**Trabajo - Prioridades y Colas**

**Grupo 12**

Planificación y Simulación de Redes

Departamento de Ingeniería Telemática

Beatriz Carretero Parra

Álvaro Espejo Muñoz

Alberto Flores Pereira

Miguel Alejandro García Benítez

Índice

[1 Definición del problema 4](#_Toc473058389)

[2 Planteamiento del proceso 5](#_Toc473058390)

[2.1 Ns-3 5](#_Toc473058391)

[3 Definición del sistema 6](#_Toc473058392)

[3.1 Abstracción 6](#_Toc473058393)

[3.2 Simplificación 6](#_Toc473058394)

[VozIP 7](#_Toc473058395)

[Interactive Video 7](#_Toc473058396)

[4 Formulación del modelo conceptual 7](#_Toc473058397)

[4.1 Simulacion.cc 7](#_Toc473058398)

[4.2 VoIP.cc 8](#_Toc473058399)

[4.3 HTTP.cc 8](#_Toc473058400)

[4.4 FTP.cc 9](#_Toc473058401)

[4.5 Observador.cc 9](#_Toc473058402)

[5 Diseño preliminar de los experimentos 9](#_Toc473058403)

[6 Preparación de los datos de entrada 9](#_Toc473058404)

[7 transcripción del modelo 10](#_Toc473058405)

[8 Verificación del diseño y validación de los resultados 11](#_Toc473058406)

[9 diseño final de los experimentos 11](#_Toc473058407)

[10 Experimentación 12](#_Toc473058408)

[11 Análisis e interpretaciones de los resultados 13](#_Toc473058409)

[12 Documentación del proceso 14](#_Toc473058410)

[12.1 Mejoras 14](#_Toc473058411)

[13 Conclusiones 15](#_Toc473058412)

# Definición del problema

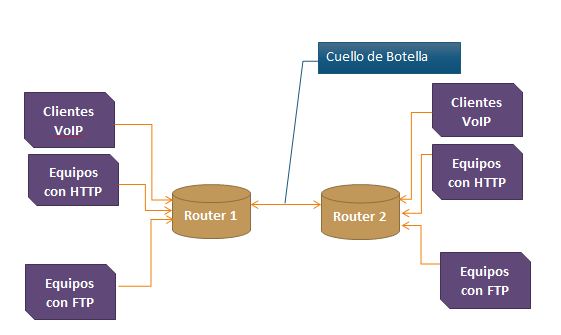
EL objetivo de la simulación es el estudio de las colas con prioridades para analizar como se comporta el tráfico de diferentes tipos y como asignar esas prioridades para alcanzar unos mínimos de QoS para aquellos tipos de tráficos que lo requieran.

Vamos a fijarnos en esta tabla de recomendación de Cisco

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Application** | **Layer 3 Classification** | | | **Layer 2**  **CoS/MPLS EXP** | |
|  | **IPP** | **PHB** | **DSCP** |  |  |
| IP Routing | 6 | CS6 | 48 | 6 |  |
| Voice | 5 | EF | 46 | 5 |  |
| Interactive Video | 4 | AF41 | 34 | 4 |  |
| Streaming-Video | 4 | CS4 | 32 | 4 |  |
| Locally-Defined Mission-Critical Data (see note below) | 3 | — | 25 | 3 |  |
| Call-Signaling (see note below) | 3 | AF31/CS3 | 26/24 | 3 |  |
| Transactional Data | 2 | AF21 | 18 | 2 |  |
| Network Management | 2 | CS2 | 16 | 2 |  |
| Bulk Data | 1 | AF11 | 10 | 1 |  |
| Scavenger | 1 | CS1 | 8 | 1 |  |
| Best Effort | 0 | 0 | 0 | 0 |  |
| **Table 1-1** **Cisco QoS Baseline** | | | | | |

Una vez realizada la simulación con los parametros y asignadas las prioridades correctamente podremos observar como dentro del escenario los paquetes que sean del tráfico más prioritario podrán alcanzar la QoS necesaria para su tipo de tráfico.

Así mismo, podríamos dar datos como la mínima velocidad del enlace para conseguir cumplir con la calidad del servicio requerida por VoIP (que es la más restrictiva con un 99% de paquetes correctos) o ver cuánto error como máximo tendría que tener el canal para cumplir con esa QoS.



# Planteamiento del proceso

Para realizar la simulación utilizaremos la herramienta de Ns3 así como un tiempo aproximado de un mes para el trabajo

## Ns-3

NS3 es un simulador de redes escrito en C++, con un codigo fuente de más de 300.000 lineas con una licencia GNU General Public License v2, lo que garantiza un codigo libre. Pueda ser modificado y compartido de nuevo, pero solo de forma libre y bajo ningun concepto convertirlo en un software privativo.

Está constituido por módulos que definen dispositivos, protocolos y utilidades reales. Es capaz de generar simulaciones de gran escala basadas en eventos discretos programados en una cola de tiempo de ejecución.

En este proyecto y debido a la necesidad de un tipo de cola **con prioridades** (***PFifoQueueDisc***) que no se encuentra disponible en la versión utilizada en clases, usaremos la versión **3-26 de ns3.**

# Definición del sistema

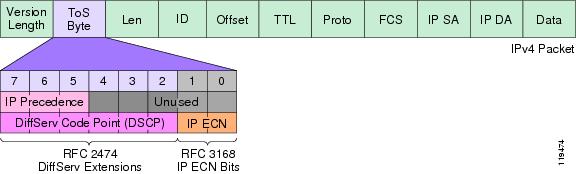
## Abstracción

El ámbito del problema de la simulación se basa en altas cantidades de tráfico de diferentes fuentes y de distinto tipo. Esto provoca que en los puntos donde existe un sumidero de tráfico para salir de una red a otra podemos encontrar cuellos de botella que ciertos tipos de tráfico no pueden permitirse.

## Simplificación

Para simplificar el modelo para realizar las simulaciones utilizaremos dos redes en espejo con 3 equipos generando tráfico Voip, Http y Ftp y 1 Router. El resultante es un cuello de botella entre los routers que necesita de tráfico con prioridades para alcanzar QoS mínima en Voip

El campo Type Of Service viaja como una etiqueta en la cabecera IP, y cuando se procesa el mensaje, se calcula su determinado DSCP.



Como únicamente usamos 3 tipos de tráfico, la tabla de ToS queda reducida a:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tráfico** | **Nombre de la categoría de prioridades** | **ToS (Type of Service)** | **DSCP** |
| VozIP | Interactive Video | 184 | 46 |
| FTP | Bulk Data | 40 | 10 |
| HTTP | Best Effort | 0 | 0 |

# Formulación del modelo conceptual

En este proyecto como hemos mencionado anteriormente, vamos a utilizar varios tipos de servicios. A continuación, se discutirán los ficheros y clases que permitirán tanto generar, como recibir y redirigir el tráfico para conseguir nuestros objetivos.

## Simulacion.cc

Este será nuestro fichero principal contendrá la función *simulación* que se encargará de generar el escenario. Nuestra intención ha sido que el proyecto sea muy personalizable según la necesidad del quien lo ejecute. Es por ello, que el cliente puede decidir a través de la línea de comandos los siguientes elementos:

* **Tipo de gráfica**: Podrá elegir si quiere que se genere una gráfica que muestre el porcentaje de paquetes correctos en función del retraso del enlace punto a punto o una gráfica que muestre el porcentaje de paquetes correctos según el tamaño de la cola.
* **Número de nodos en las redes LAN**: Donde se podrá elegir el número de nodos totales, y no sólo eso, si no la cantidad de nodos que generen tráfico HTTP, VoIP y FTP que deseamos.
* **Capacidad de las LAN**: Se puede configurar las tasas de las redes, tanto la de la izquierda como la de la derecha.
* **Retardo**: Configuración del retardo de las LANs.
* **Retardos máximos y mínimos para el canal p2p**: Que será necesario para la generación de la gráfica 1.
* **Tamaño mínimo y máximo de la cola**: Para poder determinar el funcionamiento de la red, y para la generación de la gráfica 2.

Si no se configuran estos valores, se podrán los valores por defecto.

A continuación, se llamará a la función simulación que se encargará de montar el escenario (incluida la cola *PFifoQueueDisc*) según los parámetros recibidos.

Brevemente, ya que en el código se ve de manera más detallada, comentaremos la función *simulación.*

La función se encarga de montar el escenario, esto es creación del número de nodos que se especifique y de los objetos de tipo VoIP, HTTP y FTP para la emisión de datos, así como la creación de sumideros para la recepción de tráfico.

Es aquí además donde asignamos las prioridades según el tipo de servicio y donde creamos los objetos de tipo Observador y nos suscribimos a las trazas para obtener los datos que necesitaremos en las gráficas.

## VoIP.cc

Será nuestra clase para los dispositivos VozIp, que heredará de la clase *OnOffHelper.*

## HTTP.cc

Será nuestra clase para los dispositivos que envíen paquetes HTTP, que heredará de la clase *OnOffHelper.*

## FTP.cc

Esta clase se utilizará para generar tráfico FTP y recibirá los parámetros al llamar al constructor. Hereda de la clase *OnOffHelper.*

## Observador.cc

El observador nos permitirá obtener las estadísticas y datos que servirán para generar las gráficas. Las funciones de Observador se activarán con la llegada de nuevos paquetes o bien con el envío de los mismos.

# Diseño preliminar de los experimentos

Según los parámetros de la simulación que comentaremos en el siguiente punto, los valores que esperamos obtener en la simulación son nada menos la demostración de que efectivamente, nuestro sistema está actuando con prioridades y qué valores mínimos de retardo y de cola necesitaremos para cumplir los QoS de cada uno de nuestros servicios.

Para ello, iremos variando el retraso y el tamaño de cola en nuestras gráficas, y comprobaremos con qué valor se cumplen las restricciones de calidad de servicio, y es el que el cliente (operadora) necesitará para orientar en torno al resultado su plan de negocio y sus futuras inversiones.

# Preparación de los datos de entrada

Este proyecto permite tiene parámetros de entrada que o bien están fijos o bien son configurados por la línea de comandos.

Respecto a los fijos, tenemos:

* **Valores de la t-student**, que se podrá modificar a través del código.
* **El tiempo de inicio y de fin de los dispositivos** (que también se utilizará para definir el tiempo en el que se para la simulación).
* **El número de simulaciones por cada punto y el número de puntos de nuestra gráfica**.
* **Los ToS** de cada tipo de tráfico.
* **El puerto** que usarán los dispositivos para enviar o recibir.

En la línea de parámetros nos encontramos los siguientes elementos configurables:

* **Opción de elegir la gráfica** a generar (o bien con eje X el tamaño de cola o retardo del canal punto a punto)
* **Número de nodos en las redes LAN**
* **Número de nodos que usan cada tipo de tráfico**
* **Capacidad de la red LAN de la izquierda y capacidad de la de la derecha**
* **Capacidad del enlace p2p**
* **Tiempo de retardo de los enlaces de la LAN de la izquierda y derecha.**
* **Tamaño de cola mínima y máxima para generación de gráficas.**
* **Retraso del canal p2p.**

Hemos decidido añadir tantos parámetros configurables para que el usuario tenga más flexibilidad a la hora de probar parámetros que le sean útiles para conseguir un resultado esperado.

# transcripción del modelo

Vamos a usar Ns3 con varios de los Helper que nos proporciona y como ya se han explicado en el apartado 4 los modelos usados se hace una referencia a ellos directamente.

[Simulacion.cc](#_Simulacion.cc)

# Verificación del diseño y validación de los resultados

Para verificar el diseño y comprobar como los resultados de simulación se ajustan a los valores teóricos calculados considerando parámetros como velocidad del canal, tiempo de on y off de los dispositivos y tamaños de paquetes hemos considerado los siguientes casos:

1. Los equipos están correctamente configurados y la cola está encolando los paquetes según su prioridad. Para ello hemos realizado varias simulaciones aumentando progresivamente el tamaño de la cola y vemos como los que salen siempre son los de la prioridad más alta independientemente del orden de llegada.
2. El tamaño de cola no está correctamente ajustado. Al aumentar el tiempo de simulación se comprueba que paquetes de baja prioridad no llegan porque son descartados.

Considerando los resultados de las simulaciones y la verificación del modelo podemos llegar a la conclusión de que nuestro modelo está correctamente diseñado y funciona como debería.

# diseño final de los experimentos

Para obtener una información concluyente pero lo más rápido posible se han diseñado de forma que con pocos puntos se acabe la simulación

Se asume que a tráfico constante en el periodo que se está cursando tráfico VoiP este siempre será el prioritario y no tendrá un régimen transitorio acusado.

En caso de simular con muchos más puntos se vería que él % es prácticamente el mismo.

# Experimentación

Dentro de los experimentos las condiciones iniciales son las más importantes, estas determinaran que prioridad tendrá el tráfico y será el valor principal que modificará y dará importancia al análisis de los resultados

Los más importantes y qué mas afectan son :

ToS de los distintos tipos de tráfico

TOS\_VOIP 184

TOS\_FTP 40

TOS\_HTTP 0

Tiempos de Ton,Toff y valores para max y min para sacar valores aleatorios de VoiP

minHablando=2.0; maxHablando=5.0; minSilencio=0.5; maxSilencio=2.5;

minOnHTTP=4.4;maxOnHTTP=6.2; minOffHTTP=0.8; maxOffHTTP=1.6;

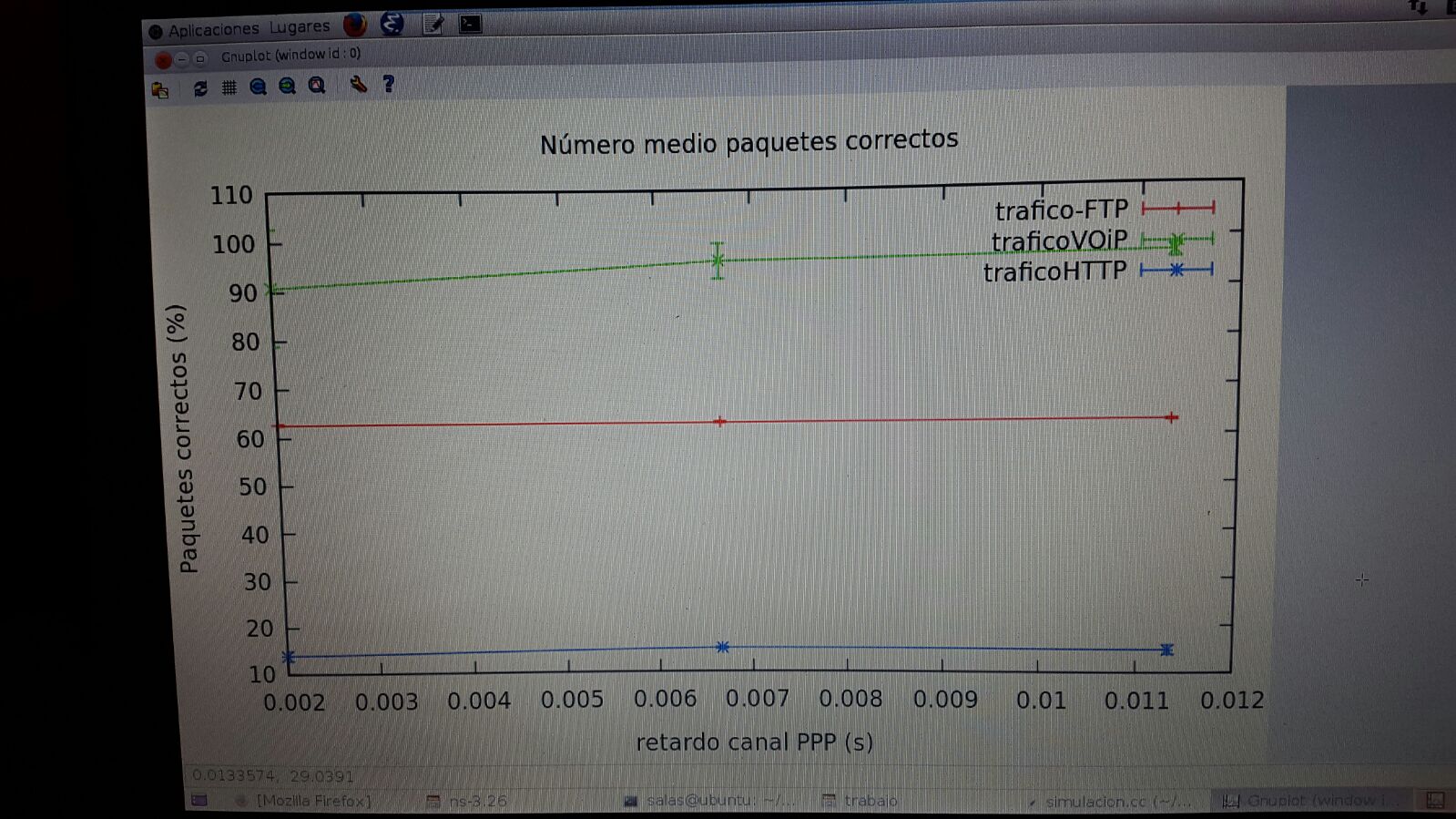
El resto de valores como los retardos y las tasas están definidos dentro del código, como son necesarios pero no son el punto principal del estudio no se les incluye.

Las simulaciones no están pensadas para cambiar el valor del ToS en la propia simulación por lo que estas no variarán dentro de una misma ejecución, en cambio los valores cambian según los retardos que se introducen al enlace entre los dos routers, viendo como varía la cantidad de paquetes perdidos cuando este aumenta.

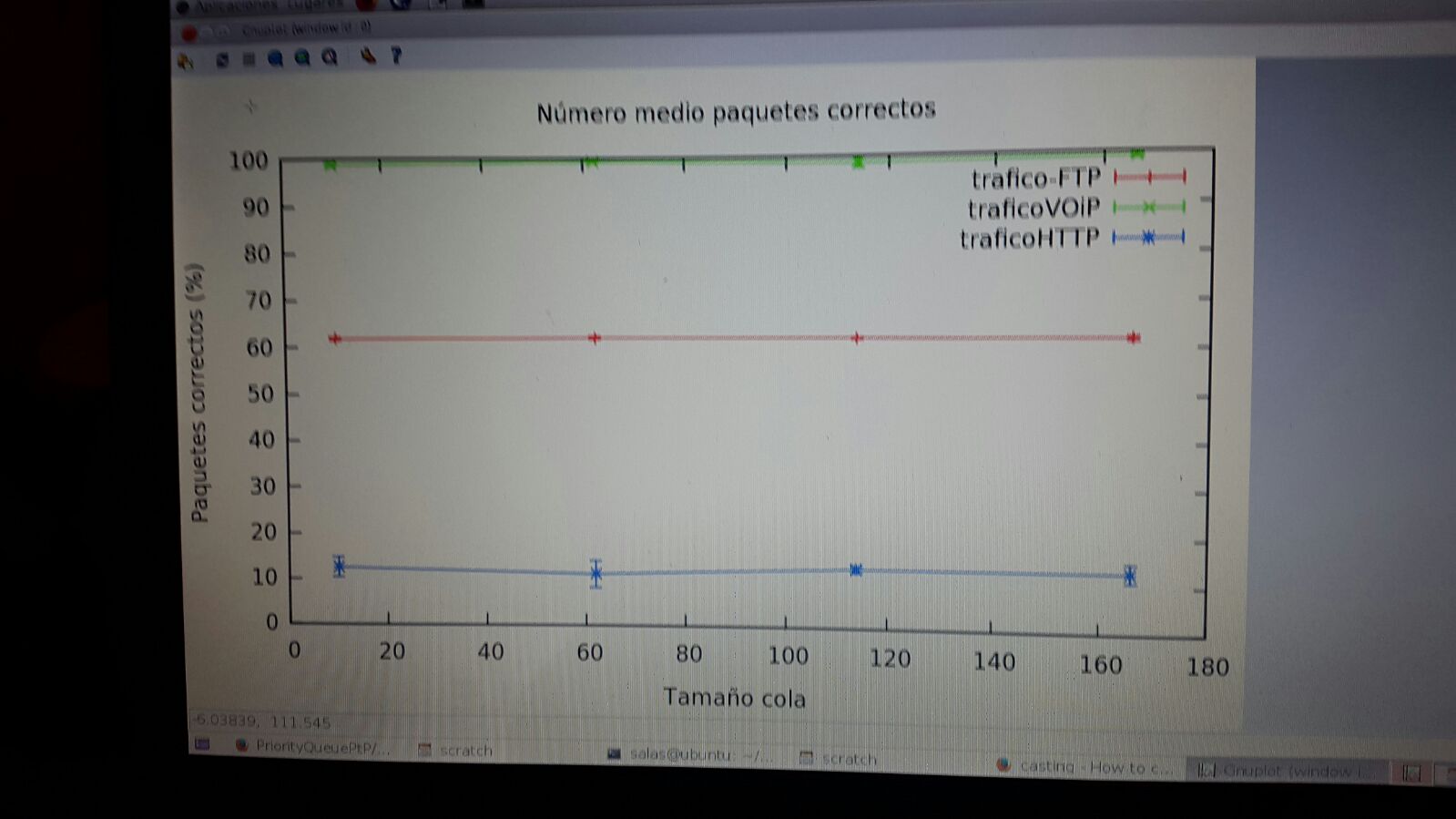
La simulación finaliza no hasta alcanzar un objetivo, esta dura tanto como puntos queramos añadir a la curva.

# Análisis e interpretaciones de los resultados

Los resultados son las siguientes gráficas:



Como el T\_ON es muy grande en el tráfico HTTP este sale constante



# Documentación del proceso

## Mejoras

El tráfico por prioridades y el uso de colas abren un sinfín de posibilidades de implementaciones y escenarios.

Este escenario puede mejorar y adaptarse a las necesidades de un cliente en muchos aspectos:

1. En primer lugar, se podría hacer una comparativa de colas en la cual se podrían comparar la cantidad de paquetes útiles y elegir la mejor cola en consecuencia. Esta idea se planteó en un primer momento, pero fue descartada ya que ns3 no ofrece más colas prioritarias que la PFifoQueueDisc, y el hecho de usar colas como FIFO o RED supondrían que claramente no se cumpliría con las especificaciones de calidad de servicio de servicios como VoIP.
2. Añadir más tipos de tráfico: Es otra de las mejoras, posibles. En este proyecto se han elegido los tipos de tráfico más significativos para probar las prioridades. Sin embargo, también existen más tipos de tráfico como Streaming, que tiene otra prioridad y por lo tanto etiqueta de ToS distinta.
3. También se podrían medir más parámetros de calidad de servicio en los que no sólo se encuentran el porcentaje de paquetes útiles

# Conclusiones

Con este proyecto hemos tenido un primer acercamiento de cómo funcionan las colas de redes hoy en día y cómo los tráficos prioritarios siempre serán los primeros en salir de la cola.

Además, gracias a este proyecto hemos aprendido más sobre la configuración de colas y su programación en ns3 ya que en clase sólo hemos tratado con la cola FIFO.

También, hemos adquirido más conocimientos sobre los tipos de prioridades que existen, sus respectivos DSCP, y configurar su ToS en consecuencia.

Por último, hemos aprendido más sobre Wireshark ya que nunca lo habíamos usado con prioridades anteriormente. Aunque el ToS aparezca en la cabecera IP, no queríamos ir paquete por paquete comprobando su prioridad, y finalmente hayamos la manera de mostrar una columna que mostraba las prioridades y etiquetas DSCP para comprobar que los resultados eran los esperados.