplo, una consulta que no contiene valor en el campo “valor”, sería: (p1.yahoo.es, ___, AAAA). En caso necesario pueden indicarse varios registros por intercambio.

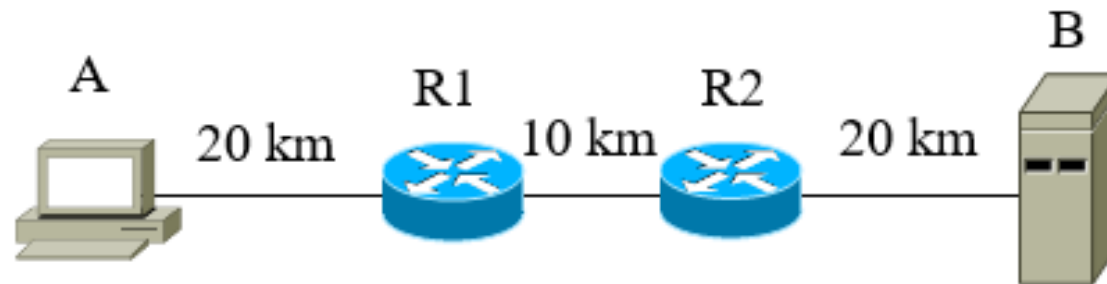
Origen	Destino	Información del registro (nombre, valor, tipo)
smtp.upv.es	dns.upv.es	gmail.com, ----, MX
dns.upv.es	TLD.com	gmail.com, ___, NS
TLD.com	dns.upv.es	gmail.com, dns.gmail.com, NS dns.gmail.com, IP_ dns.gmail.com, A
dns.upv.es	dns.gmail.com	gmail.com, ___, MX
dns.gmail.com	dns.upv.es	gmail.com, smtp.gmail.com, MX smtp.gmail.com, IP_ smtp.gmail.com, A
dns.upv.es	smtp.upv.es	gmail.com, smtp.gmail.com, MX smtp.gmail.com, IP_ smtp.gmail.com, A

Primer parcial enero 2017

El computador A desea transmitir al computador B un mensaje de 20.000 bytes mediante conmutación de paquete. La ruta entre ambos computadores atraviesa dos routers: R1 y R2. La longitud de cada enlace es la que se muestra en el dibujo (20 km y 10 Km). La velocidad de propagación es de 2×10^8 m/s. La velocidad de transmisión en todos los enlaces es 100 Mbits/s. La longitud máxima de los paquetes de datos es de 1.000 bytes y los paquetes de reconocimiento son de 50 bytes. Consideraremos despreciables los tiempos de procesamiento en hosts y routers.

Suponemos que cada vez que se recibe un paquete en B, éste envía un paquete de reconocimiento de 50 bytes (ACK).

1. Calcule el tiempo que transcurre desde que sale el primer bit del mensaje hasta que se recibe el último ACK en A, en las siguientes condiciones:
 - a. Si el algoritmo empleado es de parada y espera.
 - b. Si el algoritmo empleado es de ventana deslizante, con tamaño máximo de la ventana de transmisión de 5 paquetes.
2. Calcule el tamaño mínimo de la ventana de transmisión necesario para conseguir el envío continuo.



Número de paquetes transmitidos = $20.000/1.000 = 20$ paquetes

$$T_{\text{trans_datos}} = 1000 \times 8 / (100 \times 10^6) = 80 \times 10^{-6} \text{ s} = 80 \mu\text{s}$$

$$T_{\text{trans_ack}} = 50 \times 8 / (100 \times 10^6) = 4 \mu\text{s}$$

$$T_{\text{prop_1}} = 20 \times 10^3 / 2 \times 10^8 = 1 \times 10^{-4} \text{ s} = 100 \mu\text{s}$$

$$T_{\text{prop_2}} = 10 \times 10^3 / 2 \times 10^8 = 1 \times 10^{-4} \text{ s} = 50 \mu\text{s}$$

El RTT es el tiempo necesario para transmitir el segmento de datos y recibir su reconocimiento asociado:

$$\text{RTT} = 3(T_{\text{trans_datos}} + T_{\text{trans_ack}}) + 4 T_{\text{prop_1}} + 2 T_{\text{prop_2}} = 3(80 + 4) + 400 + 100 = \mathbf{752 \mu\text{s}}$$

1.a) Parada y espera

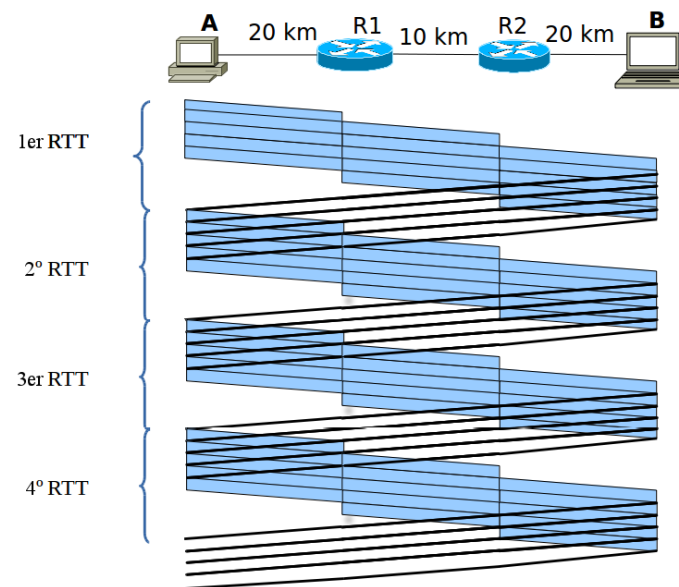
Al emplear parada y espera el tiempo de transmisión total serán 20 RTTs

$$T_{\text{total}} = 20 \text{ RTT} = 20 \times 752 \mu\text{s} = 15040 \mu\text{s} = \mathbf{15,040 \text{ ms}}$$

1.b) Ventana deslizante

Como A puede tener como máximo 5 paquetes pendientes de reconocimiento, puede enviar hasta 5 paquetes antes de recibir un ACK.

Como $5 \times T_{\text{trans_datos}} = 5 \times 80 = 400 \mu\text{s}$ es un tiempo inferior al RTT ($752 \mu\text{s}$), A puede enviar 5 paquetes hasta alcanzar el tamaño máximo de la ventana, después debe detenerse hasta que reciba el primer ACK. Tras lo cual, por cada ACK que reciba podrá enviar un nuevo paquete. Los ACKs siguientes llegarán con una diferencia de tiempo de $T_{\text{trans_datos}}$ cada uno. Por lo tanto, una vez recibido el primer ACK, A puede transmitir los restantes 5 paquetes sin detenerse. Cuando hayan transcurrido 4 RTTs habrá transmitido los 20 paquetes que tiene que transmitir en total y recibirá el ACK del décimo sexto paquete. El tiempo necesario para que lleguen los cuatro ACKs restantes es $4 \times T_{\text{trans_datos}}$. El 4º RTT coincide con el fin de la 4ª llave, que es cuando llega el 16º ACK). El tiempo necesario para que lleguen los cuatro ACKs restantes es $4 \times T_{\text{trans_datos}}$, como puede verse en la figura.



$$\begin{aligned} T_{\text{total}} &= 4 \times \text{RTT} + 4 \times T_{\text{trans_datos}} \\ &= 4 \times 752 + 4 \times 80 = 3008 + 320 = \\ &3328 \mu\text{s} = \mathbf{3,328 \text{ ms}} \end{aligned}$$

2) Envío continuo

Para conseguir envío continuo se requiere que $n \times T_{\text{trans_datos}} > RTT$, donde n es el tamaño máximo de la ventana de transmisión. En nuestro caso, el valor mínimo de n será:

$$n \times 80 \mu s > 752 \mu s$$

$$n > 752/80 = 9,4$$

Como n debe ser un valor entero, el valor mínimo para conseguir envío continuo será **$n = 10$** .

Primer parcial 2014

La siguiente gráfica muestra la evolución del valor de la **ventana de recepción** en una conexión TCP. Dibuje sobre la gráfica la evolución de la ventana de congestión y de transmisión del transmisor suponiendo que no hay ningún error en la comunicación y que siempre hay datos de aplicación disponibles para ser enviados y que no se emplean reconocimientos retardados. (Inicialmente Umbral = WIN)

