

QoS 802.11

Práctica 2 – Gestión de Infraestructuras

Objetivo

Estudiar las ventajas que proporciona el esquema de QoS proporcionado por 802.11e.

Escenario

Nuestra red está formada por dos nodos que se comunican a través de un punto de acceso inalámbrico. Uno de los nodos envía datos al otro utilizando cuatro flujos UDP, cada uno con una categoría de acceso (AC) 802.11e distinta: voz, vídeo, *best effort* y *background*. Simularemos nuestra red primero con la QoS deshabilitada y luego con la QoS habilitada. Los nodos utilizan el modo PHY de 54 Mb/s y los flujos de tráfico tienen unas tasas originales (en bit/s) realistas: tanto el tráfico *background* como el *best effort* tienen tasas altas (24 y 28.8 Mb/s, resp.), el tráfico de vídeo una tasa menor (4.8 Mb/s) y el de voz otra aún menor todavía (100 Kb/s).

No debería haber diferencias de retardo ni velocidad entre los flujos de tráfico en el caso sin QoS porque la capa MAC no diferencia entre las categorías de acceso. En el caso con QoS, debería haber menos retardo y mayor velocidad para los paquetes de mayor prioridad.

Ejecución de la simulación

1. Cree un proyecto OMNeT++ yendo al menú *File > New > OMNeT++ Project...*, introduzca el nombre *XI-P2*, a continuación, seleccione *Empty project* y pulse el botón *Finish*.
2. Descargue del campus virtual los ficheros de la práctica. Abra un explorador de archivos de Ubuntu y colóquese en la carpeta que contiene dichos archivos. Seleccione todos los archivos y arrástrelos sobre el nombre del proyecto *XI-P2* en el OMNeT++ IDE.
3. Indique que el proyecto *XI-P2* utiliza el proyecto *inet-4.4*. Para ello, seleccione el nombre del proyecto *XI-P2* y vaya al menú *Project > Properties*. En el cuadro de diálogo que aparece, seleccione *Project References*, marque la casilla correspondiente a *inet4.4* y pulse el botón *Apply & Close*.
4. Cree una nueva configuración de ejecución para el proyecto que utilice *cmdenv* y ejecución multiproceso. Para ello, vaya al menú *Project > Properties*. Seleccione *Run/Debug Settings* en el cuadro de diálogo que aparece, pulse el botón *New...* e indique *OMNeT++ Simulation* para el tipo de configuración. En la casilla *Name* escriba *XI-P2*. En el apartado *Execution* fije ***Cmdenv como User Interface***, marque la casilla *Allow multiple processes* y fije a **1** la casilla ***Runs per process***. La casilla *Number of CPUs to use* debe fijarla al número de procesadores virtuales de la máquina virtual (puede consultarlos ejecutando *lscpu | egrep 'CPU'* en una terminal).
5. Ejecute la simulación pulsando el botón *Run*. Una vez concluida, los ficheros de resultados estarán contenidos en la carpeta *results*.

Análisis de resultados

Abra el fichero de análisis *Qos_py.anf*. En la pestaña *Charts* hemos definido tres gráficas: retardo extremo-a-extremo, variación del mismo y velocidad.

Retardo extremo-a-extremo

La Fig. 1 muestra el retardo extremo-a-extremo. Observe como en el escenario sin QoS todos los flujos experimentan un retardo (latencia) similar en el rango 0 – 0.25 segundos. Este retardo es pequeño para las necesidades del tráfico *background* y *best effort* pero, en cambio, es inasumible para el tráfico de vídeo y de voz.

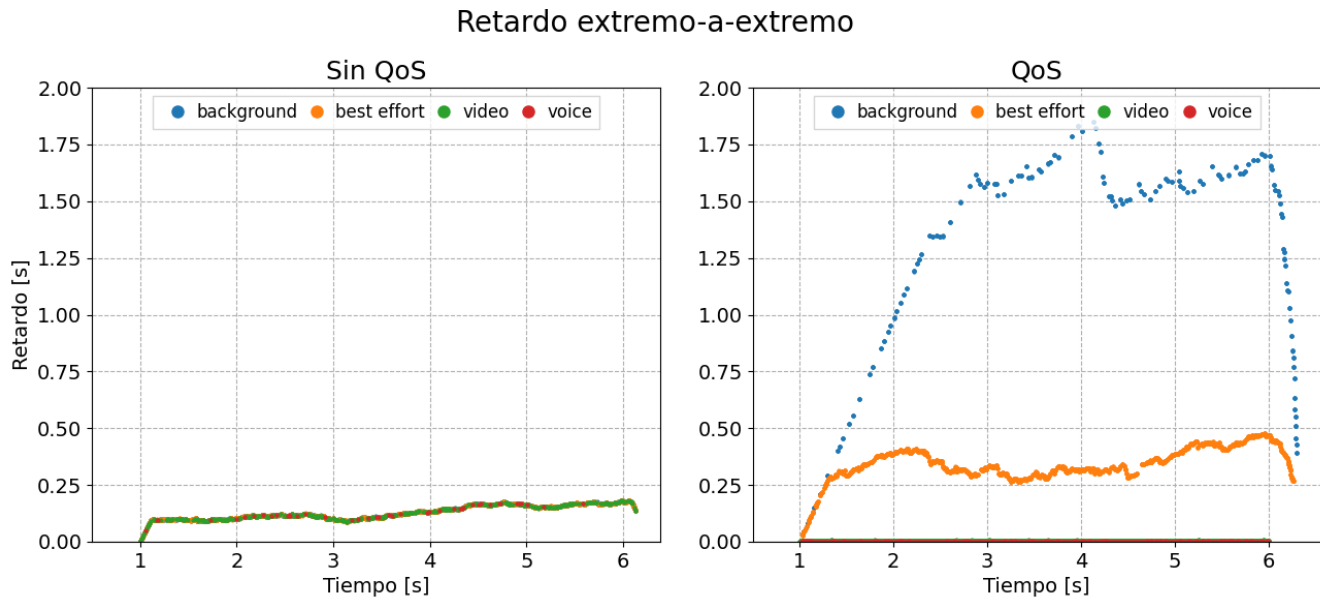


Fig. 1: Retardo extremo-a-extremo.

Cuando aplicamos el esquema de QoS proporcionado por la EDCF de 802.11e, el retardo de vídeo y voz pasa a ser mínimo, a costa de incrementar el del tráfico *best effort* y *background*. También garantizamos que no se pierde ningún paquete de los flujos de vídeo, voz y *best effort*, a cambio de que se puedan perder paquetes del flujo *background*.

Calculamos la media y la desviación típica del retardo en el intervalo [2 – 5] segundos para que los valores del principio y del final de la simulación no nos distorsionen el resultado. Para ello, primero obtenemos una copia del vector `endToEndDelay` de los módulos `QosShowcase.server.app[0]` (tráfico *background*), `QosShowcase.server.app[1]` (*best effort*), `QosShowcase.server.app[2]` (vídeo) y `QosShowcase.server.app[3]` (voz).

- Pinchamos en la pestaña *Browse Data*.
- Pinchamos en el título de la columna *Name* para ordenar la tabla por dicha columna.
- Ajustamos el ancho de las columnas para poder ver bien el nombre de los vectores.
- Seleccionamos el vector `endToEndDelay` de los módulos `QosShowcase.server.app[0]` a `QosShowcase.server.app[3]` con *Run number 0* (escenario sin QoS).
- Botón derecho → *Export Data* → *OMNeT++ Vector File...* y escribimos `SinQoS_retardo_25` en *Target file*, 2 en *Start time* y 5 en *End Time*. Marcamos la casilla *Open with default editor afterwards* y pulsamos *Finish*. En el cuadro de diálogo que sale a continuación pulsamos *Finish*.
- Repetimos los pasos anteriores, pero con los vectores con *Run number 1* (con QoS) usando `ConQoS_retardo_25` como nombre del fichero nuevo.

En los nuevos ficheros *SinQoS_retardo_25.anf* y *ConQoS_retardo_25.anf* podemos ver los cuatro vectores *endToEndDelay* correspondientes a los cuatro flujos y sus medias y desviaciones típicas. Pinche en la columna *Module* para ordenarlos por flujos de tráfico.

Preguntas

- Haciendo *zoom* sobre las gráficas de Retardo extremo-a-extremo, ¿Cuánto es el retardo del tráfico de vídeo y de voz en $t = 4s$ en el caso Sin QoS? ¿Y con QoS?
 SinQoS -> Video = 0.13s | Voz = 0.131s
 ConQoS -> Video = 0.001s | Voz = 0.001s
- Rellene la siguiente tabla con los datos mostrados en la pestaña *Browse Data* de los ficheros *SinQoS_retardo_25.anf* y *ConQoS_retardo_25.anf*. Use una precisión de dos decimales.

Flujo	Escenario	Retardo Media [ms]	Retardo Desv. Típica [ms]
<i>Background</i>	Sin QoS	127.28	23.48
	QoS	1516.72	194.67
<i>Best effort</i>	Sin QoS	126.57	23.32
	QoS	327.26	36.43
Vídeo	Sin QoS	127.17	24.07
	QoS	1.93	1.07
Voz	Sin QoS	131.95	24.29
	QoS	1.03	0.65

Velocidad

La Fig. 2 muestra la velocidad alcanzada por los distintos flujos. Observe como en el escenario sin QoS la velocidad de cada flujo depende fundamentalmente de su tasa original, ya que los cuatro flujos compiten entre sí para transmitir sin que ninguno tenga prioridad sobre otro. Esto provoca que los flujos de vídeo y voz no sean transmitidos a la velocidad suficiente.

Cuando aplicamos el esquema de QoS proporcionado por la EDCF de 802.11e, el tráfico se prioriza en el orden voz, vídeo, *best effort* y *background*. La velocidad de la voz y el vídeo pasan a ser máximas y se igualan a las tasas originales (100 Kb/s y 4.8 Mb/s), a costa de reducir la tasa de los flujos *best effort* y *background*.

Calculamos la media y la desviación típica de la velocidad en el intervalo [2 – 5] segundos de la misma forma que hicimos para el retardo, usando *SinQoS_velocidad_25* y *ConQoS_velocidad_25* como nombres para los nuevos ficheros. El vector que nos interesa ahora es el llamado *throughput*.

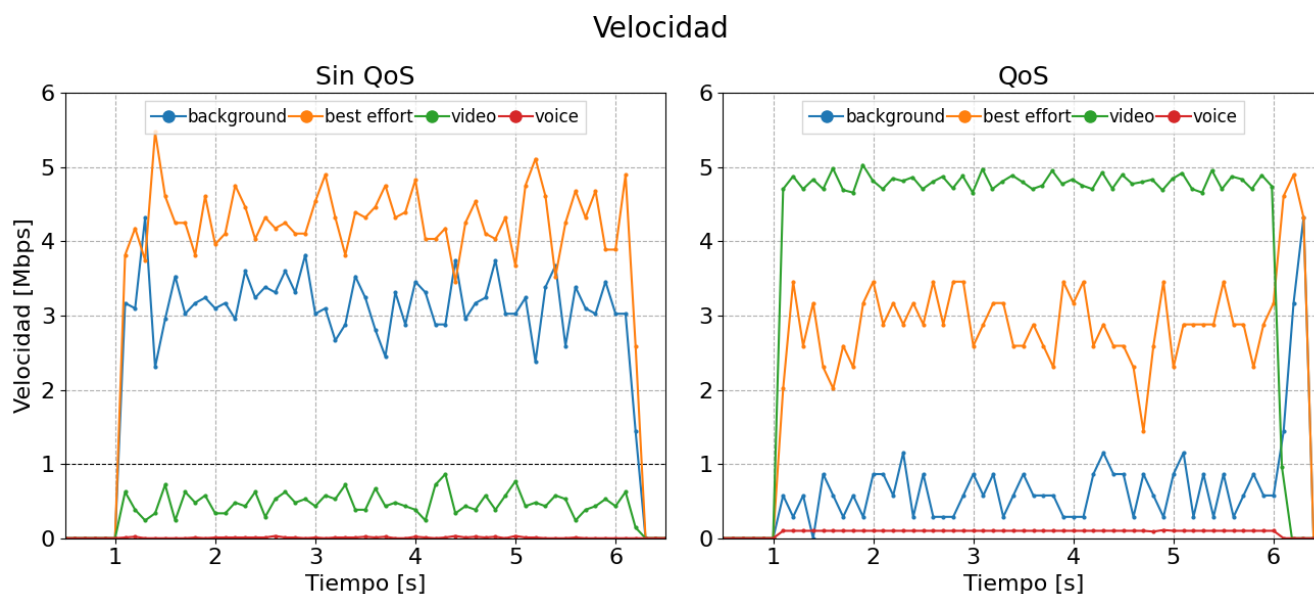


Fig. 2 Velocidad.

Preguntas

- Haciendo *zoom* sobre las gráficas de Velocidad, ¿Cuánto es la velocidad del tráfico de vídeo y de voz en $t = 4s$ en el caso Sin QoS? ¿Y con QoS?.
- Rellene la siguiente tabla con los datos mostrados en la pestaña *Browse Data* de los ficheros *SinQoS_retardo_25.anf* y *ConQoS_retardo_25.anf*. Utilice una precisión de tres decimales.

Flujo	Tasa fuente [Mb/s]	Escenario	Velocidad Media [Mb/s]	Velocidad Desv. Típica [Mb/s]
Background	24	Sin QoS		
		QoS		
Best effort	28.8	Sin QoS		
		QoS		
Vídeo	4.8	Sin QoS		
		QoS		
Voz	0.1	Sin QoS		
		QoS		

- Calcule la suma de las velocidades medias en ambos escenarios. ¿En qué escenario es mayor? ¿Por qué dicha suma no es igual a la tasa de 54 Mb/s utilizada?

Variación instantánea del retardo (*jitter*)

La Fig. 3 muestra la variación instantánea del retardo, también denominada *jitter*. Observe como en el escenario sin QoS el *jitter* instantáneo de los distintos flujos es similar. Esto supone valores del *jitter* demasiado altos para los flujos de vídeo y voz experimenten. Cuando el *jitter* es pequeño, podemos compensarlo en recepción con un pequeño búfer. Cuando es muy alto, ese búfer ha de ser grande y, sobre todo, la experiencia de usuario para servicios como la voz y el vídeo se ve impactada gravemente.

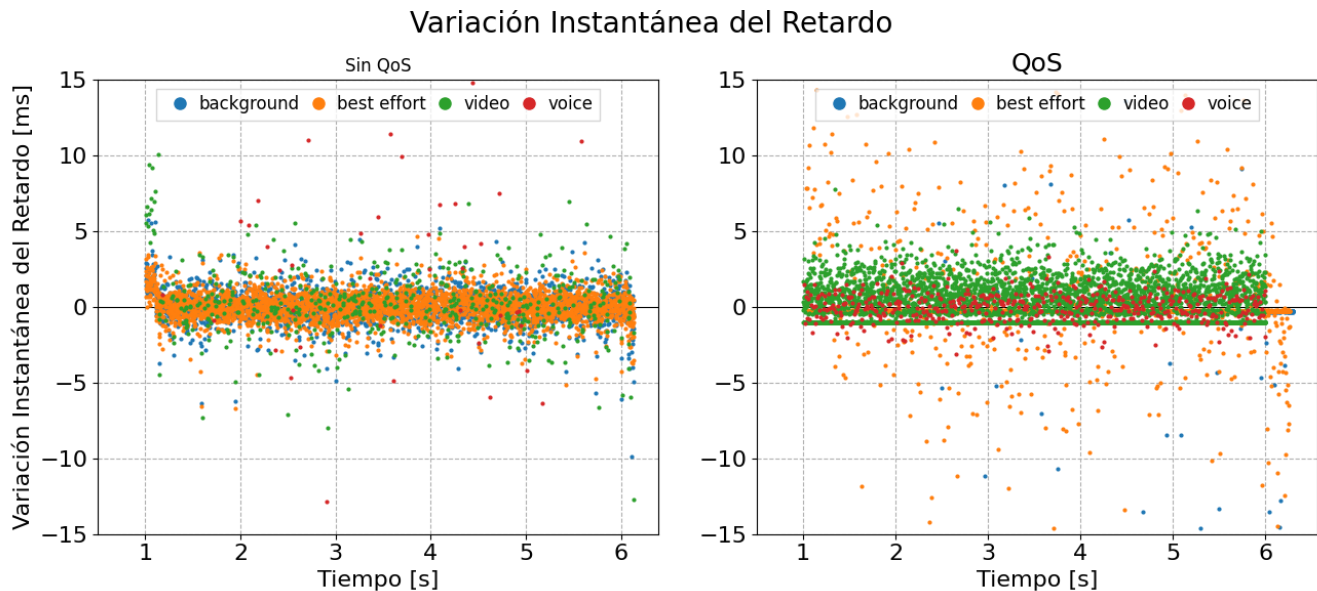


Fig. 3: Variación instantánea del retardo (*jitter*).

Cuando aplicamos el esquema de QoS proporcionado por la EDCF de 802.11e, el tráfico se prioriza en el orden voz, vídeo, *best effort* y *background*. El *jitter* de la voz y el vídeo se reduce al mínimo a costa de aumentar el *jitter* de los flujos *best effort* y *background*.

Preguntas

6. A la vista de la gráfica del *jitter* para el caso con QoS, ¿Experimentan la voz y el vídeo el mismo *jitter*? ¿Por qué?

Conclusión

Al aplicar el esquema de QoS proporcionado por la EDCF de 802.11e, el tráfico se prioriza en el orden voz, vídeo, *best effort* y *background*. El retardo de vídeo y voz pasa a ser mínimo, a costa de incrementar el del tráfico *best effort* y *background*. La variación instantánea del retardo (*jitter*) también pasa a ser mínima. Asimismo, garantizamos que no se pierde ningún paquete de los flujos de vídeo, voz y *best effort*, a cambio de que se puedan perder paquetes del flujo *background*.

La velocidad de la voz y el vídeo pasan a ser máximas y se igualan a las tasas originales (100 Kb/s y 4.8 Mb/s), a costa de reducir la velocidad de los flujos *best effort* y *background*.