Deusto
Facultad de Ingeniería
Faculty of Engineering

TELEMÁTICA
Entrega 1 – TCP y Retardos

Vista rápida de la entrega

Nota	½ del 20% de la nota final (hay tres problemas)
Fecha entrega	16 de noviembre del 2020

1. Problemas de retardo (10%)

Considere el siguiente escenario de la Figura 1: **A** y **B** desean establecer una comunicación a través de dos routers intermedios **R1** y **R2**. El ancho de banda, la distancia y la velocidad de propagación para cada uno de los tramos se provee en la figura. Como se puede observar, la velocidad de propagación de los dos primeros tramos está marcada como una "**M**". Ésta es una medida individual por alumno/a y se debe sustituir por el último número de vuestro DNI.

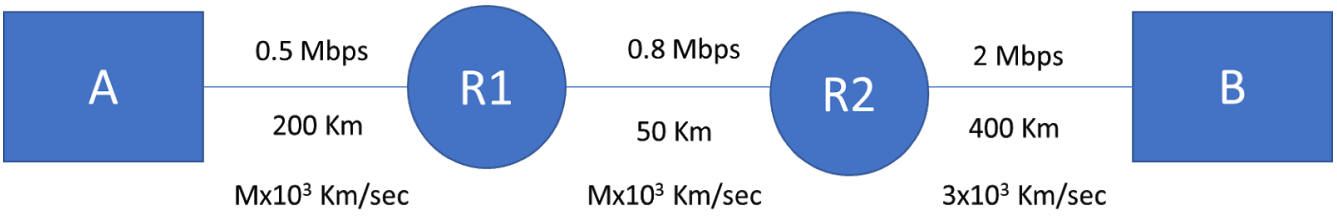


Figura 1. Escenario de comunicación

El host **A** envía un mensaje de 200 Kbyte al host **B.** El máximo tamaño de paquete que **A** puede enviar a **B** es **P** (donde **P** será el primer número de vuestro DNI sin incluir el 0) multiplicado x 10 Kbytes. Es decir, si el primer dígito de tu DNI es 3, el tamaño máximo del mensaje será: **P** = 30 Kbytes.
Con esta información:

a) [1%] Cuántos paquetes harán falta para enviar el fichero completamente? Por favor, darse cuenta que dependiendo del valor de P el número variará y que el último paquete puede ser de menor tamaño, pero sigue contando como 1.
b) [6%] De acuerdo a los dos ofrecidos en el ejercicio, ¿Cuánto tiempo pasa desde que **A** empieza a enviar el primer bit hasta que **B** recibe el último bit del último paquete que **A** envió? Es decir, hasta

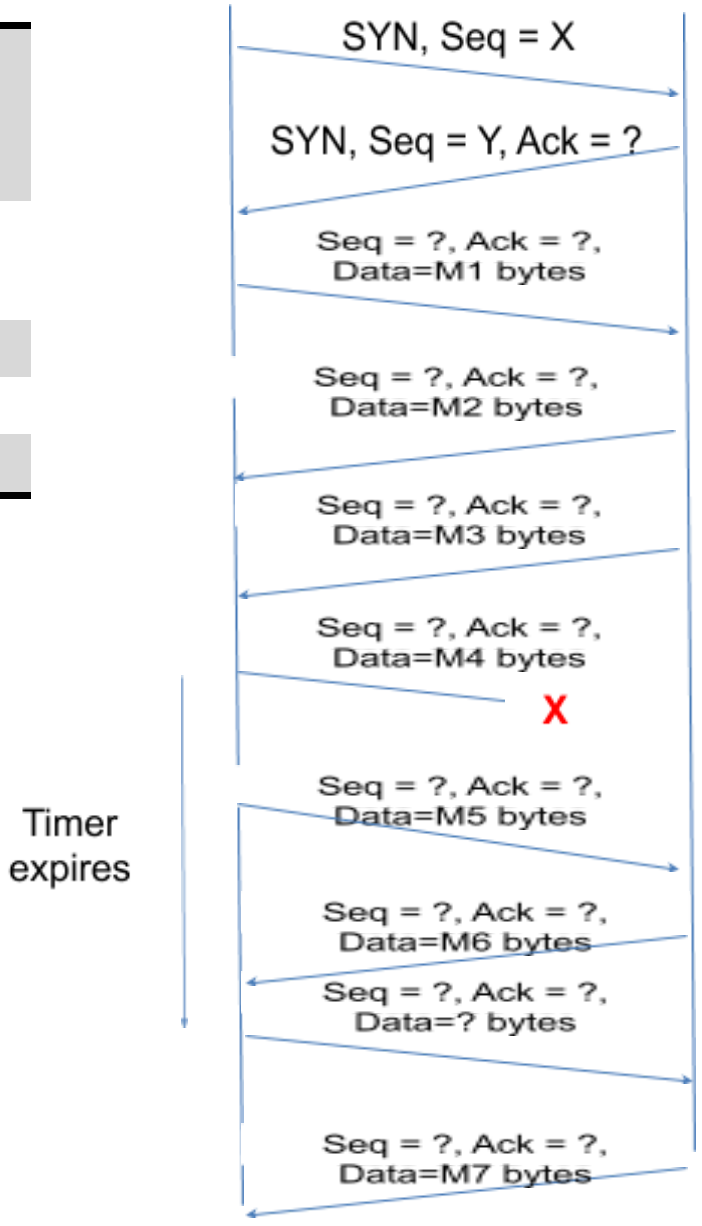
que el mensaje llega completamente al destino. Por favor, provee un diagrama del envío de datos entre **A** y **B**.

c) [3%] En el caso de que el host **B** respondiese a **A** con un paquete de confirmación por cada paquete recibido que ocupase 1 Kbyte, ¿Cuánto tiempo pasaría desde que **A** empieza a transmitir hasta que **A** recibe la confirmación de que se ha recibido el último paquete enviado? Al igual que en el ejercicio previo, realiza un diagrama del envión y confirmación de paquetes.

2. Problema 2: Diagrama TCP (5%)

Compleata el diagramas siguiente rellenando correctamente los números donde haya un símbolo de interrogación "?". Como puedes observar, cada estudiante tendrá un número de secuencia diferente así como un número de Bytes trasmitidos diferente también. Para este ejercicio, vamos a asumir que el host receptor es capaz de guardar segmentos que llegan fuera de orden a la espera de que lleguen los segmentos que le faltan (bien porque el canal sea ruidoso y tarden más en llegar, o porque se perdieron y hay que retrasmitirlos)

X	Los dos últimos dígitos de tu DNI
Y	Los dos primeros dígitos de tu DNI
M1	120
M2, M6, M7	El día de tu cumpleaños multiplicado x 2
M3	El mes de tu cumpleaños x 30
M4	El día de tu cumpleaños multiplicado x 15
M5	El mes de tu cumpleaños x 10



3. Problema 3: Control de congestión[1] (5%)

El Host A (el cliente) crea una conexión TCP con el Host B (servidor) y comienza la transmisión usando el mecanismos de *SlowStart*[2] *(*solo puede haber un segmento en tránsito*)* ya que no conoce cómo se va a comportar la red ni los RTTs precedentes. Inicialmente se dispone a trasmitir un fichero que ocupa 116+2*N Kbyte (donde N es el mes de tu cumpleaños. Por ejemplo, si naces en el mes de julio el tamaño de tu fichero sería 116+2x7=130 KByte). El MSS (máximo tamaño de segmento) se establece en Kbyte. Con esta información:

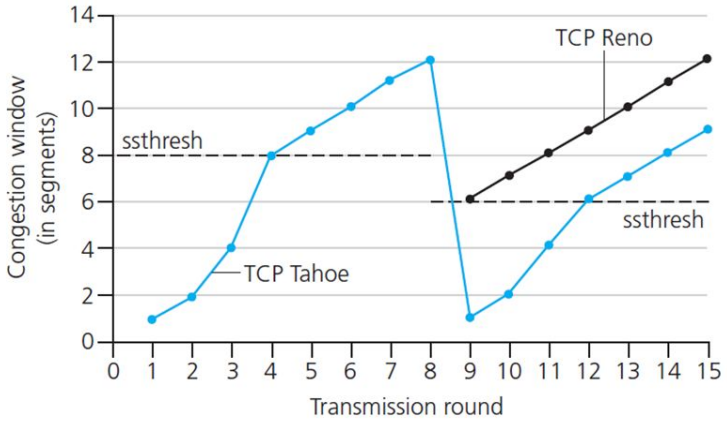
a) (0.5%) Asumiendo que no se pierde ningún segmento, ¿en qué ronda de transmisión se completaría el envío total del fichero? **Justifica tu respuesta.**

b) (1%) Imagina que se reciben 3 ACKs duplicados en el segmento número 64 (se asume que ese segmento se perdió y que ya no hay más pérdidas) y que TCP usa ahora el algoritmo Tahoe[3] para el control de congestión. ¿en qué ronda de transmisión se completaría el envío total del fichero? **Justifica tu respuesta.**

c) (0.5%) Dibuja un diagrama similar al de la figura 3.58 del libro de Kurose (la que tenéis abajo) describiendo la evolución de la ventana de congestión durante cada una de las rondas de transmisión referidas a la pregunta B. Nótese que entra en juego también el algoritmo TCP Reno[4].

d) (2%) Empezando de nuevo con la pregunta "a", si la ventana de recepción (*rwnd*) del Host B es tan grande como el día de tu cumpleaños + 6 (Si naciste el 12, tu *rnwd* sería 18KByte) y teniendo el cuenta que la aplicación del Host B es capaz de leer toda la información que hay en el buffer de una sola vez, ¿en qué ronda de transmisión se completaría el envío total del fichero si no hay pérdida de segmentos? **Justifica tu respuesta.**

Diagrama para la pregunta c):



[1] Varios de los mecanismos que usaremos en esta práctica están someramente descritos en (https://en.wikipedia.org/wiki/TCP_congestion_control#TCP_Tahoe_and_Reno). Se recomienda repasar el concepto, "Congestion avoidance": Congestion Avoidance es un algoritmo muy relacionado con Slow Start y su objetivo es adaptar la tasa de la fuente TCP a la situación de la red: ausencia de congestión, en cuyo caso va aumentando la tasa lentamente o congestión, percibida como pérdida de paquetes (Timeout), en cuyo caso toma medidas similares a Slow Star, que implican una reducción del ritmo de incremento de la ventana de transmisión

[2] Slow Start es un algoritmo diseñado para que el transmisor adapte su tasa de transmisión a la tasa que la red puede aceptar, y se basa en el principio de que la tasa a la que el transmisor puede enviar paquetes debe ser la tasa a la que le llegan los reconocimientos (ACK) del otro extremo

[3] TCP Tahoe es una estrategia usada en TCP para controlar la congestión cuando llegan 3 acks duplicados. Para ello, reacciona igual que si fuese un *timeout*. Lo primero, retransmite el segmento que se perdió *fast retransmit*. Luego, disminuye el *ssthresh* (umbral de Slow Start inicial) al valor/tamaño original de la ventana de congestion (*cwnd*) y establece el tamaño de los ventana a 1 segmento como *Slow start*.

[4] TCP Reno es el sucesor de TCP Tahoe. Es una estrategia usada en TCP para controlar la congestión cuando llegan 3 acks duplicados. Para ello, reacciona igual que si fuese un *timeout*. Disminuye a la mitad el valor actual de *ssthresh* (umbral de Slow Start). Para cada nuevo ACK duplicado (el cuarto, quinto, sexto, etc.), la ventana de congestión aumenta en una unidad. Una vez que el cliente recibe el segmento perdido, TCP volverá a Slow Star si hay un timeout.