

Práctica 2: Medidas de rendimiento

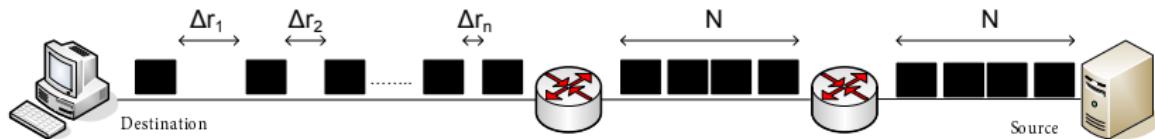
Redes Multimedia
Universidad Autónoma de Madrid

16 de octubre de 2025

- Para poder desplegar un servicio multimedia sobre una red IP necesitamos conocer la calidad de la misma
- Para describir la calidad de la red usamos un conjunto de parámetros de calidad de servicio (QoS)
 - Ancho de Banda
 - Pérdidas
 - Retardos
 - Jitter
 - Otros parámetros (por ejemplo disponibilidad)
- A la hora de contratar/ofertar un servicio se acuerdan los valores máximos y típicos para los parámetros de QoS en un contrato llamado Service Level Agreement (SLA)
- Tanto si estamos en el lado del proveedor como en el lado del cliente es importante conocer y saber medir estos valores

- Existen diversos métodos:
 - Descarga de ficheros
 - Pares de paquetes
 - Trenes de paquetes
- En esta práctica vamos a implementar el método de trenes de paquetes por ser el más fiable y menos intrusivo de los anteriores
- Los anteriores métodos inyectan tráfico en la red para medir y se conocen como medidas activas
- Existen otros métodos que no introducen tráfico en la red para medir los parámetros. A estas técnicas se las conoce como medidas pasivas.

- Enviamos N paquetes de longitud L bits back-to-back (a tasa máxima) desde un origen a un destino
- En el destino analizamos la dispersión entre paquetes para estimar el ancho de banda
- Todos los paquetes que enviemos deben contener:
 - Número de secuencia: para detectar pérdidas
 - Tiempo de salida del paquete: para calcular el retardo



- ¿ Cómo estimar los parámetros de QoS?
 - Ancho de banda medio: $\frac{(N-1)L}{r_N - r_1}$
 - Ancho de banda instantáneo: $\frac{L}{\Delta r_i}$
 - OWD (One-Way Delay): En recepción se marca el tiempo de llegada y se resta del tiempo de salida (contenido en el paquete)
 - Jitter: Estimación a partir del OWD. Cálculo de la desviación estándar de los OWD
 - Pérdidas: Contador número de paquete dentro de cada paquete
 - ¿Podemos obtener anchos de banda negativos? ¿y retardos?
- * Δr_i es el interarrival entre el paquete $i+1$ e i y r_i es el tiempo de llegada del paquete i

- Conocer e implementar el método de medida de trenes de paquetes para estimar ancho de banda, retardo, jitter y pérdidas
- Realizar medidas y aprender a valorar los parámetros de QoS
- Aplicar las medidas realizadas a un caso de uso concreto (VoIP)

- Se proporciona un código que:
 - Lee por argumentos una IP y puerto destino, una longitud de tren y longitud de paquete
 - Crea un socket y envía a tasa máxima paquetes RTP de medida. Cada paquete tiene:
 - Número de secuencia incremental
 - Tiempo de envío del paquete: con resolución de decenas de microsegundos porque en el timestamp de RTP no cabe más.
 - SSRC: Usamos como identificador el número total de paquetes que debe de tener el tren
- Modificaciones a realizar:
 - Hacer comprobaciones y adaptar tamaños en función de si usamos localhost o no

- Se proporciona un código que:
 - Crea un socket y escucha en una IP y puerto
 - Recibe paquetes de medida (enviados por clienteTren.py) y añade su tiempo de llegada y su contenido a una lista
 - Pasados unos segundos sin tráfico procesa los paquetes recibidos
 - Para cada paquete extrae los campos de la cabecera RTP
- Modificaciones a realizar:
 - Añadir código necesario para calcular:
 - Ancho de banda medio y máximo,mínimo e instantáneo (por cada par de paquetes)
 - Retardo en un sentido medio, máximo y mínimo
 - Variación del retardo (como la desviación estándar de los retardos)
 - % de pérdidas

- Para medir el OWD, pérdidas y Jitter **NO** se puede saturar la red enviando paquetes a máxima tasa. Para ello necesitamos poder enviar a menos tasa
- Modificar clienteTren.py para permitir un nuevo argumento por línea de comandos que sea la tasa a la que emitir
- Para emitir a una tasa determinada solo hay que calcular el tiempo entre un paquete y el siguiente y usar time.sleep() para esperar
- $\Delta r = \frac{L}{Tasa}$

- Modificación del programa que envía paquetes clienteTren.py (1 punto)
- Modificación del programa que recibe paquetes servidorTren.py (1 punto)
- Realización de medidas en localhost y entre 2 equipos de laboratorio. Análisis de las medidas (1.5 puntos)
- Implementación del programa que envía a tasa variable clienteTren2.py (1 punto)
- Realización de medidas y estimación de los parámetros de QoS proporcionados por el emulador (1.5 puntos)

- Captura de tráfico y cálculo de parámetros de QoS usando Wireshark (1.5 puntos)
- Aplicación de medidas al dimensionado y despliegue de un servicio de VoIP en base a los parámetros anteriormente calculados (1.5 puntos)
- Simulación del servicio anteriormente dimensionado (1 punto)