



SIMULADOR DE PRESTACIONES DE UNA RED CORPORATIVA

Sánchez Peris, Santiago

Tutor: Romero Martínez, José Oscar

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2015-16

Valencia, 5 de julio de 2016

Resumen

En este Trabajo de Fin de Grado se tratará de desarrollar una herramienta integral que cubrirá la necesidad de estimar uno de los parámetros críticos en cualquier red corporativa: el tiempo. Esta herramienta estará compuesta por un simulador donde podremos detallar la topología de la red que queremos analizar y de una metodología de medida para la caracterización de los diferentes dispositivos que queramos introducir en el simulador.

El simulador consistirá en un software que se desarrollará para ofrecer la funcionalidad de poder desplegar cualquier tipo de topología de red, teniendo capacidad de elegir tipos de dispositivos, tipos de enlace, estado de utilización de los enlaces, velocidad de los enlaces, tamaño de paquete transmitido por la red... Este software también nos ofrece la posibilidad de cargar las medidas de ciertos dispositivos mediante una base de datos o simplemente introducirlas manualmente después de caracterizar nuestros dispositivos. El desarrollo de este software se explicará a grandes rasgos pues no es el objetivo del gestor de redes saber cómo funciona el software, sino como utilizarlo.

Centraremos nuestros esfuerzos en ofrecer una explicación técnica de la metodología de medida que describiremos posteriormente y de cómo integrar estas medidas generadas en el simulador, para conseguir una herramienta realmente funcional en el despliegue de redes corporativas.

Resum

En aquest Treball de Fi de Grau es tractarà de desenvolupar una ferramenta integral que cobrirà la necessitat de estimar un dels paràmetres crítics en qualsevol xarxa corporativa: el temps. Esta ferramenta estarà composta per un simulador on podem detallar la topologia de la xarxa que volem avaluar y de una metodologia de mesura per a la caracterització de els diferents dispositius que vulguem introduir al simulador.

El simulador consistirà en un software que es desenvoluparà per a oferir la funcionalitat de poder desplegar qualsevol tipus de topologia en la xarxa, tenint la capacitat per elegir els tipus de dispositius, tipus de enllaços, estat de utilització de els enllaços, velocitat dels enllaços, mida de paquet transmès per la xarxa... Aquest software també ens ofereix la possibilitat de carregar les mesures de certs dispositius mitjançant una base de dades o simplement introduir-les manualment després de caracteritzar els nostres dispositius. El desenvolupament de este software s'explicarà amb poques paraules doncs no es el objectiu del gestor de xarxes saber com funciona el software sinó com gastar-lo.

Centrarem els nostres esforços en oferir una explicació tècnica de la metodologia de mesura que descriurem posteriorment i com integrar estes mesures generades en el simulador, per a aconseguir una ferramenta realment funcional en el desplegament de les xarxes corporatives.

Abstract

The purpose of this paper is to present an integral tool that serves the need to estimate one of the critical parameters of any corporate network: latency. This tool consists in a simulator where you can design your own network topology together with a measurement methodology that characterizes the different types of devices that are added in the simulator.

The simulator is primarily a software package that allows the possibility to choose any network topology, the different devices, the types, utilization and the speed of the connections, the packets size transmitted through the network... This software also allows us to integrate different device measures by a data base or simply by typing it in the software. The software core architecture is

only briefly mentioned because the main focus is on how to use the tool rather than explain how the software is built.

The main objective is to provide a technical insight of the measurement methodology as per a later description and how we can integrate these measures into the simulator. In this way we will achieve a fully functional tool in the corporate network deployment.

Índice:

Capítulo 1.	Introducción	3
Capítulo 2.	Objetivos	4
Capítulo 3.	Planteamiento base teórica	5
3.1	Emulador vs Simulador	5
3.2	Definición funcionamiento y parámetros del simulador	5
3.2.1	Descripción general	5
3.2.2	Tamaño Trama	6
3.2.3	Topología	6
3.2.4	Modelo matemático	6
3.3	Definición de los dispositivos y sus parámetros	7
3.3.1	Velocidad	7
3.3.2	Utilización	7
3.4	Estudio pila protocolos	8
3.4.1	Protocolos Utilizados	8
3.4.2	Cabeceras protocolos	9
Capítulo 4.	Metodología medición	10
4.1	Visión general de la metodología	10
4.1.1	Problema Sincronización	10
4.1.2	Metodología Alternativa	10
4.2	Descripción del software empleado para las medidas	11
4.2.1	IP Traffic – Test and Measure	11
4.2.2	Wireshark	15
4.2.3	Coefficiente de “Ajuste”	15
4.3	Topologías de medición	16
4.3.1	Enlace Ethernet	16
4.3.2	Enlace Ethernet con utilización	17
4.3.3	Enlace Serial	17
Capítulo 5.	Medidas	19
5.1	Equipos utilizados	19
5.1.1	PCs (Host)	19
5.1.2	Router CISCO 1841	19
5.1.3	Router CISCO 1921	20
5.1.4	Switch CISCO Catalyst 2950 series	21
5.1.5	Switch CISCO Catalyst 2960 series	21
5.2	Planteamiento medidas	22

5.3	Medidas Router CISCO 1841	22
5.3.1	Interfaces Fast Ethernet.....	22
5.3.2	Interfaces Fast Ethernet con utilización	22
5.3.3	Interfaces Serial.....	23
5.4	Medidas Router CISCO 1921	23
5.4.1	Interfaces Ethernet.....	24
5.4.2	Extrapolación velocidades altas: 10Gbps.....	24
5.5	Medidas Switch CISCO Catalyst 2950 series	25
5.6	Medidas Switch CISCO Catalyst 2960 series	25
Capítulo 6.	Integración de las medidas en el Software	26
Capítulo 7.	Evaluación global de la herramienta	28
Capítulo 8.	Conclusiones y futuras líneas de trabajo	33
Capítulo 9.	Bibliografía.....	34
ANEXO A:	Desarrollo del Software.....	35
ANEXO B:	Guía Simulador.....	37

Capítulo 1. Introducción

A lo largo de esta memoria vamos a detallar y explicar los principios técnicos de la herramienta desarrollada en este Trabajo de Fin de Grado.

Empezaremos por recoger y explicar los diferentes objetivos que pretende abarcar esta memoria junto al software desarrollado, especificando a que público se puede dirigir esta solución y en los ámbitos y situaciones en la que es aplicable.

En la primera parte expondremos y analizaremos la base teórica sobre la cual se fundamentará toda la metodología desarrollada posteriormente para lograr caracterizar temporalmente cierto dispositivo de nuestra red.

Sobre esta base teórica podremos desarrollar la metodología de medida en cualquier dispositivo, enumerando las distintas herramientas que necesitaremos para poder ejecutarla, describiendo su funcionamiento y las diferentes características que emplearemos.

Con la metodología descrita y analizada podremos realizar las primeras mediciones sobre dispositivos reales y comentar e interpretar los distintos resultados que arrojen estos datos, para después explicar las diferentes opciones de integración de estas medidas en el software y caracterizar así los diferentes dispositivos medidos.

Por último una vez alcanzado nuestro objetivo, poder ofrecer los conocimientos y herramientas para caracterizar un dispositivo e incluirlo en el simulador, realizaremos una evaluación de la precisión del mismo comparando los resultados del simulador con los resultados prácticos medidos con la misma topología, consiguiendo así una referencia válida para posteriormente enumerar y explicar las conclusiones del conjunto del Trabajo de fin de Grado.

Capítulo 2. Objetivos

El objetivo de este proyecto es aportar los conocimientos y herramientas necesarias a cualquier individuo que se proponga realizar un estudio de la latencia, es decir retardo, en cualquier red corporativa.

En primer lugar se transmitirán los conocimientos fundamentales críticos para poder aprender a caracterizar un dispositivo de la red, mediante la realización de una medida y su posterior integración en el software de simulación.

Aprenderemos las diferencias entre simuladores y emuladores, el funcionamiento general del simulador, entenderemos las diferencias de funcionamiento entre los distintos dispositivos de las redes, la pila de protocolos y sus cabeceras que nos interesan para el funcionamiento del simulador.

Razonaremos el procedimiento que utilizaremos para evaluar las prestaciones de cierto dispositivo bajo determinadas condiciones pudiendo caracterizarlo en situaciones que tengan un grado de similitud alto respecto al uso común del dispositivo.

Conoceremos como utilizar adecuadamente el simulador evitando fallos o bugs y facilitando así una experiencia de usuario óptima con unos resultados satisfactorios.

Por último, seremos capaces de conocer el grado de precisión de este simulador gracias a un caso de red real, donde buscaremos analizar y razonar su fiabilidad, pudiendo así conocer la exactitud del resultado que nos aporta el simulador.

Capítulo 3. Planteamiento base teórica

3.1 Emulador vs Simulador

Antes de empezar a definir con mayor precisión el proyecto necesitamos conocer exactamente cuál es la función de un simulador y sus diferencias respecto a un emulador.

Un emulador es un software que permite ejecutar programas en una plataforma (sea una arquitectura de hardware o software, un sistema operativo) diferente de aquella para la cual fueron escritos originalmente.

Si observamos la definición de simulador vemos: Dispositivo o aparato que reproduce finge o imita un fenómeno, el funcionamiento real de otro aparato o dispositivo o las condiciones de entorno a las que están sometidos una máquina, aparato o material.

La diferencia entre emulador y simulador queda en relieve tras leer detenidamente dos posibles definiciones, por ejemplo un emulador trata de ejecutar programas de una intentando ser lo más fidedigno posible a como lo hacía en su plataforma original, un simulador tan solo reproduce el comportamiento de dichos programas.

Por tanto en nuestro caso un simulador de prestaciones será un programa que se encargará de reproducir el comportamiento de otro dispositivo, en nuestro caso una red, centrándose en las obtención de las prestaciones, es decir en el retardo interno que tendrá esta red.

Ya existen en el mercado muchos simuladores de redes, Cisco Packet Tracer, Omnett++, GNS3, NetSim..., capaces de imitar el comportamiento de las redes en prácticamente cualquier aspecto, enrutamiento, control listas de acceso, firewalls este proyecto se va a diferenciar de estos simuladores gracias a que va a estudiar el retardo en las redes corporativas y va a ofrecer complementariamente una metodología para poder caracterizar cualquier dispositivo y emplearlo en el simulador.

3.2 Definición funcionamiento y parámetros del simulador

3.2.1 Descripción general

Conocemos que nuestro programa debe reproducir el comportamiento de una red para poder evaluar correctamente su retardo interno, pero falta definir que dispositivos vamos a simular con que características y bajo que condiciones. Intentaremos que las características y las condiciones se acerquen a la realidad de cualquier red corporativa.

Actualmente en las redes encontramos principalmente dos tipos de dispositivos relevantes, los Routers que son dispositivos de capa 3 capaces de encaminar paquetes entre varias redes y los Switch dispositivos de capa 2 que se encargan de gestionar las tramas entre los distintos segmentos de la red. Por tanto deberemos poder incluirlos a ambos en nuestro simulador.

El modo en el que se relacionan estos dispositivos en una red es mediante enlaces. Dependiendo de los interfaces de cada dispositivo tendremos diferentes tipos de enlaces, los que se encuentran más extendidos son los enlaces Ethernet, bien en su versión básica de 10Mbps, en su versión Fast Ethernet de 100Mbps o en la versión de Gigabit Ethernet de 1Gbps. Existen también un tipo de enlace entre Routers llamado enlace Serial que también es interesante incluir pues los Routers comerciales actuales suelen llevar módulos preparados para soportar estos tipos de enlaces también configurables en varias velocidades reducidas respecto a la tecnología Ethernet.

Estos enlaces se encuentran bajo ciertas circunstancias como puede ser la utilización del mismo, el comportamiento del mismo variará en función del porcentaje de utilización que tengamos. Sería interesante poder analizar estos enlaces con un porcentaje de utilización nulo (0%), medio (50%) y alto (80%).

3.2.2 Tamaño Trama

Otro factor a definir es el tamaño de trama que atravesará la red, pues es razonable pensar que este tamaño afectará directamente al retardo interno de la red. Para poder evaluar una red tiene sentido que podamos hacerlo con varios tamaños de trama representativos, es decir uno pequeño, un par muy utilizados y el MTU que en este caso será 1518 bytes contando con la cabecera Ethernet. Por tanto podemos definir tamaños de trama de 64, 576, 1024, y 1518 bytes que estudiaremos con más detenimiento cuando abarquemos la pila de protocolos que empleará la red corporativa que pretendemos simular.

3.2.3 Topología

No podemos olvidarnos de incluir en nuestro simulador un modo de indicar la topología o el camino que el simulador tiene que simular. Podríamos indicarle la dirección de red del dispositivo origen y la del dispositivo destino pero así entrarían en juego los protocolos de routing y esto se aleja del objetivo de nuestro simulador, que se aleja del comportamiento de encaminamiento en las redes de la mano de protocolos como pueden ser RIP o OSPF. Para especializar nuestro simulador en el estudio temporal, simplificando en primera instancia su comportamiento y facilitando así poder simular solo segmentos de una red una vez establecida la topología.

3.2.4 Modelo matemático

Después de definir los distintos parámetros del simulador tenemos que plantear un modo de funcionamiento, en primer lugar tenemos que pensar en que vamos a simular redes que son un conjunto de dispositivos bien Routers o Switch, que se encuentran enlazados mediante enlaces serial o Ethernet que pueden tener distintos porcentajes de utilización.

Con esta visión podemos modelar el retardo interno de la red como la suma de los retardos de los diferentes dispositivos que la constituyen que se encuentran bajo ciertas condiciones. Así también tiene sentido que estos retardos se computen al salir de dispositivo pues mientras esta en el dispositivo es cuando se consume el tiempo.

Conceptualmente la red será una cadena de bloques que tendrán como característica el tipo de dispositivo, el tipo de enlace, la utilización del mismo, una cierta velocidad y el retardo que añaden a la red bajo ciertas condiciones. Por tanto la suma de todos estos retardos de los diferentes bloques de la red constituirá el retardo interno de la red.

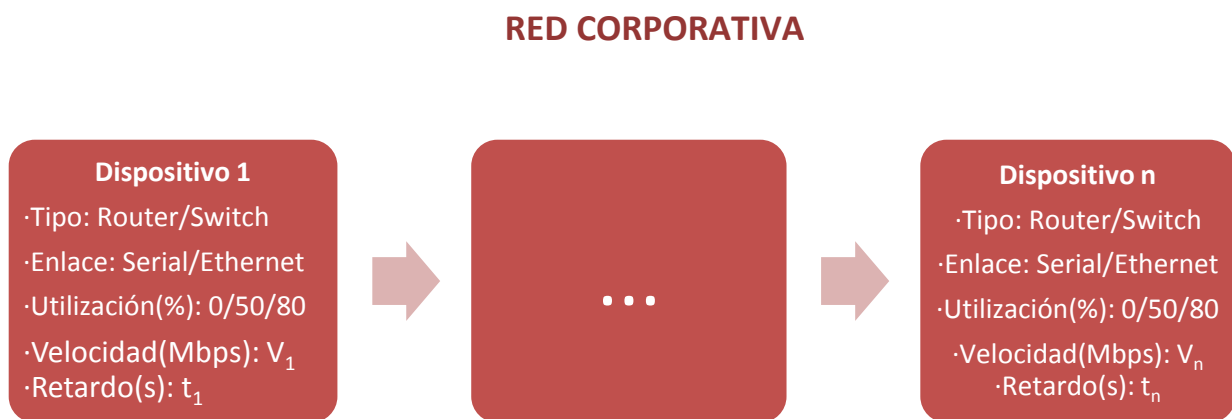


Figura 1. Modelo “Cadena” de Red Corporativa.

$$T_{tot} = \sum_{i=1}^n t_{disp} \quad \begin{cases} T_{tot}: \text{Retardo total en la red} \\ t_{disp}: \text{Retardo del dispositivo} \end{cases} \quad (3.1)$$

Como apreciamos en la formula anterior (3.2) se define el modelo matemático del simulador que será la suma de los retardos de los diferentes dispositivos en unas condiciones determinadas, estos retardos se habrán obtenido gracias a la metodología de medida que expondremos en el siguiente capítulo y se almacenaran sobre una base de datos a la cual se consultará con los distintos parámetros para poder obtener los tiempos determinados para cada dispositivo en particular y posteriormente poder calcular el retardo total de la red.

3.3 Definición de los dispositivos y sus parámetros

Como se ha comentado anteriormente en nuestro simulador permitiremos dos tipos de dispositivos: Routers y Switch, cada elemento tendrá una funcionalidad diferente en la red, pese a esto ambos dispositivos comparten las mismas características básicas, que vamos a explicar a continuación, estas características vamos a especificarlas detalladamente y tendremos que ser coherentes con ellas a lo largo del desarrollo de nuestra herramienta integral de simulación.

3.3.1 Velocidad

Como ya hemos nombrado anteriormente la velocidad será una característica de nuestros dispositivos pero. ¿A que nos referimos con velocidad?

Cuando nos referimos a la velocidad de un enlace hablamos de la velocidad máxima de transmisión de los datos en el mismo, es decir entre las dos interfaces del enlace. Por ejemplo, podemos conectar un enlace entre una interfaz Gigabit Ethernet y una Fast Ethernet, sus velocidades son respectivamente 1Gbps y 100Mbps, este enlace transmitirá a la máxima velocidad que soporten ambos dispositivos en este caso serán 100Mbps.

Cuando indiquemos la velocidad de un enlace estaremos indicando la velocidad establecida entre ambas interfaces y supondremos siempre que intentamos transmitir a la máxima velocidad siempre.

3.3.2 Utilización

La utilización la definiremos como el volumen de tráfico que ya estaba transmitiéndose por una interfaz del dispositivo respecto al volumen de tráfico, o velocidad, que vamos a intentar transmitir nosotros, este parámetro estará expresado en tanto por ciento (%). Vamos a exponer una situación para entender el concepto. Si por una interfaz de un Router se transmite un flujo ajeno de datos de 50Mbps y nosotros vamos a intentar transmitir 100Mbps la utilización será el cociente entre las velocidades de estos flujos del siguiente modo:

$$\text{Utilización (\%)} = \frac{\text{Velocidad flujo ajeno}}{\text{Velocidad flujo propio}} * 100 \quad (3.3)$$

En nuestro ejemplo tendríamos:

$$\text{Utilización (\%)} = \frac{50 \text{ Mbps}}{100 \text{ Mbps}} * 100 = 50 \% \quad (3.2)$$

Cuando indiquemos que existe cierta utilización en un enlace, estaremos afirmando que en ambos Routers que forman este enlace existe una utilización de cierto tanto por ciento.

Como podemos observar a nivel conceptual los parámetros los asociaremos a los dispositivos y a sus interfaces, pero a nivel intuitivo es decir de usuario estos parámetros se identificarán con el enlace.

3.4 Estudio pila protocolos

3.4.1 Protocolos Utilizados

Conocer el tamaño de las tramas enviadas y recibidas es necesario para el desarrollo del simulador, por tanto es esencial conocer los protocolos implicados en la transmisión de los mensajes a lo largo de la red, pasando por sus distintos niveles. Como referencia vamos a emplear el modelo OSI y sus siete capas descritas brevemente en la siguiente figura.

Las capas que nos van a importar a nivel de generación del tamaño de la trama enviado por la red serán la capa de transporte, en nuestro caso UDP pues no tiene sentido plantearnos un protocolo orientado a la conexión como podría ser TCP, la capa IP encargada de facilitar a los Routers el encaminamiento a través de la red y por último la capa de enlace de datos cuya tecnología también suele definir la capa física, en nuestro caso al tener dos tipos de enlace tendremos 2 tecnologías posibles: Ethernet o Serial.



Figura 2. Pila Protocolos Modelo OSI vs Capas Utilizadas.

Como bien sabemos, cada capa vista anteriormente generará una cabecera en el paquete, aumentando así su tamaño y el ancho de banda que generaría un hipotético flujo de datos compuesto por estos paquetes.

3.4.2 Cabeceras protocolos

Tenemos que tener claro que tendremos 2 tamaños diferentes de paquetes dependiendo de si nuestro enlace será serial o será Ethernet.

Primero analizaremos las cabeceras cuando utilizamos un enlace Ethernet:



Figura 3. Cabeceras paquete Ethernet.

Para el caso donde emplearemos el enlace serial, la encapsulación que emplearemos será PPP, por tanto nos quedara el siguiente conjunto de cabeceras:



Figura 4. Cabeceras paquete Serial.

Capítulo 4. Metodología medición

4.1 Visión general de la metodología

Para empezar a explicar la metodología que vamos a emplear primero tenemos que definir exactamente cuál es la medida que queremos para posteriormente poder razonar un protocolo para poder realizar la medida y entender las limitaciones de los equipos utilizados, para considerar si esta forma de proceder se adecua a nuestras necesidades.

La medida que queremos obtener es el retardo que inserta un dispositivo, tras recibir un paquete por una interfaz, gestionarlo en su interior y reenviarlo por otra interfaz. Para esto deberemos emplear varios ordenadores, uno que genere el paquete y el otro que lo reciba, para poder obtener las distintas magnitudes y obtener la que queremos, el retardo del dispositivo.

4.1.1 Problema Sincronización

Para realizar una medida de cuánto tarda un paquete en llegar desde un ordenador hasta otro atravesando un Router o Switch la lógica nos incita a que creamos una marca temporal en el ordenador origen y cuando llegue a destino, poder obtener el tiempo que ha tardado mediante:

$$t_{\text{retardo}} = t_{\text{destino}} - t_{\text{origen}} \quad (4.1)$$

Desafortunadamente nos asalta el primer problema, la referencia temporal debe ser la misma por tanto necesitamos sincronización entre los dos relojes de los ordenadores para que este método pueda aportarnos alguna solución válida.

Si analizamos el orden de magnitud de las medidas que se encontraran entre los ms y μs vemos que necesitaremos una precisión notable pues estamos midiendo ordenes de $10^{-3} - 10^{-6}$ también sería recomendable que fuera estable, es decir que tuviese una deriva de la frecuencia de funcionamiento reducida.

En cualquier ordenador de uso diario está disponible un reloj, es decir un oscilador, que suele ser de cuarzo y marca la hora del sistema empleándose en las distintas tareas del sistema operativo. Por lo general en los sistemas operativos Windows, no se suele disponer de tal precisión en el reloj, por tanto aunque solucionaríamos el problema de sincronización nos encontraríamos con el problema de la precisión.

El problema de la sincronización y de la exactitud de los relojes en los sistemas operativos penaliza la metodología lógica, que cabría esperar que funcionara. Pero esta es difícilmente aplicable en nuestro caso. Como el objetivo es diseñar una metodología intentando que sea más accesible y no se necesiten equipos especializados ni sistemas operativos que gestionen más exactamente el tiempo, como podrían ser ciertas distribuciones Linux. Por esto emplearemos un método de medida más creativo intentando salvar estos problemas sin restringir la metodología de medida al gran número de usuarios de plataformas Windows que carece de conocimientos para utilizar distribuciones Linux.

4.1.2 Metodología Alternativa

Para evitar los problemas detallados en el apartado anterior vamos a intentar independizar la medida a realizar del tiempo al máximo posible, así podremos esquivar los problemas relacionados con el mismo.

La única forma que tenemos de obtener el tiempo indirectamente es a partir de la velocidad, medida que si podemos medir con exactitud en los ordenadores. Tenemos que desarrollar el método para poder calcular el tiempo que emplea el dispositivo en procesar un paquete.

Sabemos que:

$$\text{Velocidad [bits/segundos]} = \frac{\text{Tamaño Paquete [bits]}}{\text{Tiempo procesado[segundos]}} \quad (4.2)$$

Por tanto si conocemos el tamaño de paquete que enviamos y podemos medir la velocidad podemos despejar del siguiente modo el tiempo de procesado que es la medida que queremos.

$$\text{Tiempo procesado}[\text{segundos}] = \frac{\text{Tamaño Paquete}[\text{bytes}] * 8[\text{bits}]}{\text{Velocidad} [\text{bits/segundos}]} \quad (4.3)$$

Hemos diseñado un procedimiento para obtener el principal parámetro que caracteriza nuestros dispositivos en el simulador, ahora necesitamos ciertas herramientas para poder aplicar esta metodología, estas herramientas son el IP Traffic – Test and Measure y el Wireshark, que explicaremos a continuación.

4.2 Descripción del software empleado para las medidas

En este apartado vamos a exponer los programas empleados en la metodología de medición explicando para cual va ser su función e indicando las principales características que van a ser de nuestro interés de cara a la realización de las medidas.

4.2.1 IP Traffic – Test and Measure

En primer lugar tenemos el programa IP Traffic – Test and Measure [1] es desarrollado por la empresa ZTI Communications. En las pruebas se ha usado la versión 2.8 December 2015 - (64 bits).

Este software será el empleado para medir las velocidades en los diferentes ordenadores en los que aplicaremos la metodología alternativa de medición que hemos propuesto en el apartado anterior.

Este programa se encargará de generar el tráfico que le indiquemos mediante las opciones entre los dos ordenadores. Puede medir la velocidad, los paquetes por segundo, la cantidad de bytes recibidos, la tasa de pérdidas...

Vamos a explicar cómo utilizarlo, el programa se ejecutará entre los dos ordenadores (emisor - receptor) que van a estar conectados mediante el dispositivo a analizar, empleando las topologías que detallaremos en el siguiente apartado.

El programa en si tiene 3 ventanas que van a ser de importancia para nosotros: IP Generator – Parameters, IP Generator – Traffic + Statistics y IP Answering – Parameters + Statistics.

En primer lugar vamos a configurar el programa en la parte del ordenador emisor del tráfico.

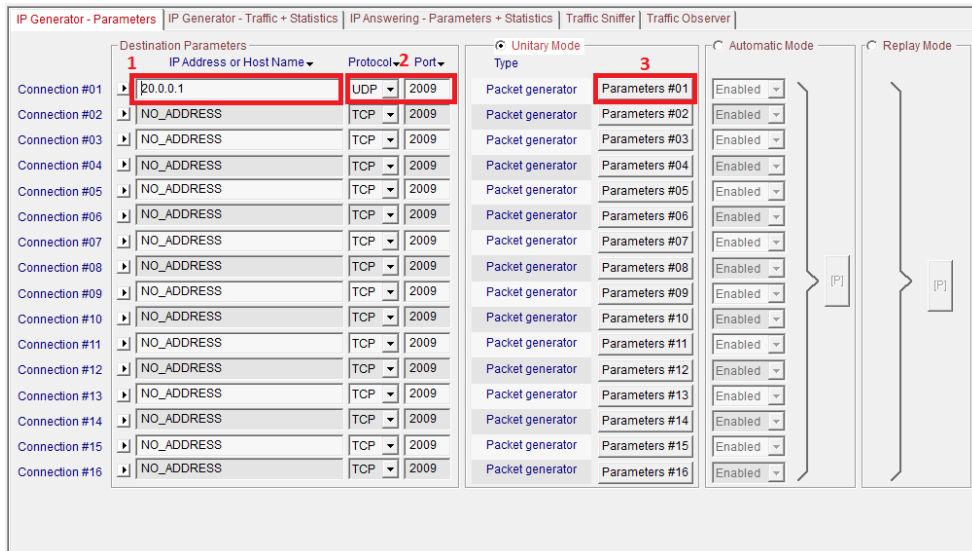


Figura 5. IP Generator - Parameters.

En la figura 5 vemos la primera ventana del programa y podemos ver 3 recuadros que corresponden a los apartados que tenemos que configurar. En primer lugar tenemos que indicar la IP del host destino de los paquetes que vamos a generar. En el segundo lugar tenemos que indicar los parámetros de la conexión ya hemos determinado en apartados anteriores realizaremos las medidas con el protocolo de transporte UDP, por tanto tenemos que elegir este protocolo y asignar un puerto libre para poder realizar la conexión. Por ultimo nos queda el botón de Parameters #01 que nos llevará a la siguiente ventana:

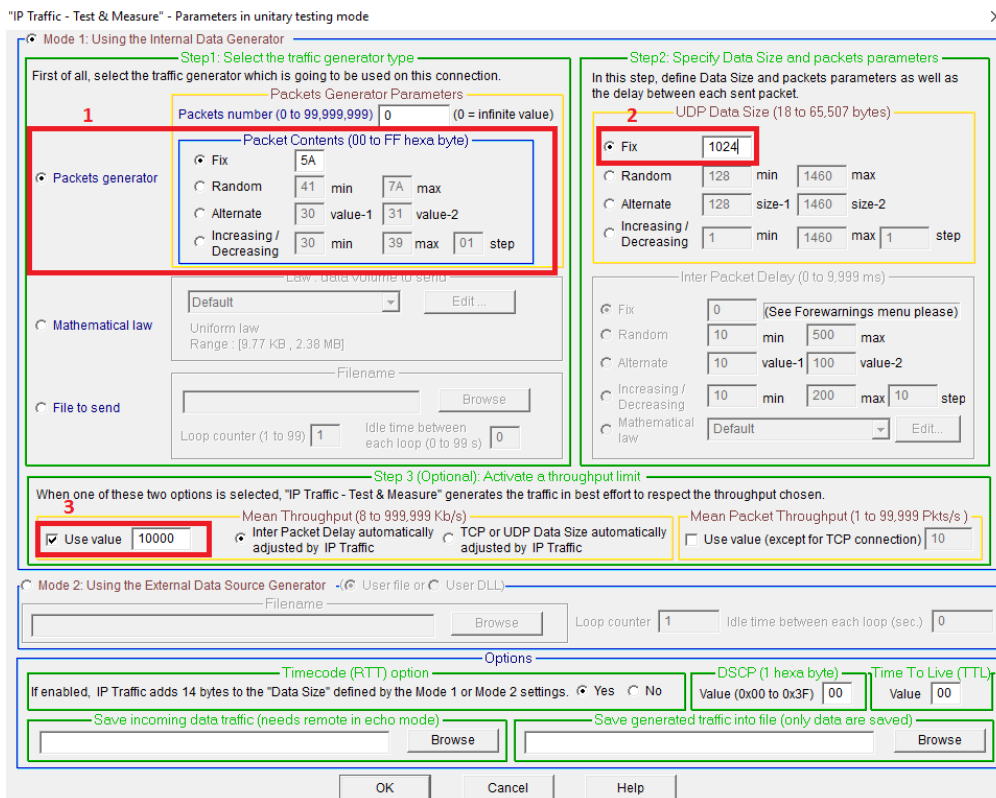


Figura 6. IP Generator – Parameters in unitary testing mode.

En primer lugar en el marcador 1, tenemos que indicar el contenido de los datos del paquete en este caso está indicado que se repita 5A durante todo el campo del datos del paquete. En el siguiente recuadro indicaremos el tamaño de los datos, en la Figura 6 vemos que se ha indicado 1024 bytes. Por ultimo en el cuadro 3 indicaremos la velocidad que generará el programa en kbps por tanto si indicamos como en la Figura 6, 10000 esto correspondería a 10 Mbps.

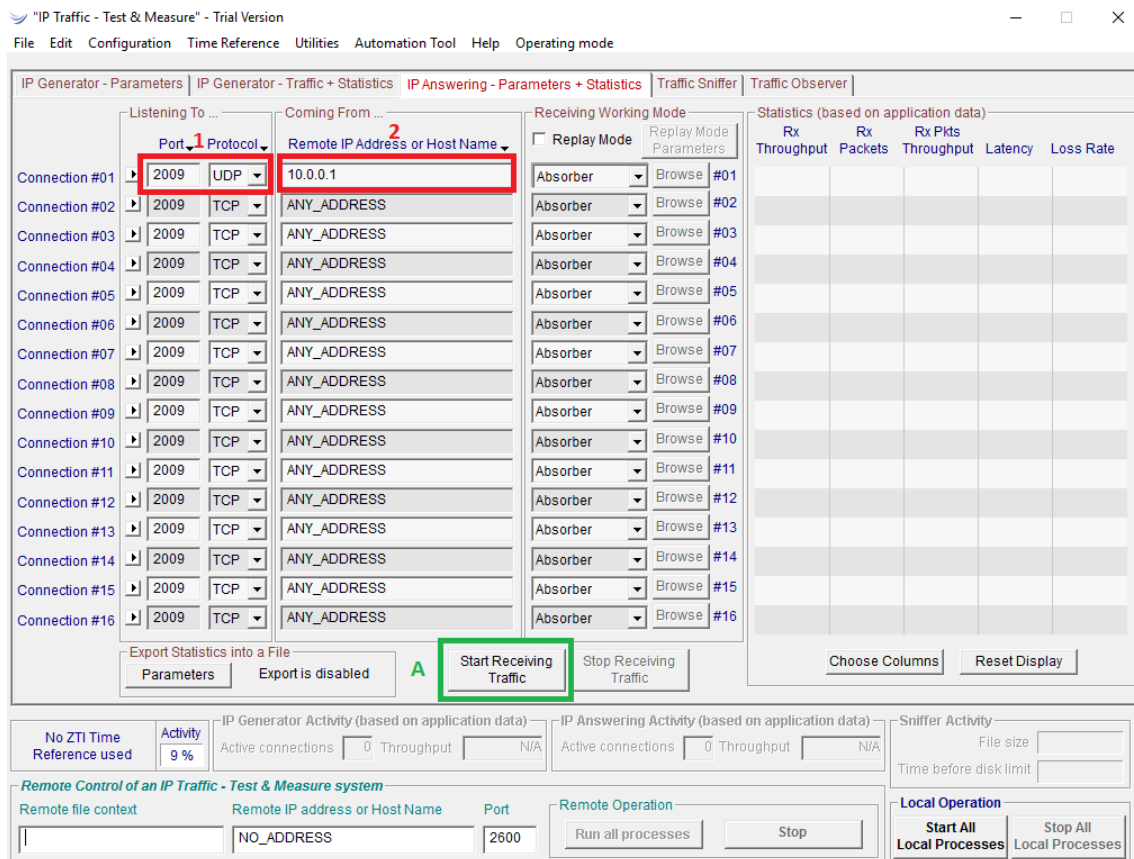


Figura 7. IP Answering – Parameters + Statistics.

Una vez configurados los parámetros del ordenador emisor deberemos pasar al receptor e indicarle desde que IP nos van enviar el tráfico, el protocolo de transporte y el puerto. Podemos verlo en la Figura 7 en los recuadros rojos 1 y 2.

Cuando ya tenemos la conexión configurada iniciamos la captura de tramas desde el recuadro A que vemos de color verde, así el receptor quedará a la espera de la transmisión de los mensajes emitidos desde el otro ordenador.

Tras configurar el programa en el receptor, volvemos al emisor a la pestaña de IP Generator – Traffic + Statistics como muestra en la Figura 8, pulsando sobre el botón del recuadro A de color verde.

Cuando pulsemos el botón veremos que en la parte de Statistics de la ventana van a empezar a actualizarse los valores de emisión, aunque los valores que nos interesarán son los del receptor del tráfico en concreto es la velocidad como podemos observar en la figura 9.

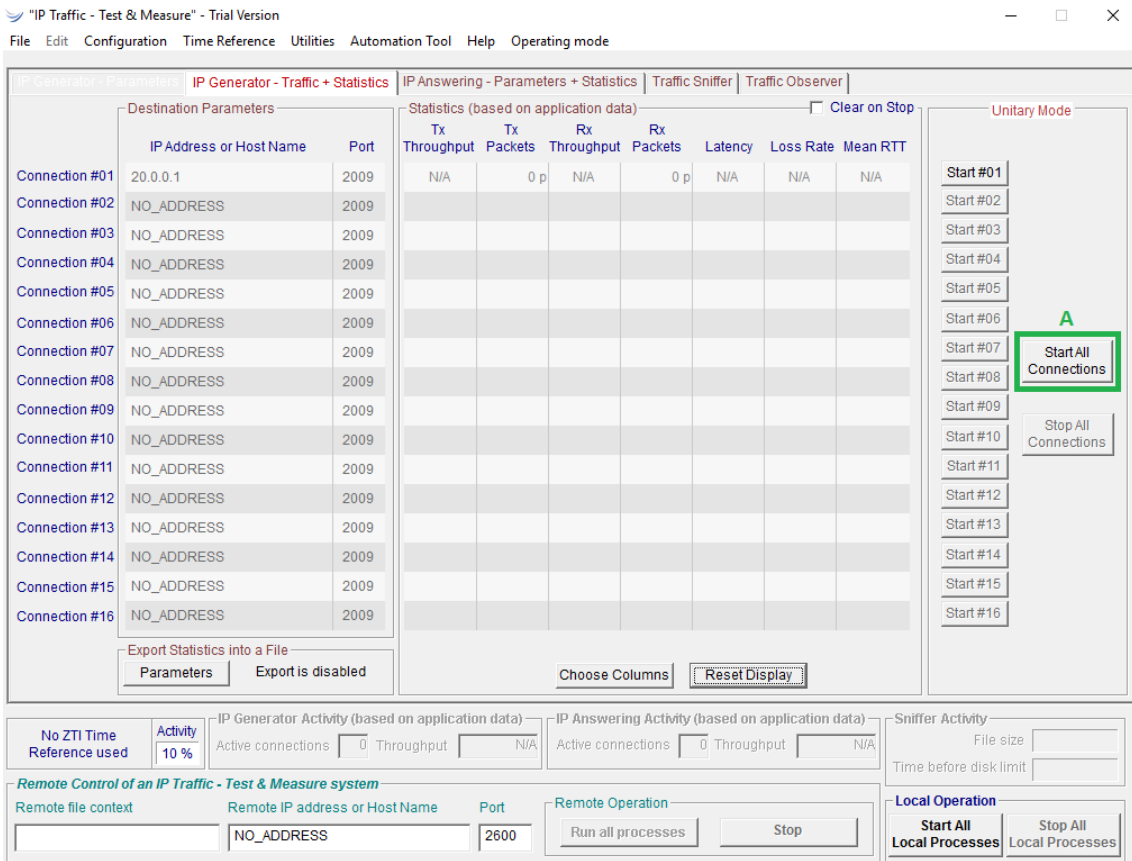


Figure 8. IP Generator – Parameters + Statistics.

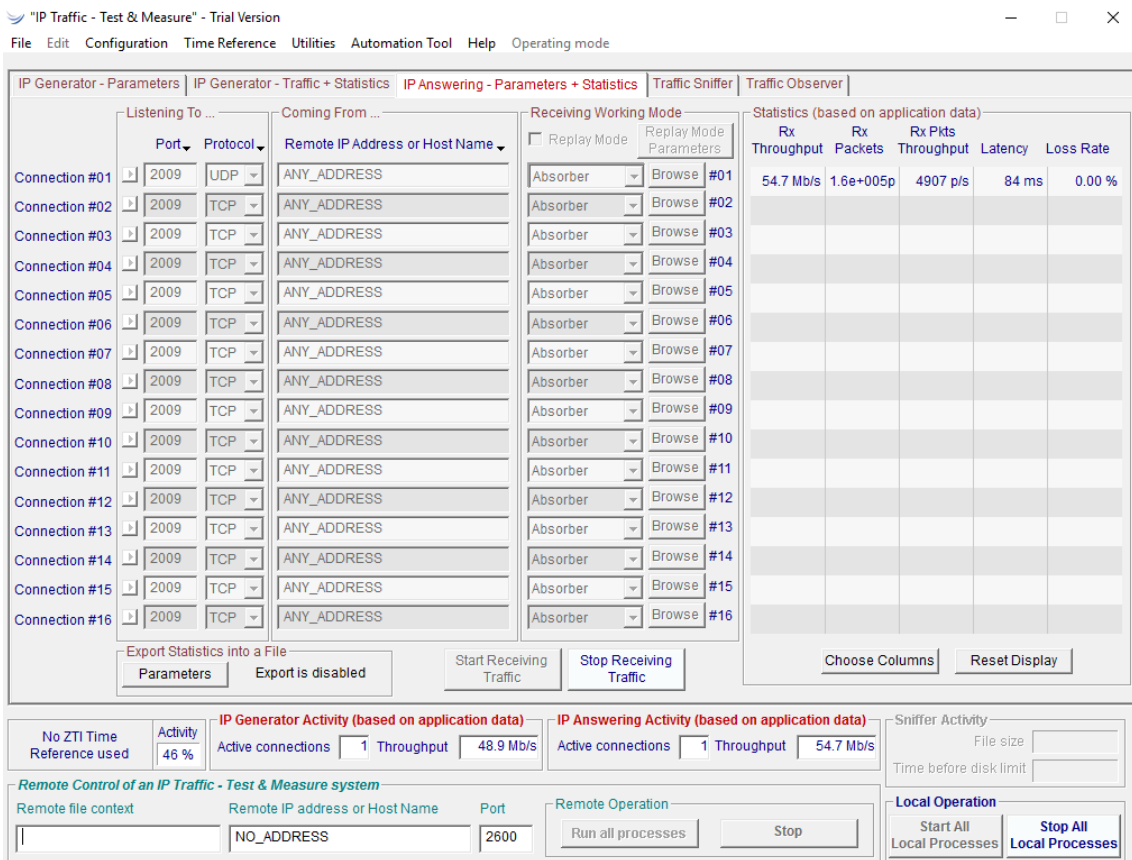


Figure 9. IP Answering – Statistics.

4.2.2 Wireshark

El Wireshark es un programa de tipo Sniffer. Su función consiste en ‘escuchar’ las comunicaciones que se producen en una interfaz y por tanto puede analizar los paquetes que se transmiten. Para nosotros su función básica va a ser monitorizar el tráfico generado por el programa anteriormente descrito y comprobar si lo que se está generando está acorde lo que se quería generar.

Tras configurar e iniciar la trasmisión de los paquetes con el programa IP Traffic, escuchamos la transmisión con el Wireshark i analizamos un paquete cualquiera.

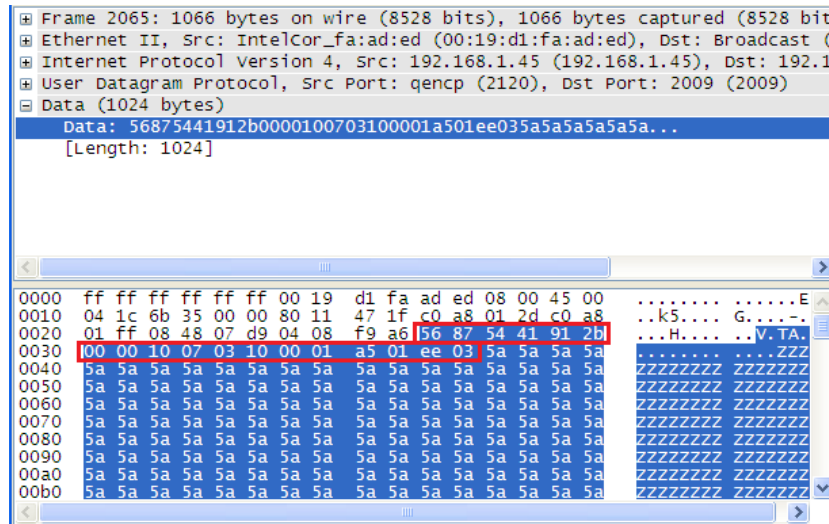


Figura 10. Wireshark: Paquete capturado con overhead.

Como podemos observar en la Figura 10, tenemos un problema con el formato del paquete. En el campo de datos vemos que el programa genera las cabeceras esperadas, pero sobre estas se generan de nuevo las cabeceras, duplicándose. Por tanto el paquete real no será de 1024 bytes como indicamos en el programa, será mayor.

A causa de este efecto tendremos un desajuste en las velocidades que indica el programa pues serán menores que en las generadas en realidad pues se creará un ancho de banda adicional asociado a la duplicación de las cabeceras, esto tenemos que caracterizarlo para poder ajustarlo y tener unas medidas que se ajusten a la realidad.

4.2.3 Coeficiente de “Ajuste”

Debemos de deducir cual es el error entre la velocidad que indica el programa y la que se genera realmente. Obviamente este error dependerá de la cantidad de bytes generados extra, es decir los de las cabeceras, respecto al tamaño del paquete. Tenemos que:

$$\text{Coeficiente Ajuste} = \frac{\text{Tamaño paquete UDP} + 46}{\text{Tamaño paquete UDP} - 46} \quad (4.3)$$

Este coeficiente siempre será mayor o igual a 1, relacionando la velocidad real, es decir la velocidad generada sobre el enlace, y la velocidad del programa que se muestra o se inserta en el programa.

$$\text{Velocidad Real} = \text{Coeficiente Ajuste} * \text{Velocidad Programa} \quad (4.4)$$

Deberemos emplear esta fórmula cuando vayamos a generar $X Mbps$ sobre el enlace y tengamos que insertar un valor menor en el programa, para que sobre el enlace solo se genere la velocidad deseada. También debemos utilizarlo cuando recogemos las medidas de las velocidades de Statistics debemos de ajustarlas pues estas serán menores a las reales.

Debemos considerar que si empleamos tamaños de paquete de 64, 576, 1024 y 1518 el porcentaje será menor contra más grande sea el tamaño del paquete, como podemos visualizar en la Tabla 1.

Tamaño Trama	64 bytes	576 bytes	1024 bytes	1518 bytes
Coefficiente Ajuste	6.111	1.1735	1.094	1.0645

Tabla 1. Valores coeficiente ajuste.

4.3 Topologías de medición

Conocidos ya los distintos programas a emplear, sus configuraciones y sus ajustes podemos pasar a proponer las distintas topologías y sus configuraciones para realizar las medidas:

4.3.1 Enlace Ethernet

En primer lugar para caracterizar un enlace Ethernet tenemos que diferenciar las configuraciones dependiendo de si vamos a medir un Router o un Switch. Para el caso del Router esta sería la topología:

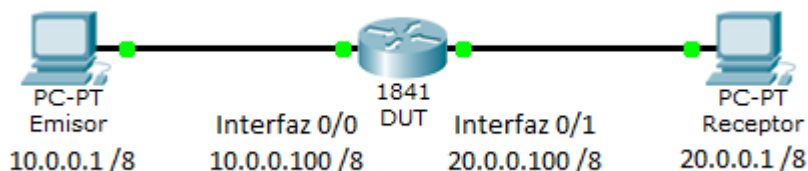


Figura 11. Configuración Topología medida enlace Ethernet en un Router.

Configuración de una interfaz del Router, por ejemplo la interfaz 0/0 a 100 Mbps:

```
Router>enable
Router# configure terminal
Router(config)#interface fastethernet 0/0
Router(config-if)#ip address 10.0.0.100 255.0.0.0
Router(config-if)#speed 100
Router(config-if)#no shutdown
```

Ahora analizaremos cual debe ser la topología y su configuración para poder medir un Switch:

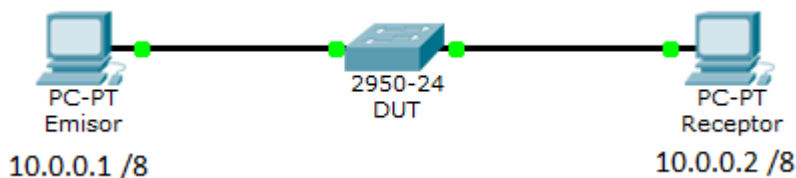


Figura 12. Configuración Topología medida enlace Ethernet en un Switch.

En este caso si no existe ninguna VLAN activa en el Switch no requeriremos de configuración, la topología será plug & play.

4.3.2 Enlace Ethernet con utilización

Para poder generar utilización en el canal deberemos tener dos flujos, el que genera la utilización y el que empleamos para la medida, y por tanto necesitaremos más dispositivos y más terminales. Cabe remarcar que no tiene sentido crear utilización sobre la interfaz de un Switch, pues estos dispositivos están muy optimizados y su funcionamiento no va a variar prácticamente en la topología que vamos a proponer, por tanto la utilización solo la consideraremos en los Routers.

Entre Emisor1 y Receptor1 tendremos una conexión del IP Traffic, la de color naranja y entre Emisor2 y Receptor2 otra distinta la de color morado, la conexión1 entre (Emisor1 – Receptor1) será la que medimos la conexión 2 será la que genere la utilización.

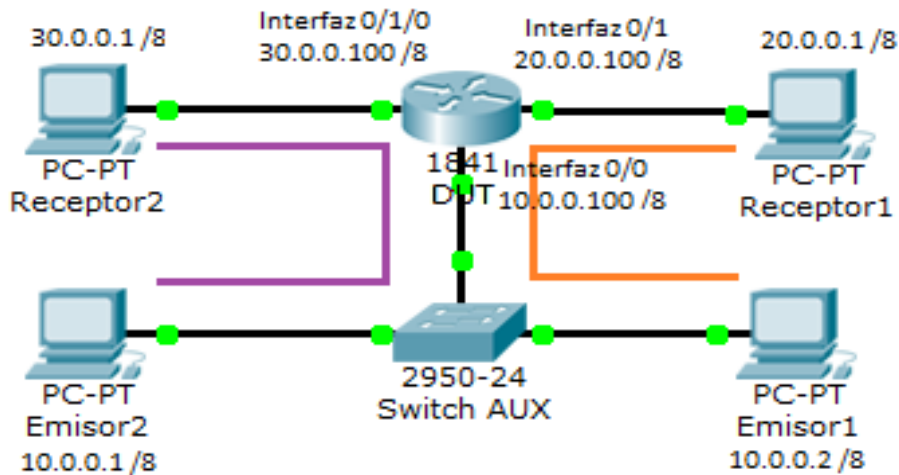


Figura 13. Configuración Topología medida enlace Ethernet en un Router con utilización.

Analizando la Figura 13, vemos que se inserta un Switch entre el Router y los PCs (Emisor-Receptor) este Switch podemos considerarlo despreciable pues al tener una optimización superior que el Router y menos funciones pues solo actúa en capa 2, no limitará los paquetes/segundo del Router y por tanto la metodología de medida propuesta anteriormente es válida.

4.3.3 Enlace Serial

La topología para medir este tipo de enlaces va a ser diferente pues los PCs no gozan de una interfaz serial tan solo tienen una interfaz Ethernet, tampoco tienen interfaces serial los Switch por tanto estas medidas tan solo corresponderán a los Routers, así que tendremos que aplicar la siguiente topología:



Figura 14. Configuración Topología medida enlace Serial en un Router.

Vemos que para medir en un enlace serial tenemos que introducir 2 Routers y emplear interfaces Ethernet entre los PCs y los Routers, esto podemos despreciarlo ya que la velocidad del enlace serial es del orden de cientos de *Kbps* y las velocidades de las interfaces Ethernet es del

orden de cientos de *Mbps* por tanto el retardo temporal que introducirán deberá ser inferior al 1% y por tanto irrelevante.

Explicada ya la topología vamos a recordar la configuración de un enlace serial:

```
Router>enable
Router#configure terminal
Router(config-if)#interface serial 0/0/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#clock rate 64000
Router(config-if)#no shutdown
```

Es interesante remarcar que el comando *clock rate* tan solo hay que añadirlo en la parte DCE del enlace serial que no es necesario añadirlo en la interfaz DTE.

Capítulo 5. Medidas

5.1 Equipos utilizados

Antes de realizar las medidas vamos a presentar los equipos que vamos a emplear. En nuestro caso van a ser: varios PCs de características idénticas, dos Routers y dos Switch todos de la marca CISCO, pues es la marca utilizada en laboratorio de redes ya que es la que mayor penetración en el mercado profesional de redes tiene.

5.1.1 PCs (Host)

Los PCs que vamos a emplear para intentar no introducir un error variable en las medidas, son todos iguales, los del laboratorio de redes. Tienen las siguientes especificaciones:

Sistema operativo	Windows 7 Professional Service Pack 1
Tipo de Sistema	Sistema operativo de 64 bits
Procesador	Intel(R) Core(TM) i3-4160 CPU @ 3.30GHz
Número de procesadores principales	2
Memoria RAM total	6.00 GBytes
Gráficos	Intel® HD Graphics 4400
Memoria de gráficos disponible	2 GBytes
Disco Duro principal	>100 Gbytes disponibles HDD
Adaptador de red	Realtek RTL8168/8111 PCI-E Gigabit Ethernet Adapter

5.1.2 Router CISCO 1841

Este va a ser el primer Router que vamos a emplear, los más numerosos en el laboratorio. Por tanto hay disponibles ranuras de expansión disponibles, tanto amplidores de Fast Ethernet como de interfaces Serial. A continuación mostraremos su aspecto, sus interfaces y sus características.



Figura 15. Router CISCO 1841

Especificaciones del Router 1841	
Forma	Desktop, 1-rack-unit
Dimensiones	(altura x anchura x profundidad) = (4.75 x 34.3 x 27.4 cm)
Peso	2.8 kg
Memoria DRAM	Synchronous dual in-line memory module (DIMM) DRAM 256 MBytes por defecto (384 MBytes máximo)
Memoria Flash	64 MBytes por defecto (128 MBytes máximo)
Ranuras modulares	dos (Slot 0: Serial x2 / Slot 1: Fast Ethernet x4)
Puertos Ethernet	dos 10/100 Mbps
Puerto USB	uno (1.1)
Puerto de Consola	uno (hasta 115.2kbps)
Puerto Auxiliar	uno (hasta 115.2kbps)
Fuente de Alimentación Interna	Sí
Alimentación	100 - 240V AC @ 50 - 60 Hz
Procesador	RM5261A-256H @ 250 MHz, Controlador Marvell GT96103A
Sistema Operativo	Cisco IOS Software, 1841 Software (C1841-ADVENTERPRISEK9-M), v12.4(25d)

5.1.3 Router CISCO 1921

Este Router es de una gama mayor que el descrito anteriormente, pues ya incluye interfaces Gigabit Ethernet y sus especificaciones son más altas, por ejemplo tiene más memoria RAM y flash. Para nosotros va a ser interesante ver como se traduce esta mejora de las especificaciones en rendimiento y poder realizar medidas con las interfaces Gigabit Ethernet.



Figura 16. Router CISCO 1921

Especificaciones del Router 1921	
Forma	Desktop, 1-rack-unit
Dimensiones	(altura x anchura x profundidad) = (4.45 x 34.29 x 29.21 cm)
Peso	3,06 kg
Memoria DRAM	Synchronous dual in-line memory module DRAM DDR2 512 MB por defecto/máxima
Memoria <i>Flash</i>	256 MB por defecto/máxima
Ranuras modulares	dos (Slot 0: - / Slot 1: -)
Puertos <i>Ethernet</i>	dos 1 Gbps
Puerto USB	uno (1.1)
Puerto de Consola	uno (hasta 115.2kbps)
Puerto Auxiliar	uno (hasta 115.2kbps)
Fuente de Alimentación Interna	Sí
Alimentación	100 - 240V AC @ 50 - 60 Hz
Procesador	RM5261A @ 350MHz
Sistema Operativo	Cisco IOS Software, 1921 Software (C1921-M), v5.4(7b)

5.1.4 Switch CISCO Catalyst 2950 series

Este Switch es tiene 24 puertos capaces de funcionar a 10BaseT y 100BaseTX, que corresponden a Ethernet, 10 Mbps, y a Fast Ethernet, 100 Mbps, respectivamente.



Figura 17. Switch CISCO Catalyst 2950 series

5.1.5 Switch CISCO Catalyst 2960 series

Este Switch es tiene 24 puertos capaces de funcionar a 10BaseT y 100BaseTX, que corresponden a Ethernet, 10 Mbps, y a Fast Ethernet, 100 Mbps, respectivamente. Y como podemos observar en la figura 13, también tiene 2 puertos Gigabit Ethernet ubicados en la parte derecha del Switch que son los que nos interesa caracterizar.



Figura 18. Switch CISCO Catalyst 2960 series

5.2 Planteamiento medidas

Conociendo ya la metodología y los distintos dispositivos del laboratorio, tan solo queda decidir que medidas vamos a realizar y sobre que equipos. Intentaremos que las medidas sean lo más representativas posibles para poder dar plena funcionalidad al software desarrollado.

Una de las principales limitaciones para realizar todas las medidas posibles es la disponibilidad de los dispositivos y sus módulos de ampliación.

En cuanto a los Switch tenemos el CISCO Catalyst 2950 series con puertos Fast Ethernet y el CISCO Catalyst 2960 series con dos puertos Gigabit Ethernet.

Tenemos dos tipos Routers, el primero el CISCO 1841 tenemos alta disponibilidad de módulos de expansión que nos van a permitir realizar medidas de Ethernet con utilización puesto tenemos el módulo de expansión de puertos de Fast Ethernet y para las medidas con congestión necesitamos 3 puertos en el Router y también podremos realizar medidas a los enlaces serial pues también tenemos módulos de expansión de enlaces serial.

En cuanto al segundo modelo de Router el 1921, es más moderno ya que ya disponemos de enlaces Gigabit Ethernet, pero desafortunadamente no tenemos muchos módulos de expansión compatibles, por tanto mediremos el rendimiento de estas interfaces a varias velocidades 10, 100, 1000 Mbps para luego poder extrapolar los retardos a 10Gbps asumiendo que el hardware estuviese preparado.

Como ya habíamos definido las medidas las tomaremos para distintos tamaños de paquetes, 64, 576, 1024 y 1518 bytes y para cuando podamos medir con utilización 0%, 50% y 80%.

Recordar que realizar la metodología de medida deberemos seguir los pasos del [Capítulo 4](#) y emplear las formulas del mismo, para obtener mediante el método propuesto los retardos de los Routers para procesar un paquete.

5.3 Medidas Router CISCO 1841

5.3.1 Interfaces Fast Ethernet

Como ya habíamos decidido realizaremos medidas con las interfaces Fast Ethernet a 100 Mbps, en primer lugar sin utilización:

Dispositivo	Router CISCO 1841			
Velocidad	100 Mbps [FA]			
Tamaño Trama	64	576	1024	1518
Medida(Mbps)	97,776	97,752	97,6942	97,0824
Retardo (µseg)	5,236	47,139	83,850	125,094

Tabla 2. Medidas interfaz Ethernet sin utilización CISCO 1841.

Con los datos del retardo ya calculados vemos que el retardo es completamente proporcional pues entre el retardo cuando el tamaño de trama es 64 bytes y cuando es 1518 tenemos una variación de menos del 1%.

5.3.2 Interfaces Fast Ethernet con utilización

Siguiendo la topología propuesta en apartados anteriores, realizamos las medidas y los cálculos pertinentes y estos son los resultados obtenidos:

Dispositivo	Router CISCO 1841											
Velocidad	100 Mbps [FA]											
Tamaño Trama	64			576			1024			1518		
Utilización (%)	0	50	80	0	50	80	0	50	80	0	50	80
Medida (Mbps)	97,77	97,77	97,77	97,75	66,77	55,85	97,69	64,65	54,26	97,08	64,82	53,86
Retardo (µseg)	5,236	5,236	5,236	47,138	69,013	82,494	83,850	126,710	150,966	125,093	187,336	225,479

Tabla 3. Medidas interfaz Ethernet con utilización CISCO 1841.

En esta tabla principalmente podemos comparar las velocidades sin utilización con las del Router CISCO 1921 y analizar el efecto de la utilización en cuanto al retardo que calculamos.

El único dato discordante lo encontramos en el tamaño de trama de 64 bytes pues pese a aumentar la utilización el retardo sigue igual. Esto se debe a que al ser pequeño el paquete, se requieren un gran número de paquetes en poco tiempo para poder ocupar el ancho de banda requerido y por los resultados medidos deducimos que la interfaz no es capaz de generar suficientes paquetes/seg para alcanzar este ancho de banda.

El resto de datos como bien podemos deducir ocurre una repartición homogénea entre los dos flujos de tráfico, en la tabla solo se muestra el efecto de esta repartición en el flujo que intenta transmitir a 100 Mbps, la máxima velocidad.

5.3.3 Interfaces Serial

Las últimas medidas que realizaremos sobre este Router serán a las interfaces serial, cuyas velocidades son 3 órdenes de magnitud inferiores a Fast Ethernet, por tanto el retardo será también un par de órdenes de magnitud superior.

Dispositivo	Router CISCO 1841							
Velocidad	64 Kbps [Serial]				128 Kbps [Serial]			
Tamaño Trama	64	576	1024	1518	64	576	1024	1518
Medida(Kbps)	58,21	55,708	54,927	52,82	125,356	117,8632	116,9352	120,76
Retardo (µseg)	8771,93	83333,33	142857,14	250000,00	4081,63	38461,54	71428,57	100000,00

Tabla 4. Medidas interfaz Serial sin utilización CISCO 1841.

Como podemos ver en los retardos estos oscilan entre 8.7 – 250 ms como habíamos deducido antes de realizar las medidas, también podemos ver que los retardos son bastante proporcionales hasta que nos acercamos a las medidas con tamaños de paquete cercanos a la MTU (*Maximum Transmission Unit*) que estamos por debajo de la recta que marcarían los puntos de las primeras medidas.

5.4 Medidas Router CISCO 1921

En este Router vamos a realizar medidas en Ethernet, Fast Ethernet y Gigabit Ethernet con velocidades de 10, 100, 1000 Mbps respectivamente, el hecho de tener varias velocidades y de querer estimar una posible medida con velocidades de 10Gbps nos fuerza a organizar una tabla para cada tamaño de paquete, en función de las velocidades. Pudiendo así observar las tendencias y analizar mejor las medias.

5.4.1 Interfaces Ethernet

Dispositivo	CISCO 1921		
Tamaño Trama (bytes)	64		
Velocidad	10 Mbps [E]	100 Mbps [FA]	1000 Mbps [GigaE]
Medida(Mbps)	9,7776	97,776	745,555
Retardo (µseg)	52,3642	5,2364	0,6867

Tabla 5. Medidas con paquete de 64 bytes Router CISCO 1921.

Dispositivo	CISCO 1921		
Tamaño Trama (bytes)	576		
Velocidad	10 Mbps [E]	100 Mbps [FA]	1000 Mbps [GigaE]
Medida(Mbps)	9,775	99,2781	820,276
Retardo (µseg)	471,4757	46,4144	5,6176

Tabla 6. Medidas con paquete de 576 bytes Router CISCO 1921.

Dispositivo	CISCO 1921		
Tamaño Trama (bytes)	1024		
Velocidad	10 Mbps [E]	100 Mbps [FA]	1000 Mbps [GigaE]
Medida(Mbps)	9,906	98,022	968,14
Retardo (µseg)	827,1298	83,5701	8,4615

Tabla 7. Medidas con paquete de 1024 bytes Router CISCO 1921.

Dispositivo	CISCO 1921		
Tamaño Trama (bytes)	1518		
Velocidad	10 Mbps [E]	100 Mbps [FA]	1000 Mbps [GigaE]
Medida(Mbps)	9,718	97,188	945,3
Retardo (µseg)	1250	124,953143	12,8467003

Tabla 8. Medidas con paquete de 1518 bytes Router CISCO 1921.

Como podemos observar en las anteriores 4 tablas vemos que las velocidades y por tanto el retardo de 10 y 100 Mbps son completamente proporcionales, en cuanto a las medidas de 1000 Mbps podemos observar que cuando los tamaños de trama son reducidos, especialmente en el caso de 64 y 576 bytes, las interfaces de los ordenadores no son capaces de generar tantos paquetes por segundo para lograr el ancho de banda máximo en el Router.

5.4.2 Extrapolación velocidades altas: 10Gbps

Pensando en el futuro, sería interesante plantear que en los próximos años las velocidades de las interfaces de los Routers tendrán que aumentar hacia velocidades como 10 y 40 Gbps, por tanto es adecuado plantear una extrapolación de estas medidas para poder tener una idea de los retardos que generarán cuando estén más implantadas en los mercados.

Dispositivo	CISCO 1921			
Velocidad	10Gbps [XGbE]			
Tamaño Trama	64	576	1024	1518
Medida(Mbps)	7008,217	7954	9583,2	9358.47
Retardo(μseg)	0,0730	0,5793	0,8548	1,2976

Tabla 9. Medidas extrapoladas 10Gbps Router CISCO 1921.

5.5 Medidas Switch CISCO Catalyst 2950 series

Habiendo medido los dos Routers que teníamos planificado, pasamos ahora a medir los dos Switch esperamos que las medidas sean bastante similares a las medidas en los Routers anteriormente:

Dispositivo	Switch CISCO Catalyst 2950 series			
Velocidad	100 Mbps [FA]			
Tamaño Trama	64	576	1024	1518
Medida(Mbps)	97,776	97,6352	97,6942	97,188
Retardo(μseg)	5,23645199	47,1965263	83,8504109	124,953143

Tabla 10. Medidas interfaz Ethernet CISCO 2950 series.

Como podemos ver los resultados son proporcionales y muy similares a los de la tabla 3 del Router 1841 cuando no tenemos utilización.

5.6 Medidas Switch CISCO Catalyst 2960 series

Realizamos las distintas medidas para el segundo de los Routers:

Dispositivo	Switch CISCO Catalyst 2960 series			
Velocidad(Mbps)	1000 Mbps [GigaE]			
Tamaño Trama	64	576	1024	1518
Medida(Mbps)	733,32	823,797	966,002	936,76
Retardo	0,69819433	5,59362327	8,48032564	12,9637792

Tabla 11. Medidas interfaz Ethernet CISCO 2960 series.

Vemos que con tamaños pequeños de trama las interfaces de los ordenadores no son capaces de generar los 1000 Mbps, por tanto vemos que con tamaños de trama de 64 y 576 bytes estamos lejos de los 1000 Mbps. En tamaños de trama grande estamos cerca del gigabit esperado en estas interfaces. Los retardos vemos que en tamaños de trama grande se acercan a ser proporcionales.

Capítulo 6. Integración de las medidas en el Software

Con los datos medidos correctamente el siguiente paso que debemos realizar es insertar todas estas medidas en el simulador, tenemos dos formas de realizar esto.

La primera es la más simple pues desde dentro del programa pulsamos sobre el botón “*Menu Modelos*” entramos en la pestaña “Insertar Medida” e introducimos una a una todas las medidas que vamos a emplear. Por ejemplo si queremos introducir la siguiente medida:

Router CISCO 1921 con un enlace Gigabit Ethernet y con un tamaño de trama de 1024 octetos tiene un retardo de 8.46 μ s, en el programa lo insertaríamos del siguiente modo:

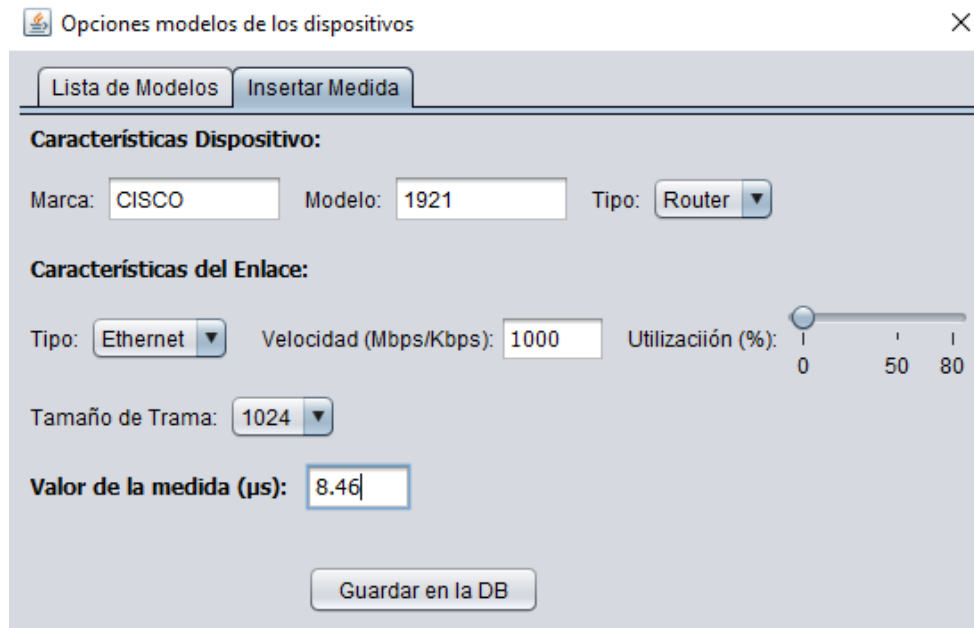


Figura 19. Insertar Medida de un Dispositivo desde el simulador.

La otra opción más elaborada pero más rápida si vamos a insertar muchas medidas es abrir el archivo “*measures.db*” que se encuentra en el directorio del *.jar* con un programa editor SQL y rellenar la tabla de esta base de datos que definimos a continuación (Si se tienen problemas con el simulador se recomienda acudir al ANEXO B: Guía Simulador).

```
-- Describe MODELOS
CREATE TABLE "modelos" (
  "id" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
  "marca" TEXT NOT NULL,
  "modelo" TEXT NOT NULL,
  "tipo" TEXT NOT NULL,
  "enlace" TEXT NOT NULL,
  "velocidad" INTEGER NOT NULL,
  "paquete" INTEGER NOT NULL,
  "congestion" INTEGER NOT NULL,
  "Time" REAL NOT NULL
)
```

Los nombres de los campos son suficientemente representativos aun así debemos remarcar que “congestion” es donde se almacena el porcentaje de utilización. Para ver un par de ejemplos insertados en la base de datos vamos a rellenarla con las medidas realizadas en el punto anterior. Abrimos la base de datos con el programa *Sqliteman 1.2.2* un editor SQL.

	id	marca	modelo	tipo	enlace	velocidad	paquete	congestion	Time
16	16	CISCO	1921	Router	Ethernet	10000	1518	0	1,29765
17	17	CISCO	1841	Router	Ethernet	100	64	0	5,2356
18	18	CISCO	1841	Router	Ethernet	100	576	0	47,1387
19	19	CISCO	1841	Router	Ethernet	100	1024	0	83,8504
20	20	CISCO	1841	Router	Ethernet	100	1518	0	125,0938
21	21	CISCO	1841	Router	Ethernet	100	64	50	5,2365
22	22	CISCO	1841	Router	Ethernet	100	576	50	69,0131
23	23	CISCO	1841	Router	Ethernet	100	1024	50	126,7106
24	24	CISCO	1841	Router	Ethernet	100	1518	50	187,3361
25	25	CISCO	1841	Router	Ethernet	100	64	80	5,2365

Figura 20. SQLiteman: Medidas en la Base de Datos.

Tenemos que ir con cuidado que el valor de tipo debe ser “Router” o “Switch” y el valor de enlace debe ser “Ethernet” o “Serial” de lo contrario el programa cuando realice las petición a la base de datos no detectará las medidas insertadas.

Capítulo 7. Evaluación global de la herramienta

Cuando ya tenemos el software totalmente funcional con las distintas medidas llega el momento de comprobar en conjunto su efectividad, es decir emplearlo en una situación práctica y ver la desviación del resultado del simulador frente a la latencia que marca en la realidad.

Vamos a intentar simular una red corporativa típica con diseño jerárquico con varias capas, como la del núcleo, la de distribución y por último la de acceso. En una red de estas características no sería común que se empleara una red por cada enlace, se emplearían subredes pero para no desviar la atención con configuraciones de lo verdaderamente importante que es la medida de la latencia desde un extremo a otro de la red.

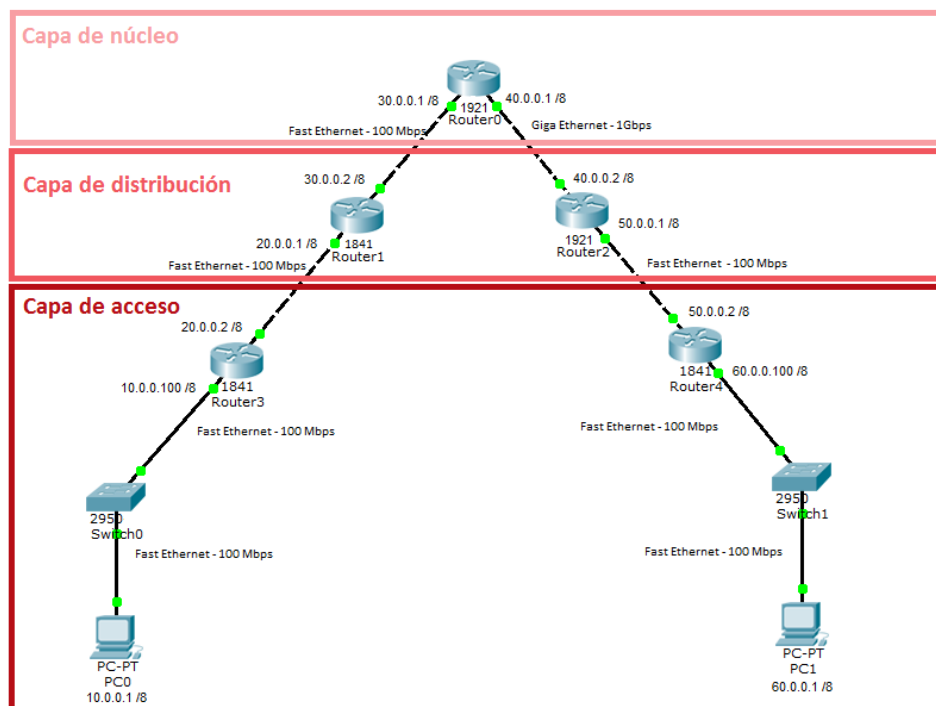


Figura 21. Topología y configuración básica red corporativa.

Podemos ver que los enlaces en las capas inferiores son de 100 Mbps y en capas superiores algunos serán de 1 Gbps pues las capas superiores soportaran mayores anchos de banda que las inferiores pues se acumulará todo el tráfico de capas inferiores en las superiores, nos limitará el número de dispositivos del laboratorio.

En el laboratorio el montaje de la red queda del siguiente modo:

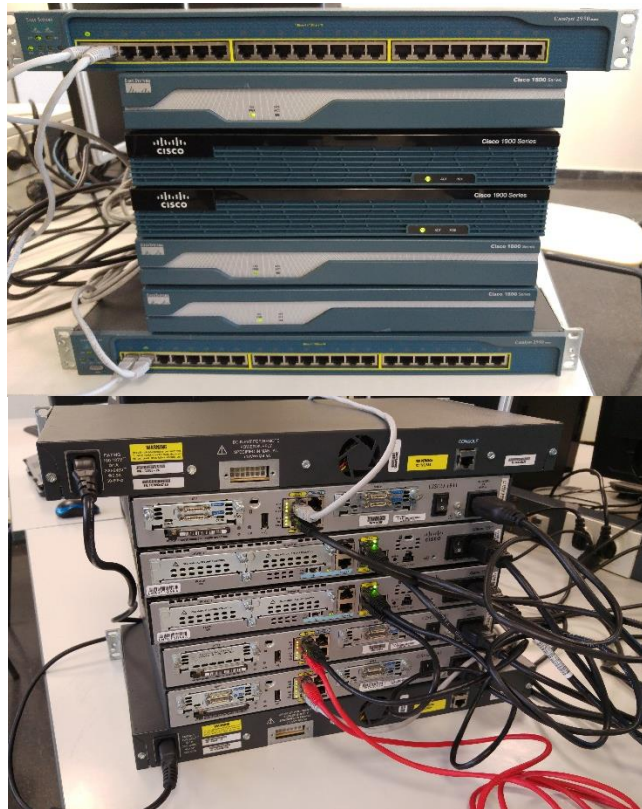


Figura 22. Fotografías montaje práctico red corporativa.

En las fotografías el Switch inferior es el correspondiente a Switch0 en el esquema de la topología, encima de este tenemos el Router0 y así sucesivamente hasta llegar al Switch1 situado arriba de la torre de Routers.

Una vez el escenario preparado tenemos que pensar un modo de realizar la medida del retardo desde el PC0 al PC1, desafortunadamente la metodología empleada anteriormente no es válida para realizar la medida, tendremos que proponer otra forma alternativa.

Como ya habíamos comentado anteriormente los sistemas operativos Windows tienen un gestor de tareas muy poco preciso en cuanto a la gestión del tiempo, por esto no podemos dar por válidas los tiempos en Windows a partir de los milisegundos, como nos podemos hacer una idea del orden de magnitud del retardo que será de cientos de microsegundos, no podemos realizar las medidas con Windows pues estas serían muy imprecisas. Estamos obligados a emplear distribuciones *on live* Linux en los ordenadores del laboratorio para poder realizar una medida con cierta exactitud. La distribución que emplearemos será Ubuntu 16.04.

Cuando tengamos ya ejecutándose Linux sobre los PCs realizaremos varios pings con el tamaño de paquete deseado para obtener el $RTT_{toda\ red}$ (Round-Trip delay Time) que es el tiempo que pasa entre que va un paquete hasta el receptor se procesa y es enviado de vuelta. El tiempo de ida y vuelta es el que nos interesa ya que la mitad de este tiempo es la latencia, que es el parámetro buscado. Por tanto tenemos que intentar eliminar el efecto de proceso del paquete en el destino de la red antes de reenviarlo. Esto podemos hacerlo conectando los dos PCs directamente y viendo cual es el $RTT_{extremos}$ para restarlo al primer RTT y obtener así la medida que queremos, para luego dividirla entre dos y así obtener finalmente el retardo que queremos podemos ver esto en la figura 7.1.

$$Retardo_{red} = \frac{RTT_{toda\ red} - RTT_{extremos}}{2} \quad (7.1)$$

Será crítico el hecho de indicar de que tamaño será el campo de datos pues nosotros queremos que el tamaño de paquete sea de un tamaño determinado. Realizamos el estudio de las cabeceras.



Figura 23. Cabeceras ping.

Tenemos 46 bytes de cabeceras por tanto el tamaño del campo de datos en el siguiente:

$$Tamaño\ campo\ datos = Tamaño\ paquete\ deseado - 28 \quad (7.2)$$

Realizamos las medidas del $RTT_{toda\ red}$ y $RTT_{extremos}$ desde los PCs corriendo Linux de manera nativa, mediante el siguiente comando:

```

ubuntu@ubuntu: ~
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ubuntu:~$ ping 10.0.0.1 -s 996
PING 10.0.0.1 (10.0.0.1) 996(1024) bytes of data.
1004 bytes from 10.0.0.1: icmp_seq=1 ttl=59 time=2.40 ms
1004 bytes from 10.0.0.1: icmp_seq=2 ttl=59 time=2.49 ms
1004 bytes from 10.0.0.1: icmp_seq=3 ttl=59 time=2.52 ms
1004 bytes from 10.0.0.1: icmp_seq=4 ttl=59 time=2.34 ms
1004 bytes from 10.0.0.1: icmp_seq=5 ttl=59 time=2.52 ms
1004 bytes from 10.0.0.1: icmp_seq=6 ttl=59 time=2.45 ms
1004 bytes from 10.0.0.1: icmp_seq=7 ttl=59 time=2.34 ms
1004 bytes from 10.0.0.1: icmp_seq=8 ttl=59 time=2.53 ms
1004 bytes from 10.0.0.1: icmp_seq=9 ttl=59 time=2.49 ms
1004 bytes from 10.0.0.1: icmp_seq=10 ttl=59 time=2.48 ms
1004 bytes from 10.0.0.1: icmp_seq=11 ttl=59 time=2.50 ms
1004 bytes from 10.0.0.1: icmp_seq=12 ttl=59 time=2.51 ms
1004 bytes from 10.0.0.1: icmp_seq=13 ttl=59 time=2.49 ms
1004 bytes from 10.0.0.1: icmp_seq=14 ttl=59 time=2.45 ms
1004 bytes from 10.0.0.1: icmp_seq=15 ttl=59 time=2.51 ms
1004 bytes from 10.0.0.1: icmp_seq=16 ttl=59 time=2.42 ms
1004 bytes from 10.0.0.1: icmp_seq=17 ttl=59 time=2.50 ms
1004 bytes from 10.0.0.1: icmp_seq=18 ttl=59 time=2.52 ms

```

Figura 24. Ping desde PC1-PC0 atravesando toda la red ($RTT_{toda\ red}$).

Conectado ahora directamente PC1 y PC0 volvemos a realizar el ping para poder obtener el $RTT_{extremos}$.

```

ubuntu@ubuntu: ~
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ubuntu:~$ ping 10.0.0.1 -s 996
PING 10.0.0.1 (10.0.0.1) 996(1024) bytes of data.
1004 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=1.20 ms
1004 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=1.23 ms
1004 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=1.21 ms
1004 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=1.19 ms
1004 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=1.21 ms
1004 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=1.23 ms
1004 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=1.16 ms
1004 bytes from 10.0.0.1: icmp_seq=8 ttl=64 time=1.24 ms
1004 bytes from 10.0.0.1: icmp_seq=9 ttl=64 time=1.22 ms
1004 bytes from 10.0.0.1: icmp_seq=10 ttl=64 time=1.23 ms
1004 bytes from 10.0.0.1: icmp_seq=11 ttl=64 time=1.23 ms
1004 bytes from 10.0.0.1: icmp_seq=12 ttl=64 time=1.17 ms
1004 bytes from 10.0.0.1: icmp_seq=13 ttl=64 time=1.23 ms
1004 bytes from 10.0.0.1: icmp_seq=14 ttl=64 time=1.23 ms
1004 bytes from 10.0.0.1: icmp_seq=15 ttl=64 time=1.16 ms
1004 bytes from 10.0.0.1: icmp_seq=16 ttl=64 time=1.21 ms
1004 bytes from 10.0.0.1: icmp_seq=17 ttl=64 time=1.16 ms

```

Figura 25. Ping desde PC1-PC0 conectados directamente ($RTT_{extremos}$).

Con las medidas realizadas podemos aplicar la fórmula 7.1 y obtener la medida del retardo de la red.

$$Retardo_{red} = \frac{RTT_{toda\ red} - RTT_{extremos}}{2} = \frac{2.34\ (ms) - 1.23\ (ms)}{2} = 555\ \mu s$$

Ahora vamos a analizar que retardo nos mostraría el simulador al insertar la misma red, con los mismos dispositivos y enlaces.

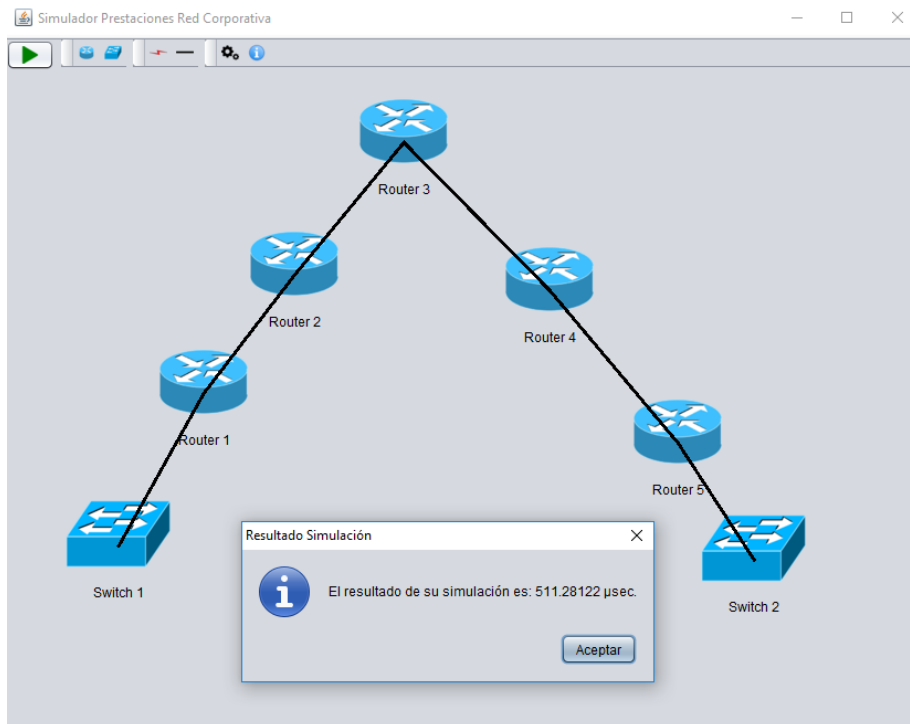


Figura 25. Resultado Simulador topología red corporativa.

Tenemos por una parte la medida realizada sobre equipos reales que son $555 \mu s$ y por otra parte tenemos el resultado del simulador que son $511.3 \mu s$, esto supone un error del 8.54% que a priori puede ser considerable, pero debemos considerar que sobre equipos reales la misma medida en varias situaciones puede variar dependiendo de factores externos, humedad, temperatura, tiempo de funcionamiento... Aun así si pensamos la situación sobre la que se genera este error vemos que en nuestra red corporativa se transitan 7 dispositivos, el error total dividido entre los dispositivos que transitamos vemos que está en torno a 1.22% un resultado fidedigno con la realidad.

Capítulo 8. Futuras líneas de trabajo y conclusiones

8.1 Futuras líneas de Trabajo

En primer lugar marcaremos las futuras líneas de trabajo para mejorar las funcionalidades del simulador y generar una herramienta aún más potente.

La primera línea de trabajo sería buscar hardware más actual para poder caracterizar dispositivos capaces de soportar velocidades de 10, 40 e incluso 100Gbps, así lograríamos ver que los retardos de los dispositivos seguirían reduciéndose hasta el orden de nanosegundos.

La siguiente línea de trabajo que podríamos llevar a cabo sería aumentar las funcionalidades del simulador y que no solo pudiera estimar la latencia de la red, sino que también pudiese calcular el tiempo de transmisión de un archivo de un tamaño determinado a lo largo de una red corporativa, es decir el ancho de banda efectivo. Sabiendo que la fórmula teórica del tiempo de transmisión si todos los dispositivos de la red y enlaces son iguales sería la fórmula 8.1, no sería complicado implementar este caso sencillo, se ha realizado la medición descrita anteriormente en la red mostrada en la figura 26, posteriormente aplicamos la fórmula 8.1 y vemos que los resultados son coherentes aunque tienen un error de en torno a un 5%. Hemos indicado en el IP Traffic que se envíen 1.000.000 *paquetes de 1024 bytes*, si sabemos que el $t_r = 83.85 \mu\text{seg}$ en el caso del Router empleado en la topología y considerando $N = 5$ saltos, tenemos:

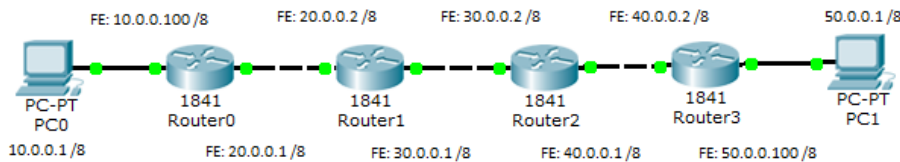


Figura 26. Topología medida tiempo transmisión de X bytes.

$$T_{tot} = N * t_r + (P - 1) * t_r \quad \left\{ \begin{array}{l} N, \text{ número de enlaces que transita el paquete} \\ P, \text{ cantidad de paquetes transmitidos} \\ t_r, \text{ tiempo de la transmisión en el dispositivo} \end{array} \right. \quad (8.1)$$

$$T_{tot} = 5 * 83.85 * 10^{-6} + (1.000.000 - 1) * 83.85 * 10^{-6} = 83.8503 \text{ seg}$$

La medida real la podemos realizar abriendo el Wireshark y restando el tiempo de llegada del último paquete en el receptor menos el tiempo de llegada del primer paquete en el receptor, esto nos da 87.6146 *seg*

Si quisiéramos implementar esta funcionalidad con distintos tipos de dispositivos en la red la fórmula se complicaría, considerar esta posibilidad sería un valor diferenciador en nuestra aplicación.

8.2 Conclusiones

Evaluando tan solo el trabajo realizado durante este proyecto vemos que hemos cumplido el objetivo que teníamos que era aportar una herramienta integral para estimar la latencia en redes corporativas, hemos explicado la caracterización de cualquier dispositivo y como integrarla en el programa. También hemos logrado evaluar su exactitud y remitiéndonos al apartado anterior consideramos que un error de en torno a un 1.25% es un error asumible y bastante contenido por esto concluimos que hemos conseguido superar notablemente los objetivos que nos marcamos al inicio de este proyecto.

Capítulo 9. Bibliografía

[1] IP Traffic – Test and Measure, web: < <http://www.zti-communications.com/iptraffic/> >

[2] Descarga Wireshark, web: < <https://www.wireshark.org/#download> >

[3] CISCO, web: < <https://www.cisco.com/c/en/us/index.html> >

[4] Descarga ISO Ubuntu 16.04, web: < <http://www.ubuntu.com/download/desktop> >

[5] Librería SQLite, web: < <http://www.sqlite.org/> >

[6] Programa SQLiteman, web: < <http://sqliteman.yarpen.cz/> >

ANEXO A: Desarrollo del Software

En ocasiones es complicado elegir una plataforma o lenguaje determinado para desarrollar un programa, en este caso no fue así pues la basta documentación acerca de java, sus miles de ejemplos de aplicaciones, sus librerías swing y AWT para desarrollo de entornos gráficos y un sistema asistido llamado palette en el NetBeans IDE 8.1 nos decantaron por elegir java como lenguaje de programación de la aplicación. Otra necesidad que teníamos era la de almacenar los datos en el programa y para esto se usó una librería externa llamada SQLite que consiste en una sencilla base de datos serverless muy empleada en aplicaciones simples, podemos ver la web oficial en la bibliografía [5].

Vamos a describir la aplicación de manera muy superficial mostrando un par de gráficos de los casos de uso de la aplicación y mostrando el diagrama de clases de esta aplicación.

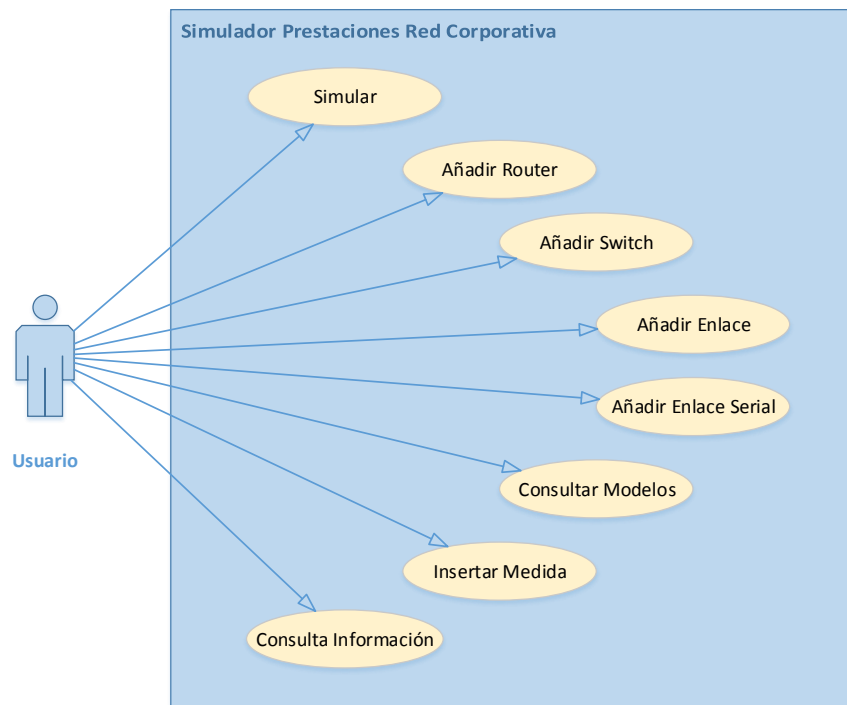


Figura A1. Casos de uso de la aplicación.

En el caso de los casos de uso de la aplicación todos son suficientemente descriptivos como para explicarlos, lo único que podríamos remarcar es que en nuestra aplicación tan solo tendríamos un rol posible, el de usuario con todas las atribuciones posibles.

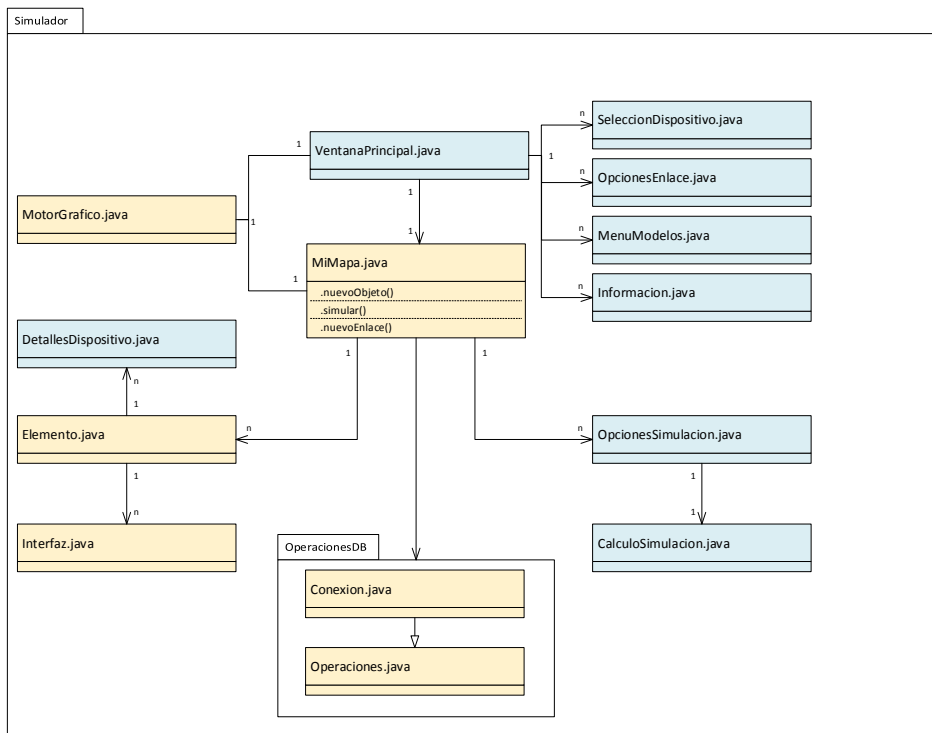


Figura A2. Diagrama de clases de la aplicación.

En este diagrama principalmente vemos dos paquetes de clases, el Simulador y OperacionesDB. El primero contiene el cuerpo principal de la aplicación el segundo se encarga de facilitar la gestión de los datos en la base de datos. Otro factor a comentar es que las clases coloreadas de color azul claro son clases que tienen interfaz gráficas las que son solo código están en amarillo.

La aplicación se sustenta sobre la clase MiMapa.java que contiene un mapa con distintos objetos de la clase Elemento y estos a su vez tiene varios objetos de la clase Interfaz en función del número de enlaces. Los objetos de la clase Elemento contienen las características de los diferentes dispositivos y de sus interfaces.

La clase MiMapa también contiene los principales métodos de la aplicación, por ejemplo añadir un Elemento, Router/Switch, .nuevoObjeto() o añadir un nuevo enlace .nuevoEnlace() y también el método que gestiona la simulación .simular().

La clase MotorGrafico.java se encarga de actualizar y representar los distintos Dispositivos y enlaces en la pantalla principal.

La clase VentanaPrincipal.java contiene todos los botones para interactuar con el programa llamando a otras ventanas, y también contiene el lugar de trabajo donde disponemos de nuestra red.

El resto de clases son representa distintas ventanas para escoger y configurar opciones y parámetros del simulador.

ANEXO B: Guía Simulador

En este Anexo se va a ofrecer una guía para el correcto funcionamiento del software, en conjunto con las medidas. En primer lugar vamos a analizar el entorno de ejecución del programa, este programa está desarrollado en java esto implica que se podrá ejecutar sobre cualquier sistemas operativo que soporte java (Windows, Linux, MAC OS...).

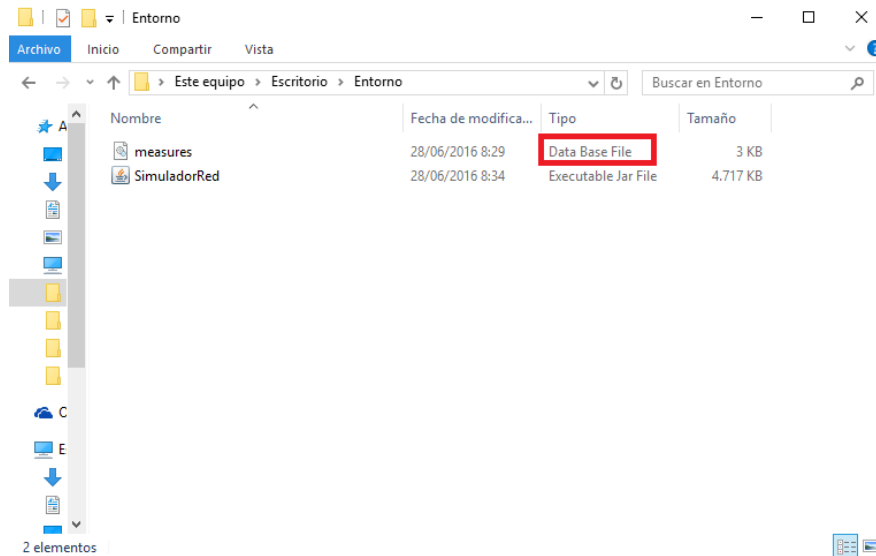


Figura B1. Entorno de ejecución del simulador.

Como podemos observar junto al archivo *.jar* (el ejecutable) se encuentra un archivo llamado “*measures*”, este archivo no es más que la base de datos donde presumiblemente hemos almacenado nuestras medidas. Este archivo se puede editar con cualquier editor SQL pero hay que tener la precaución de si el programa editor cambia el formato, tendremos que renombrarlo como “*measures.db*”, así el sistema lo detectara como base de datos y no se creará otra vacía. Podemos observar en el recuadro rojo de la Figura B1 como el sistema operativo ha detectado correctamente nuestra base de datos.

Cuando tenemos ya el entorno preparado ejecutamos el archivo *.jar* iniciándose el simulador, en la figura B2, podemos observar su apariencia donde tenemos diferenciadas las dos principales partes de la ventana principal recuadradas en verde y en naranja.

- Recuadro Verde: Zona donde se engloban todos los botones del simulador. Tendremos la posibilidad de añadir Routers, Switch, enlaces Ethernet, enlaces serial, ver información acerca del software, Ver/Insertar medidas de modelos y por ultimo simular cualquier topología.
- Recuadro Naranja: Zona de trabajo del simulador, será donde visualicemos y configuremos nuestras redes en el simulador.

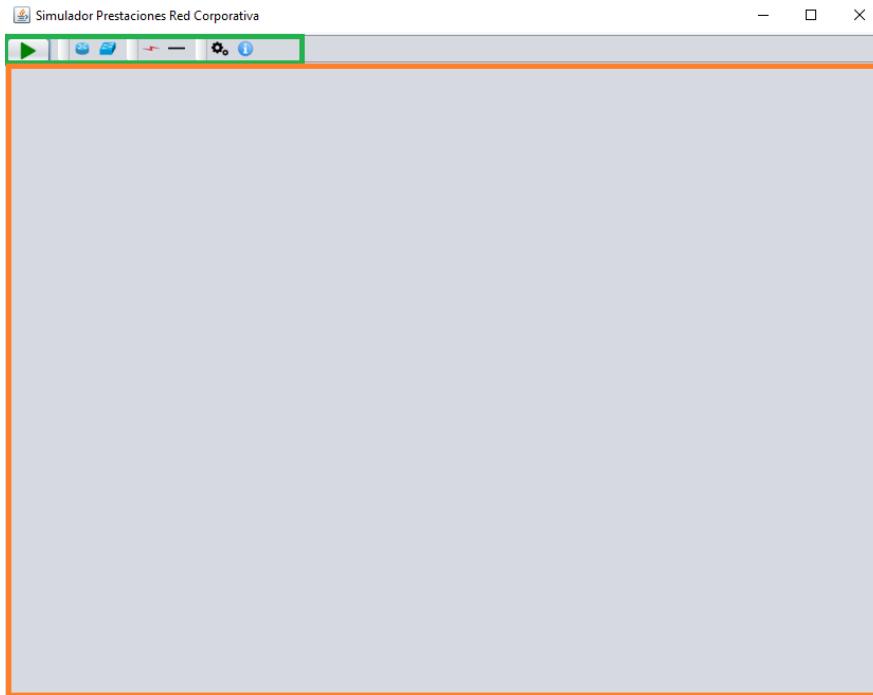


Figura B2. Apariencia del simulador.

En primer lugar deberemos comprobar que la base de datos se ha cargado bien en el programa, para hacer esto tenemos que pulsar sobre el sexto botón, el que parece unos engranajes. Se abrirá una nueva ventana donde veremos dos pestañas, nos posicionaremos sobre la pestaña “Lista de Modelos” y pulsaremos el botón “Actualizar Lista” de este modo en la tabla superior se mostrarán los distintos modelos de dispositivos de los que tenemos medidas en la base de datos.

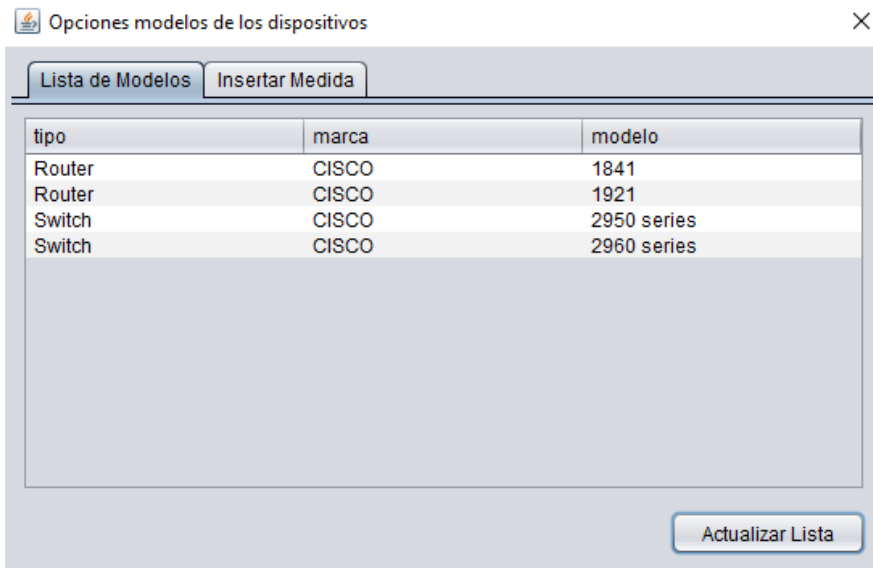


Figura B3. Pestaña Modelos de los Dispositivos disponibles.

Opcionalmente, si cambiamos de pestaña podremos añadir medidas a la base de datos desde el programa.

The screenshot shows a dialog box titled "Opciones modelos de los dispositivos" with a close button (X) in the top right corner. It has two tabs: "Lista de Modelos" and "Insertar Medida", with the latter being active. The form is divided into two sections: "Características Dispositivo:" and "Características del Enlace:". Under "Características Dispositivo:", there are input fields for "Marca:" (CISCO), "Modelo:" (1921), and a dropdown for "Tipo:" (Router). Under "Características del Enlace:", there is a dropdown for "Tipo:" (Ethernet), an input field for "Velocidad (Mbps/Kbps):" (1000), a slider for "Utilización (%)" ranging from 0 to 80, and a dropdown for "Tamaño de Trama:" (1024). At the bottom, there is an input field for "Valor de la medida (µs):" (8.46) and a "Guardar en la DB" button.

Figura B4.Ejemplo Insertar Medida de un Dispositivo.

Una vez comprobada la base de datos e insertadas las medidas adicionales, obtenidas con la metodología propuesta en esta memoria, que creamos que vamos a emplear podemos empezar a crear nuestra red, para posteriormente simularla. Este es el orden que se propone para el correcto funcionamiento del software:

1. Añadir todos los dispositivos (Router/Switch) que queramos emplear.
2. Generar enlaces entre estos dispositivos
3. Iniciar la Simulación

Para ejemplificar los pasos propuestos vamos a realizar la simulación de una topología simple paso a paso.

El primer paso es añadir un dispositivo debemos pulsar sobre el botón con forma de Router o Switch y seleccionar el modelo:

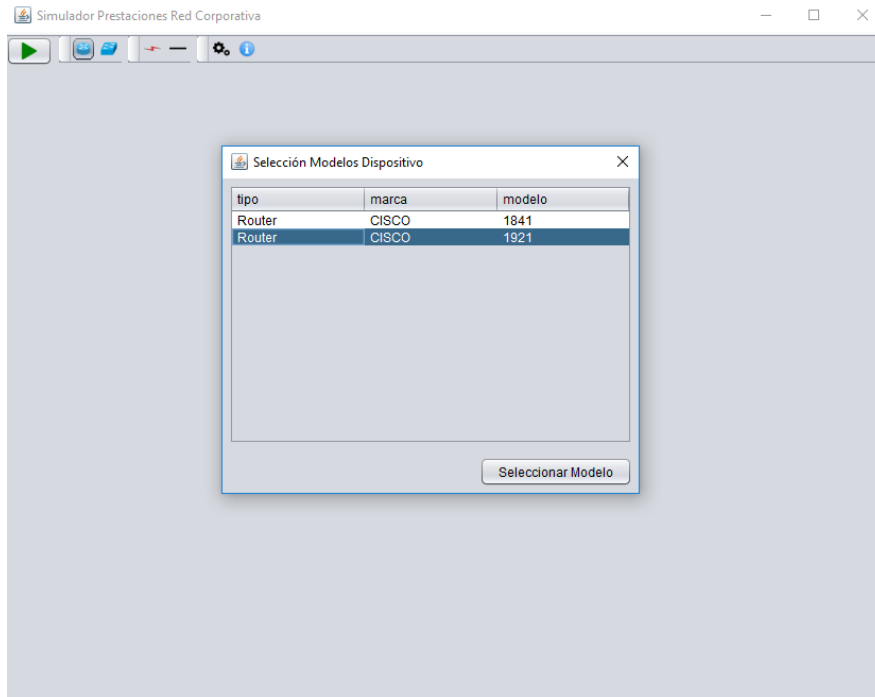


Figura B5. Añadir Router/Switch a la red.

Tras crear varios dispositivos nos disponemos a realizar un enlace Ethernet entre ellos pulsando sobre el enlace de color negro, seleccionamos los dos dispositivos a enlazar seleccionamos la velocidad de las interfaces de ambos dispositivos y su utilización y pulsamos sobre Enlazar:

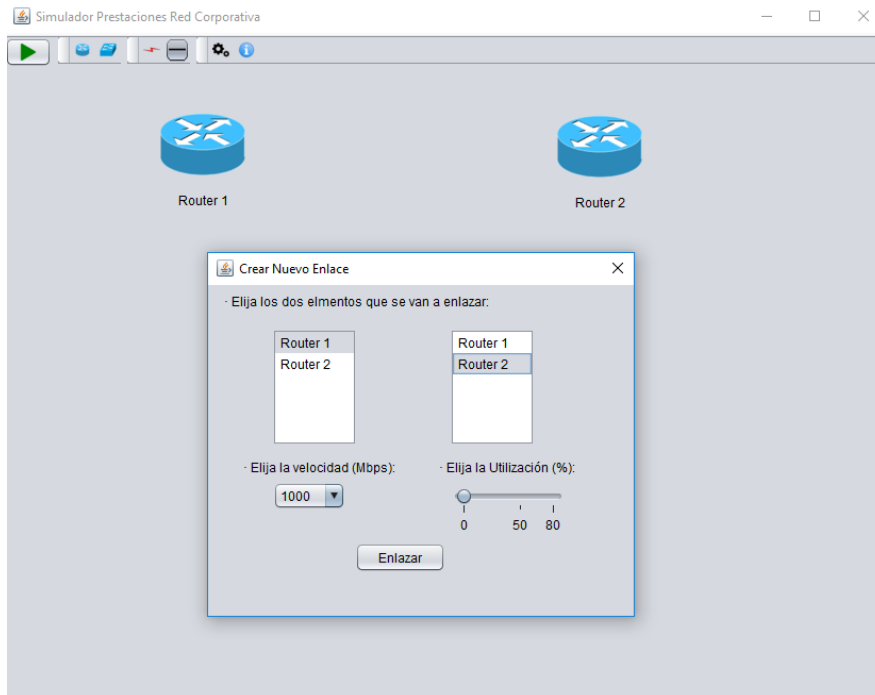


Figura B6. Enlazar varios dispositivos de la red.

En la zona de trabajo (recuadrada en la Figura B2 de color naranja) tenemos los dispositivos, bien Router o Switch, si pulsamos sobre él veremos sus detalles la marca el modelo y las diferentes interfaces, que definen los enlaces entre los dispositivos.

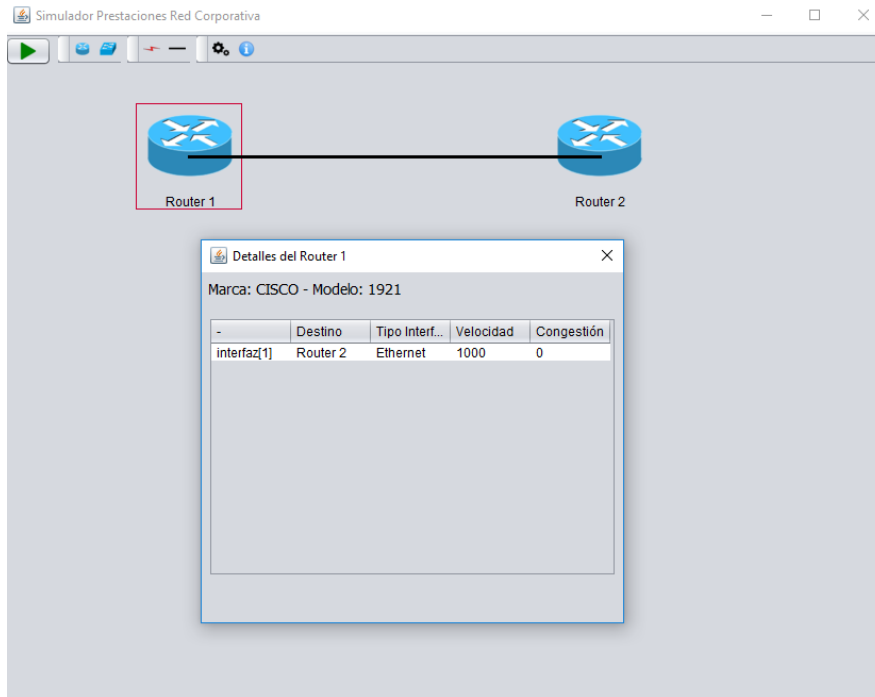


Figura B7. Detalles Router/Switch.

En la imagen anterior podemos observar que ya tenemos una topología simple, dos Routers y un enlace entre ellos, podemos seleccionar el botón “play” para iniciar la simulación:

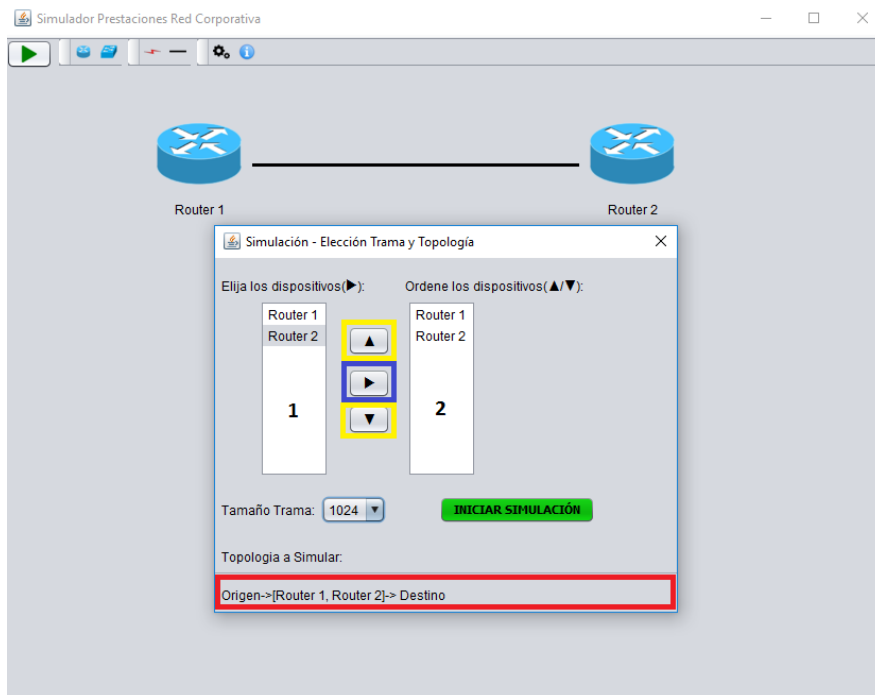


Figura B8. Simulación: Selección Topología.

En la pestaña que aparece para configurar la simulación vemos 2 listas numeradas, en la primera se muestran todos los dispositivos que tenemos en la zona de trabajo, seleccionando cualquiera y pulsando el botón recuadrado de azul pasaremos el dispositivo a la segunda lista donde tenemos que indicar el orden de los dispositivos para determinar la topología se puede modificar el orden de los Routers en la segunda lista seleccionando un dispositivo y pulsando cualquiera de los botones recuadrados de amarillo, en todo momento vemos en la parte de la ventana recuadrada de rojo la topología que se va a simular. Desde esta ventana también

seleccionamos el tamaño de paquete que será el mismo para toda la red y empezamos a simular cuando pulsemos sobre el botón “Iniciar simulación” de color verde en la pantalla.

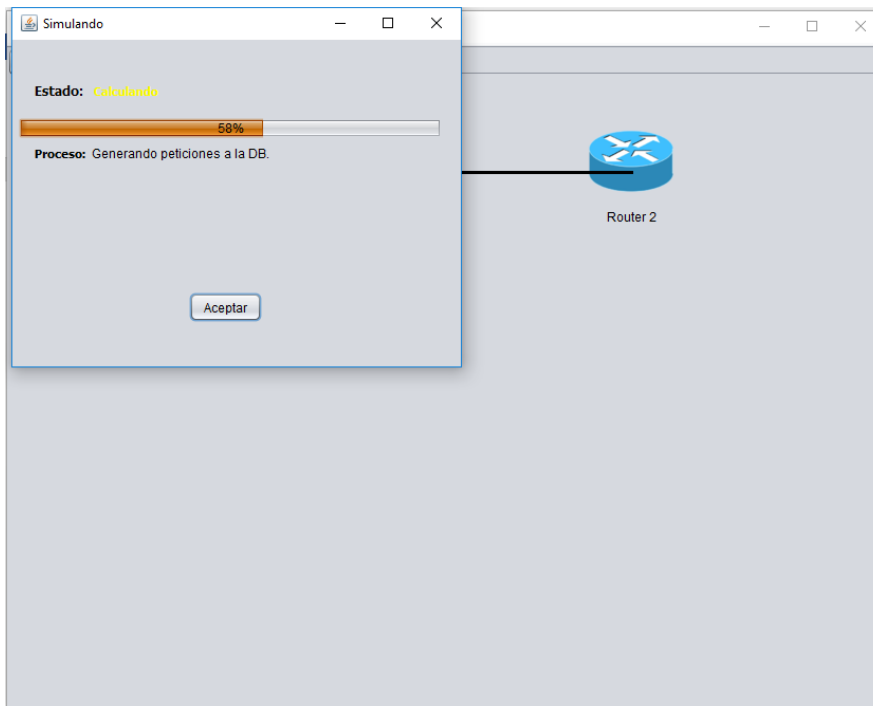


Figura B9. Obtención resultados de la simulación.

Después de pulsar aparecerá una ventana como la de la Figura B9 que realizará las operaciones pertinentes para poder calcular el resultado cuando esta barra llegue al 100% nos aparecerá:

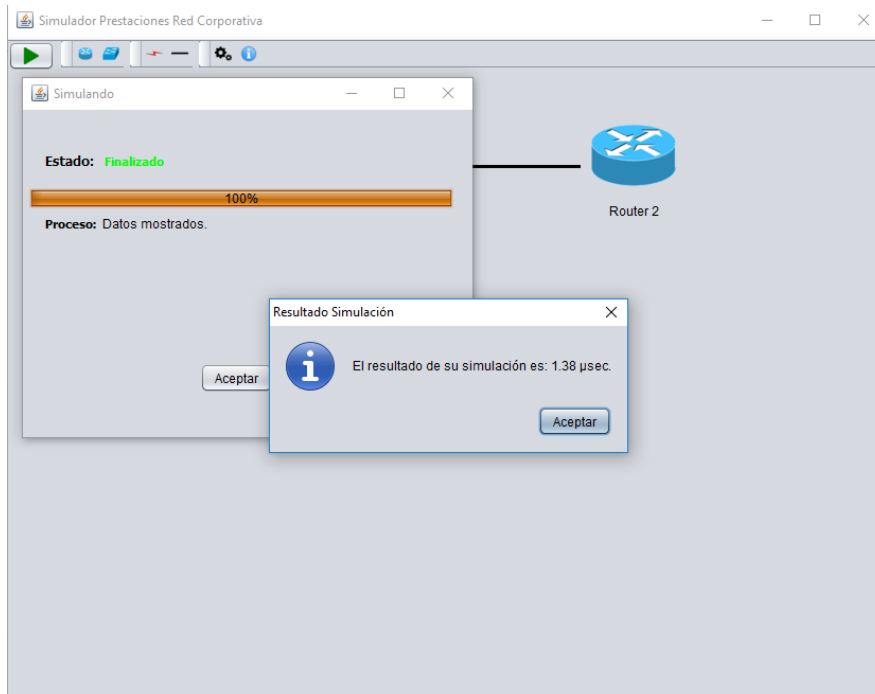


Figura B10. Resultados Simulación.

Siguiendo estos sencillos pasos hemos logrado simular una topología sencilla, el programa no limita la complejidad de la red mostrada en la zona de trabajo, por tanto podemos crear redes todo lo complicadas que queramos.