**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 NS3 ( Network Simulator) 2](#_Toc472993771)

[2 Objetivo 2](#_Toc472993772)

[3 Escenario 3](#_Toc472993773)

[4 Configuraciones 4](#_Toc472993774)

[5 Simulaciones 4](#_Toc472993775)

[6 Validez de los Resultados 4](#_Toc472993776)

[7 Conclusiones 5](#_Toc472993777)

# NS3 ( Network Simulator)

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.

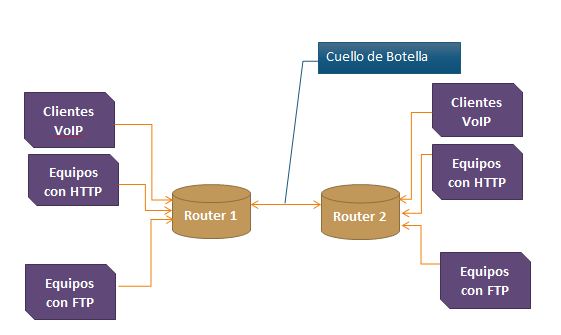
Está constituido por módulos que definen dispositivos, protocolos y utilidades reales. Es capaz de generar simulaciones de gran escala basadas en eventos discretos programados en una cola de tiempo de ejecución.

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.**

# Escenario

El escenario planteado podría asemejarse al de dos grandes compañías operadoras, unidas por un enlace, por el cual pasa tráfico HTTP, VozIP y FTP en ambas direcciones.

Tendrá la siguiente estructura:



Como hemos mencionado anteriormente, simularemos un escenario que podría tratarse perfectamente del de dos operadoras con sus respectivos tráficos (generados por clientes) y que necesitan comunicarse entre ellas y reenvían tráfico constantemente buscando el destinatario.

Como muestra la imagen, nos interesaremos por tráfico VoIP, tráfico HTTP y tráfico FTP, ya que son tráficos que se tratan de manera distinta, porque tienen diferentes prioridades.

En el escenario nos encontramos dos routers, unidos por un enlace, que serán los encargados del reenviar el tráfico a otra LAN.

Por otro lado, a las otras interfaces de cada routers tenemos nuestros dispositivos clientes, que se encargarán de enviar tráfico y de recibirlo, teniendo como ruta por defecto el router de su LAN.

Los componentes del escenario por defecto son por tanto:

* Clientes:
  + VoIP
  + HTTP
  + FTP
* 2 Routers conectados a través de un enlace.

Para facilitar las cosas, hemos decidido que cada cliente, también pueda actuar como servidor para garantizar la respuesta a posibles peticiones de la otra operadora.

Por tanto, cada cliente, periódicamente envía tráfico del tipo que corresponda, pero si recibe peticiones de tráfico de ese tipo, también las responderá. Esto lo hemos decidido para hacer un escenario básico y fácil de implementar.

Anteriormente, hablamos, de escenario por defecto, y es que, el escenario que muestra la foto es el escenario cuando no indicamos valores pasados por parámetros, ya que, hemos añadido la funcionalidad de permitir elegir el número de equipos con los que queremos simular cada tráfico.

# Objetivo

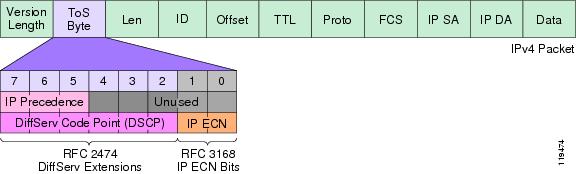
Como ya sabemos, hay grandes diferencias entre los tipos de tráfico que utilizaremos, y la mayor de ellas es la calidad de servicio de cada uno.

Por ejemplo, VozIP necesita un porcentaje de paquetes correctos igual o mayor que un 99%, al contrario que HTTP y FTP.

Es por ello que catalogaremos nuestros servicios en 3 grupos al que se le asignarán distintas prioridades según las características del tráfico:

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

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



Para poder procesar este tipo de paquetes es necesario dispositivos que estén adaptados para la lectura de etiquetas, es por ello que dotaremos de nuestros dispositivos frontera de una cola de prioridades.

Ya que nuestro escenario está orientado a una compañía operadora que ofrece servicios de este tipo, el principal objetivo de este proyecto además de medir el comportamiento del escenario según los datos de entrada (como son número de equipos, retrasos del canal, velocidad…), que explicaremos más adelante cuales son, es comprobar qué calidad de servicio de cada uno de los tráficos obtenemos variando los parámetros según la operadora requiera.

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.

Estas simulaciones con esta temática las consideramos muy interesantes ya que cualquier operadora necesita conocer la calidad de su tráfico y cómo se comporta según los distintos parámetros para realizar posibles mejoras, solucionar problemas ya existentes y continuar atrayendo a nuevos clientes.

# Configuraciones

Dependiendo de la programación final

|  |  |  |
| --- | --- | --- |
| Trafico | Nodo Destino | Prioridad |
| VoIP | Router 2 | 1 |
| Streaming | Router 2 | 2 |
| HTTP | Router 2 | 3 |
| FTP | Router 2 | 4 |
| VoIP | Router 2 | 2 |
| Streaming | Router 2 | 1 |
| HTTP | Router 2 | 3 |
| FTP | Router 2 | 4 |

# clases de la simulación

En las simulaciones hemos usado la clase PfifoFastQueueDisc, que usa tres colas FIFO, acordes a las tres bandas que posee, basadas en prioridades dadas a los paquetes. Los paquetes de una banda con mayor prioridad siempre saldrán de la cola antes que los paquetes de otra banda menos prioritaria.

Hemos recreado el protocolo VoIP y simulado una comunicación real de este protocolo entre dos usuarios, intentando conseguir más del 99% de los paquetes correctos en la comunicación para lograr que ésta sea correcta.

También hemos recreado un servicio de streaming y un cliente que lo usa en la red, por lo que podemos estudiar cuando el usuario consigue más del 95% de paquetes correctos para una comunicación correcta

Y por último hemos intentado simular trafico HTTP y FTP enviando paquetes a un servidor de echo que los devuelve.

Con estos tres servicios queremos comenzar a simular modificando parámetros de la cola y asignando prioridades en paquetes.

Clases creadas para la simulación comentadas a continuación:

## VoIP.cc

## HTTP.cc

## FTP.cc

## Escenario.cc

## Simulacion.cc

## Observador.cc

# Simulaciones

Explicar proceso de simulación.cc

# Validez de los Resultados

Considerando los resultados obtenidos de las simulaciones, tenemos que validarlos según el criterio de simulación y verificar que tienen sentido en la simulación realizada y que hemos llegado a los valores esperados antes de realizarlas.

# Conclusiones

Comprobamos que aplicando una cola con prioridades en un escenario simple con cuello de botella logramos tener una red funcional para los servicios con mayor requisito de ancho de banda como puede ser VoIP y Streaming, dejando los demás servicios en segundo plano, pero no totalmente bloqueados por los prioritarios ya que si no se convertiría en un sistema en el que los servicios prioritarios toman el control del canal.