

Problemas de Arquitectura de Redes, Sistemas y Servicios
3º Ingeniería de Telecomunicación
Hoja de problemas 8

Problema 8.1: Se pretende construir un servicio de radio sobre internet en el que se enviará un canal de sonido comprimido de unos 50kbps. Se enviará el audio utilizando paquetes UDP de 500bytes de datos. Se pretende que el receptor, no sólo reproduzca el sonido en tiempo real, sino que también pueda grabarlo en el disco duro para poder escucharlo después. Por ello se decide que emisor y receptor utilizarán técnicas de transporte fiable para asegurar que todos los paquetes lleguen a su destino. Se plantea utilizar stop and wait.

Si el retardo de propagación a un receptor dado (incluyendo el tiempo de proceso y en los routers intermedios) es de aproximadamente 50ms.

- a) ¿Podemos conseguir la velocidad necesaria para reproducir en tiempo real utilizando stop and wait?
- b) Si utilizamos un protocolo que nos permite enviar una ventana de n paquetes mientras esperamos la confirmación de cada uno de ellos. ¿Cual es el valor de n que debemos configurar en nuestro protocolo, para conseguir la recepción adecuada del audio en tiempo real?

Problema 8.2: Se pretende construir un servicio de televisión sobre internet en el que se enviará un canal de video comprimido de menos de 150kbps. Se enviará el vídeo utilizando paquetes UDP de 1400bytes de datos de video. Se pretende que el receptor, no sólo muestre el video al usuario en tiempo real, sino que también pueda grabarlo en el disco duro para poder verlo después. Por ello se decide que emisor y receptor utilizarán técnicas de transporte fiable para asegurar que todos los paquetes lleguen a su destino. Se plantea utilizar stop and wait.

Si el retardo de propagación a un receptor dado (incluyendo el tiempo de proceso y en los routers intermedios) es de aproximadamente 50ms.

- a) ¿Podemos conseguir la velocidad necesaria para ver el video en tiempo real utilizando stop and wait?
- b) Si utilizamos un protocolo que nos permite enviar una ventana de n paquetes mientras esperamos la confirmación de cada uno de ellos. ¿Cual es el valor de n que debemos configurar en nuestro protocolo, para conseguir la recepción adecuada del vídeo en tiempo real?

Problema 8.3: En una línea de 2Mbps y 9000km utilizamos un protocolo stop-and-wait con paquetes de 500 bytes, el timeout para reaccionar a la pérdida de un paquete se configura a 200ms.

- a) Si no hay errores en el envío, ¿Cual es el throughput que conseguimos enviar al destino? ¿Cuanto tardaríamos en enviar un fichero de 100MB? ¿Cual es en ese caso la utilización de la línea?
- b) ¿Como cambian los resultados si aproximadamente el 5% de los paquetes se pierden?
- c) ¿Qué es mas eficaz para mejorar la velocidad, doblar el tamaño de paquete utilizado o disminuir a la mitad el valor del timeout?
- d) ¿Qué ocurriría si el timeout estuviera configurado a 40ms? ¿Cuál sería el throughput y la utilización entonces?
- e) Resuelva el mismo problema si la línea fuera de 256kbps en lugar de 2Mbps

Problema 8.4: Se utiliza un protocolo de transporte de ventana deslizante de tipo go-back-N que utiliza paquetes de datos de 500 bytes y paquetes de ACK de 60 bytes. El protocolo tiene configurada una ventana de 8 paquetes. En un escenario de enlace full-duplex de 10Mbps con $RTT=70ms$...

- a) ¿cual es la máxima velocidad de envío que permite el protocolo?
- b) Dibuje el diagrama espacio-tiempo que muestre el máximo throughput que permite el protocolo. ¿Cual es la utilización del canal en ese caso? ¿Cual es la utilización del canal de retorno?
- c) ¿Como cambian los resultados si configuramos la ventana a 32 paquetes sin confirmar?

Problema 8.5: El protocolo del problema anterior se utiliza en un escenario de ADSL. Supongamos que el canal de subida tiene una velocidad de transmisión de 2Mbps y el canal de bajada tiene una utilización de 128Kbps. Si utilizamos el protocolo para transmitir un fichero de 100MB desde un servidor con el que tenemos 70ms de RTT hasta el usuario residencial del ADSL.

- a) ¿Cuanto tiempo tardara en llegar el fichero en el mejor caso?
- b) ¿Qué ocurre si configuramos el protocolo con una ventana de 32 paquetes? ¿Cuales son la utilización de la subida y la bajada en ese caso?
- c) ¿Cual es la velocidad máxima que deberíamos contratar de subida para poder aprovechar la bajada completamente con este protocolo?

Problema 8.6 En un juego multijugador online la comunicación entre el servidor y los jugadores se realiza mediante el protocolo UDP. El programa de los jugadores envía los movimientos en paquetes UDP al servidor y el servidor envía a cada jugador la información de todo lo que ve en paquetes UDP. De esta manera los envíos tienen el mínimo retardo posible y si alguno de los paquetes se pierde simplemente el jugador se quedara congelado un momento.

Sin embargo se necesita también enviar cierta información de forma fiable. Cuando un jugador se acerca a una zona nueva hay que mandarle el mapa de esa zona y la pérdida de esos datos si que debe ser evitada. Se plantea que el servidor enviará los datos al jugador utilizando paquetes UDP con 1000 bytes de datos. El cliente confirmara cada uno de estos paquetes con un paquete UDP que ocupa 30 bytes y para simplificar el programa el servidor no envía el siguiente paquete hasta que recibe la confirmación. Para evitar pérdidas se utiliza un temporizador de 70ms en cada paquete que en caso de caducar dispara el reenvío. Si el jugador llega a la zona nueva cuando aun no ha recibido el mapa entero el juego se quedara congelado hasta que acabe de transferir el fichero.

Si el fichero de mapa de cada zona que queremos enviar al cliente ocupa 500kB y se configura el servidor para que dispare el envío del mapa cuando calcula que el jugador llegara a la zona nueva en 30segundos.

- a) ¿Cual es el máximo retardo extremo a extremo con los jugadores que podemos tolerar sin que haya problemas de carga del mapa?
- b) Si el round trip time típico con nuestros jugadores es de 65ms y un 1% de los paquetes se pierden. ¿Cuanto tiempo se quedará congelado el jugador esperando a recibir el mapa?
- c) ¿Cómo mejoraría el protocolo para que sea capaz de cargar el mapa en la situación de la pregunta (b) manteniendo toda la simplicidad posible del programa?

Problema 8.7: Tenemos un enlace serie entre dos PCs configurado a 115200bps full-duplex. Se construye un programa simple para enviar datos sobre el puerto serie que utiliza un protocolo stop and wait. Enviando bloques de 3000 bytes que el receptor contesta enviando ACKs. Los ACKs ocupan 25 bytes. Antes de empezar a transferir datos el programa estima el retardo para ello envía un paquete pequeño de 25 bytes que el receptor contesta con un ACK de 25 bytes. El tiempo medido por el programa con esta técnica es de 3.5ms.

- a) ¿Que throughput podemos alcanzar con ese programa sobre dicho enlace?
- b) ¿Cuanto tardaremos en enviar un fichero de 100MB?