ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN DE BARCELONA

PROYECTO FINAL DE CARRERA

Evaluación de la herramienta GNS3 con conectividad a enrutadores reales

AUTOR:

Lisset Díaz Cervantes

DIRECTOR:

Xavier Hesselbach Serra

INDICE

CAPÍTULO 1: PREÁMBULO 5

1.1 INTRODUCCIÓN 5
1.2 MOTIVACIÓN 5
1.3 OBJETIVOS 6
1.4 CONTENIDO DE LA MEMORIA 6

CAPÍTULO 2: CONCEPTOS GENERALES 8

2.1 CISCO IOS 8
2.1.1 Características del Cisco IOS 8
2.1.2 Versiones del Cisco IOS 11
2.2 MODOS DE OPERACIÓN DE LOS DISPOSITIVOS DE RED 12
2.3 MEMORIAS DE LOS DISPOSITIVOS DE RED 13
2.4 CABLEADO EN DISPOSITIVOS DE RED 13
2.4.1 Cableado para la configuración de un router real 13
2.4.2 Cableado para conectar dispositivos reales 14
2.5 ARQUITECTURA DEL EMULADOR 15
2.5.1 Dynamips 15
2.5.1.1 IDLE-PC 16
2.5.1.2 Herramientas de optimización del uso de memoria 17
2.5.2 Dynagen 17
2.5.2.1 Network File 17
2.5.3 GNS3 18
2.6 HERRAMIENTAS Y PROTOCOLOS ADICIONALES USADOS EN ESTE TRABAJO 19
2.6.1 Protocolo ICMP 19
2.6.2 Protocolo de enrutamiento RIP versión 2 19
2.6.2.1 Configuración de RIP versión 2 19
2.6.3 Protocolo SNMP 20
2.6.3.1 Configuración de protocolo SNMP 21
2.6.4 IPERF 21
2.6.4.1 Configuración e instalación de IPERF 22
2.6.5 MRTG: Multi Router Traffic Grapher 22
2.6.5.1 Configuración e instalación de MRTG 23
2.6.6 RTR: Response Time Reporter 24
2.6.6.1 Configuración de RTR 24

CAPÍTULO 3: HERRAMIENTA GNS3 25

3.1 REQUERIMIENTOS DEL SISTEMA 25
3.1.1 Requerimientos del sistema en Windows XP 25
3.1.2 Requerimientos del sistema en Linux (Ubuntu 9.4) 26
3.1.3 Observaciones y Recomendaciones 28
3.2 INSTALACIÓN Y CONFIGURACIÓN 28
3.2.1 Instalación y configuración en Windows XP 28
3.2.1.1 Descargar el archivo de instalación 28
3.2.1.2 Instalar GNS3 29
3.2.1.3 Cargar de los CISCO IOS 31
3.2.1.4 Comprobar el *path* hacia Dynamips 33
3.2.2 Instalación y configuración en Linux (Ubuntu 9.4) 34
3.2.2.1 Descargar el archivo de instalación 34
3.2.2.2 Ejecutar el emulador 34
3.2.2.3 Cargar la Cisco IOS 36
3.2.2.4 Comprobar el path hacia Dynamips 36
3.3 USO DE GNS3 36
3.3.1 Emulación de Routers CISCO 37
3.3.2 Emulación de Switches Ethernet 40
3.3.3 Emulación de Dispositivos ATM y Frame Relay 42
3.3.4 Simulación de PCs 42
3.3.4.1 Usando Virtual PC 42

3.3.4.2 Routers que actúan como PC 44
3.3.4.3 Usando un PC real 44
3.3.5 Emulación de Firewall PIX 44
3.3.6 Creación de Enlaces 45
3.3.7 Creación de enlaces a equipos reales 45
3.3.8 Almacenamiento de topología 47
3.3.9 Captura de datos 48
3.3.10 Operación en modo Hipervisor 50

CAPÍTULO 4: ANÁLISIS DEL RENDIMIENTO ENTRE EQUIPO REAL Y VIRTUAL 54

4.1 LATENCIA 54
4.1.1 Objetivo 54
4.1.2 Escenario de prueba 55
4.1.2.1 Configuración del escenario 55
4.1.3 Medición y tratamiento de datos 56
4.1.4 Captura de tráfico 58
4.1.5 Resultados 59
4.1.6 Conclusiones y observaciones parciales 59
4.2 THROUGHPUT (CAUDAL EFICAZ) 59
4.2.1 Objetivo 60
4.2.2 Escenario de prueba 60
4.2.2.1 Configuración del escenario 61
4.2.3 Medición y tratamiento de datos 63
4.2.4 Captura de tráfico 65
4.2.5 Resultados 66
4.2.6 Conclusiones y observaciones parciales 67
4.3 JITTER 67
4.3.1 Objetivo 67
4.3.2 Escenario de prueba 68
4.3.3 Medición y tratamiento de datos 68
4.3.4 Captura de tráfico 70
4.3.5 Resultados 70
4.3.6 Conclusiones y observaciones parciales 71
4.4 PROBLEMAS DETECTADOS Y SOLUCIONES 71
4.4.1 Obtención del IOS C2621 71
4.4.2 Formato de la cisco IOS en el router 74
4.4.3 Igualdad de características en los equipos 74
4.4.3.1 Memoria Dinámica (DRAM) 75
4.4.3.2 Memoria no volátil (NVRAM) 75
4.4.3.3 Memoria FLASH 76
4.4.3.4 Memoria ROM 76

CAPÍTULO 5: ANÁLISIS DE LAS COMUNICACIONES INVOLUCRADAS EN LA SIMULACIÓN 78

5.1 PRUEBAS CON COMUNICACIONES INTERNAS 78
5.1.1 Objetivo 78
5.1.2 Escenario de prueba 78
5.1.2.1 Configuración del escenario 79
5.1.3 Medición y tratamiento de datos 80
5.1.3.1 Prueba 1: Latencia de ICMP 80
5.1.3.2 Prueba 2: Latencia de TCP 81
5.1.3.3 Prueba 3: Latencia de UDP 82
5.1.3.4 Prueba 4: Throughput en la red 84
5.1.3.5 Prueba 5: Jitter en la red 85
5.2 PRUEBAS CON COMUNICACIONES EXTERNAS 85
5.2.1 Objetivo 85
5.2.2 Escenario de prueba 86
5.2.2.1 Configuración del escenario 87
5.2.3 Medición y tratamiento de datos 87
5.2.3.1 Prueba 1: Latencia de ICMP 87
5.2.3.2 Prueba 2: Latencia de TCP 87
5.2.3.3 Prueba 3: Latencia de UDP 88

5.2.3.4 Prueba 4: Throughput en la red 89
5.2.3.5 Prueba 5: Jitter en la red 89
5.3 CAPTURA DE TRÁFICO 90
5.3.1 Prueba 1: Latencia de ICMP 90
5.3.2 Prueba 2: Latencia de TCP 92
5.3.3 Prueba 3: Latencia de UDP 93
5.3.4 Prueba 4: Throughput de la red 94
5.3.5 Prueba 5: Jitter en la red 95
5.4 COMPARACIÓN DE RESULTADOS 95
5.4.1 Prueba 1: Latencia de ICMP 95
5.4.2 Prueba 2: Latencia de TCP 96
5.4.3 Prueba 3: Latencia de UDP 96
5.4.4 Prueba 4: Throughput en la red 96
5.4.5 Prueba 5: Jitter en la red 97
5.5 CONCLUSIONES 97
5.6 OBSERVACIONES 98
5.6.1 Asignaciones de direcciones MAC 98

CAPÍTULO 6: ANÁLISIS LA DE COMUNICACIÓN ENTRE VARIOS EMULADORES .. 100

6.1 OBJETIVO 100
6.2 ESCENARIO DE PRUEBA 100
6.2.1 Configuración de escenario 101
6.3 CAPTURA DE TRÁFICO 102
6.4 GRÁFICAS MRTG DE TRÁFICO TRANSPORTADO 103
6.4.1 Prueba 1: Latencia de ICMP 103
6.4.2 Prueba 2: Throughput en la red 104
6.4.3 Prueba 3: Jitter en la red 104
6.5 CONCLUSIONES 105

CAPÍTULO 7: CONCLUSIONES Y RECOMENDACIONES 106

CAPÍTULO 8: LÍNEAS FUTURAS 108

CAPÍTULO 9: BIBLIOGRAFÍA 109

CAPÍTULO 1: PREÁMBULO

1.1 Introducción

La implantación de una red o su mejora no es tarea fácil ya que además de requerirse conocimientos de planificación, diseño e implementación se requiere de una gran inversión económica a la hora de adquirir dispositivos de red que cumplan unas necesidades específicas. En este sentido, cada vez es más común que las empresas usen *"testbeds"* para probar nuevos conceptos o nuevas funcionalidades que implementarán en sus redes en el futuro, con el fin de reducir riesgos de inversiones fallidas.

Los *"testbeds"* solían hacerse sobre prototipos, es decir, se realizaban pruebas sobre equipos reales con el fin de ir mejorando su rendimiento. Para evitar usar equipos reales y así abaratar costes en las pruebas, se crearon alternativas a los prototipos como son los emuladores y los simuladores, cuyo uso es cada vez más extendido debido a su gran evolución y constante mejora.

Un emulador permite duplicar las funcionalidades de un sistema usando otro diferente, de modo que el segundo sistema se comporte exactamente como el primero. En la actualidad, existen numerosos emuladores que permiten imitan el comportamiento de programas e incluso toman la forma de dispositivos hardware reales. Algunos ejemplos serían *VMware* que emula sistemas operativos ó **GNS3** [03][04] que emula dispositivos de red a partir de imágenes reales de IOS CISCO y que será el tema principal de este trabajo.

Un simulador permite representar ciertas características o comportamientos claves (modelo) del sistema físico o abstracto que se quiere simular, por lo tanto, el sistema resultante será más limitado. Un ejemplo de un simulador de routers sería *BOSON* [31] diseñado con fines académicos para soportar las funcionalidades requeridas para la obtención de certificaciones CISCO como son CCNA ó CCNP.

GNS3 es un emulador libre, con una GUI muy amigable, fácil de instalar y manejar pero su funcionamiento se basa en la obtención de una IOS de CISCO la cual es propietaria y requiere de permisos para su obtención. Una alternativa a esta deficiencia sería usar *Vycatta* [32]. *Vycatta* es un sistema operativo que puede ser instalado en cualquier ordenador para dotarlo de funcionalidades propias de un router, posee una *CLI* (Comand Line Interface) propietaria parecida a la implementada por JUNIPER en sus equipos. Este sistema operativo se encuentra disponible de forma gratuita en Internet en su versión más limitada, si se desea obtener funcionalidades más complejas este software deja de ser gratuito.

1.2 Motivación

GNS3 es una aplicación que tiene gran cabida en el ambiente empresarial ya que su uso reduce el coste de implementación de las redes. Con GNS3 es posible probar y experimentar nuevas funcionalidades de los CISCO IOS sin correr el riesgo de poner en peligro la integridad de una red real con configuraciones erróneas, además permite asegurarse de no hacer un planeamiento y diseño inexactos para ciertos requerimientos. Por otro lado, en el ambiente académico GNS3 puede ser usado como

una herramienta de entrenamiento en el estudio del networking, haciendo que los estudiantes tengan acceso más fácil a dispositivos de red tan ampliamente utilizados como lo con los routers de CISCO.

No cabe duda que CISCO es una empresa líder en tecnologías de redes y que el uso de sus dispositivos de red es ampliamente extendido en todo el mundo, con lo cual un emulador que emule sus equipos, como es el caso de GNS3, necesita ser estudiado.

Por lo tanto, el emulador **GNS3** se presenta como una aplicación de gran utilidad tanto en el ambiente empresarial como en el académico, cuyas capacidades merecen ser estudiadas y analizadas profundamente para conocer cuál es el rendimiento de los dispositivos que emula y cuál es el comportamiento cuando se forman redes con dispositivos emulados o con dispositivos reales.

1.3 Objetivos

El presente PFC tiene los siguientes objetivos:

a. Profundizar es el estudio de las capacidades que ofrece el entorno GNS3 para la simulación de redes con dispositivos de red emulados.

b. Detallar el uso de las capacidades del entorno GNS3.

c. Estimar los requerimientos de procesamiento que un host con GNS3 debe tener para conseguirse un buen funcionamiento del sistema.

d. Analizar definiciones y medidas de rendimiento realizadas en routers emulados usando GNS3, basándose en las RFC's propuestas por el IETF, para poder describir su rendimiento.

e. Comparar el rendimiento de los routers emulados usando GNS3 cuando se cambian las características de procesamiento del emulador.

f. Estudiar la transmisión de datos (trafico, protocolos, tablas de enrutamiento, etc) existente entre routers emulados usando GNS3.

g. Estudiar la transmisión de datos (trafico, protocolos, tablas de enrutamiento, etc) existente entre routers emulados usando GNS3 y routers reales.

h. Analizar el rendimiento de redes formadas por enrutadores emulados usando GNS3 y redes mixtas que también contienen routers reales.

i. Mostrar la posibilidad de funcionamiento conjunto de varios emuladores GNS3.

1.4 Contenido de la memoria

El presente PFC consta del Capítulo 1, denominado *Preámbulo* donde se realiza una introducción al tema a tratar y se comentan los motivos y objetivos para plantear y realizar este Proyecto Fin de Carrera. En el Capítulo 2, denominado *Conceptos Generales*, se estudia el concepto de IOS en un router CISCO, los modos de operación y las memorias de los dispositivos de red, se detalla el cableado utilizado entre los dispositivos de red, la arquitectura de la herramienta GNS3 y las

herramientas adicionales utilizadas en la realización de este proyecto. En el Capítulo 3, denominado *Herramienta GNS3*, se estudian los requisitos del sistema para usar GNS3, su instalación y configuración tanto en el sistema operativo Windows como Linux; así como también se describen los pasos a seguir para la emulación de routers, switches, PCs, etc. En el Capítulo 4, denominado *Análisis del Rendimiento entre equipo real y virtual*, se estudia los principales parámetros de rendimiento en una red como la Latencia, Throughput y Jitter en dispositivos emulados y reales; asimismo se describe los problemas y soluciones encontrados durante el análisis de los parámetros anteriores. En el Capítulo 5, denominado *Análisis de las Comunicaciones Involucradas en la simulación*, se estudia el comportamiento del GNS3 en redes formadas por dispositivos emulados y por dispositivos reales externos. En el Capítulo 6, denominado *Análisis de Comunicación entre varios Emuladores*, se estudia el comportamiento de varios escenarios emulados en GNS3 ubicados en diferentes emuladores que conforman una misma red. En el Capítulo 7, denominado *Conclusiones y Recomendaciones*, se dan las conclusiones y recomendaciones generadas como consecuencia del análisis del rendimiento y uso del GNS3. En el Capítulo 8, denominado *Líneas Futuras*, se comentan las nuevas líneas de estudio relacionados con el uso del emulador GNS3 en el campo de la investigación. Finalmente en el Capítulo 9 se indica la *Bibliografía* utilizada para la realización del presente PFC.

CAPÍTULO 2: CONCEPTOS GENERALES

2.1 Cisco IOS

El IOS (**I**nternetwork **O**perating **S**ystem) de CISCO es el sistema operativo usado en los dispositivos de red de CISCO: routers, switches, para la implementación de redes que forman la gran Internet (usando dispositivos CISCO). Una forma de acceso al CISCO IOS es usando CLI (**C**ommand **L**ine **I**nterface) el cual está basado en un conjunto de comandos. Estos comandos están ubicados en los diversos modos de operación definidos en cada equipo de red: usuario, privilegiado, configuración global.

La selección de una adecuada versión del CISCO IOS está relacionada con las características técnicas que pueden ofrecer los equipos de red, por ejemplo el soporte de IPv6, MPLS, DiffServ, Seguridad, entre otros. Además, como los routers tienen diversos tipos de interfaces, siendo las más comunes Ethernet y Serie, el IOS debe contener un grupo de drivers para soportar esta variedad de interfaces [01].

Las imágenes de CISCO IOS poseen nombres específicos y estandarizados por CISCO que reflejan sus principales características como la que plataforma soporta, las funciones realiza y la versión de IOS utiliza para facilitar su actualización.

Es importante conocer el nombre de la imagen y la versión que está corriendo en un router antes de comenzar a configurarlo. Para ello usamos el comando **show version** que se nuestra en la figura 2.1 [07]

```
MPLS_2#show version
Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-I-M), Version 12.3(3h), RELEASE SOFTWARE (fc2)
```

Figura 2.1: Comando que muestra la imagen y versión de un router.

Típicamente el nombre del archivo que contiene a una imagen CISCO IOS consta de una parte que indica las características de la IOS y otra que corresponde a la versión de la misma. En la figura 2.1, el nombre de la imagen IOS es **C2600-I-M**, la versión que se usa es la **12.3 (3h)** y el estado de la versión es **fc2**.

2.1.1 Características del Cisco IOS

Las características que posee una imagen IOS se indican en tres grupos de letras estandarizadas por CISCO, el primer grupo hace referencia a la plataforma, el segundo grupo a las funcionalidades de la imagen y el tercero al formato de la imagen.

Plataforma: Corresponde al primer grupo de caracteres e identifica la plataforma para la cual la imagen fue creada. La tabla 2.1 muestra los identificadores de plataformas más comunes. [06]

Identificador	Plataformas
As	Access Server 5200 series
Ca	Cisco Advantage
Cpa	Cisco Pro
Cs	Communication Server
c800, c1000, c1005, c1600	Cisco 800, 1003, 1004, 1005, 1600
c2500	Cisco 2500-2525 series routers

c2600	Cisco 2610-2613 series routers
c2800	Catalyst 2800
c2900	Cisco 2910, 2950
c29atm	Cisco 2900 ATM
c3620, c3640, c3800	Cisco 3620, 3640, 3800
c5200, c5300	Cisco AS5200, AS5300
c4000,c4500,c4700	Cisco 4000 series routers
c5fsfc	Catalyst 5000 series
c5rsm	Catalyst 5000 RSM
c5atm	Catalyst ATM
c6400s, c6400r	Cisco 6400 NSP, Cisco 6400 NRP
c6msm	Catalyst
c7000	Cisco 7000, 7010
c7200	Cisco 7200
Igs	IGS, 2500, 3000, and 5100 series routers
gs3	AGS and AGS+ gateway routers
gs7	7000 series gateway routers
Gsr	Gigabit Switch Router (12000)
ls1010	LightStream 1010
Mc3810	Multiservice Cisco 3810
Rpm	MGX 8850
RSP	Cisco 7500 series

Tabla 2.1: Principales identificadores de plataforma. [06]

Grupo de características: Este grupo representa las funcionalidades que un router con una IOS puede realizar, dichas funcionalidades pueden combinarse pero siempre dependen de que la plataforma pueda soportarlas. La tabla 2.2 muestra las funcionalidades más comunes. [06]

Grupo	Funcionalidades
A	APPN
A2	ATM
B	AppleTalk
BOOT	Boot image
C	Remote Access Server
D	Desktop subset (SNMP, IP, BRIDGING, WAN, Terminal Services, IPX, ATALK, ARAP)
D2	Desktop Plus
Dsv	Desktop/IBM
EBOOT	Ethernet boot image for MC3810
F	FRAD subset
G	ISDN subset
G2	Gatekeeper proxy, voice and video
G3	ISDN subset for c800 (IP, ISDN, Frame Relay)
I	IP subset
I2	IP subset for 3600
I3	Reduced IP subset with BGP/EGP/NHRP removed
Is	IP Plus
Is56i	IP IPSec 56
J	Enterprise subset (protocol translation)
Js40	Enterprise 40
Jsv56	Enterprise IPSec 56
Js	Enterprise Plus
Js40	Enterprise Plus 40
Js56i	Enterprise Plus IPSec 56
K	Kitchen sink
K1	Privacy key encryption (11.3 and above)
K3	Triple DES (11.3 and above)
K4	168-bit encryption with SSH
L	IpeXchange IPX, static routing, gateway
M	RMON
N	IPX

O	Firewall
O2	Firewall (3xx0)
O3	Firewall with SSH (36x0, 26x0)
P	Service provider (IP RIP/IGRP/EIGRP/OSPF/BGP CLNS ISIS/IGRP)
R	RSRB (remote source route bridging); this option can be added to other feature sets
S	Source route switch (SNMP, IP, BRIDGING, SRB)
V	VIP and dual RSP support
V2	Voice V2D
V3	Voice feature card
X	X.25 (11.1 and earlier), and on 12.0T on c800 series
X	Frame Relay in 11.2
X	H.323 Gatekeeper/Proxy for 2500, 3620, 3640, MC3810
Y	Reduced IP (SNMP, IP RIP/IGRP/EIGRP, BRIDGING, ISDN, PPP), for low-end routers
Y2	IP variant (SNMP, IP RIP/IGRP/EIGRP, WAN -X.25, OSPF, PIM)
Y3	IP/X.31
Y4	Reduced IP variant (Cable, MIBs, DHCP, EZHTTP)
Y5	Reduced IP variant (Cable, MIBs, DHCP, EZIP) for home office
Y6	Reduced IP variant (c800)

Tabla 2.2: Principales funcionalidades de los routers. [06]

En la actualidad, CISCO ha cambiado el modelo para nombrar las imágenes IOS incorporando el concepto de "paquetes heredables". Los paquetes se definen como "grupos de características" y tienen la finalidad de clarificarlas y simplificar la selección de una imagen. Cada vez que una nueva característica es agregada a un paquete, ésta es heredada por los paquetes de niveles superiores. La figura 2.2 muestra los ocho diferentes tipos de paquetes que existen para los routers CISCO y las relaciones entre ellos.[08]

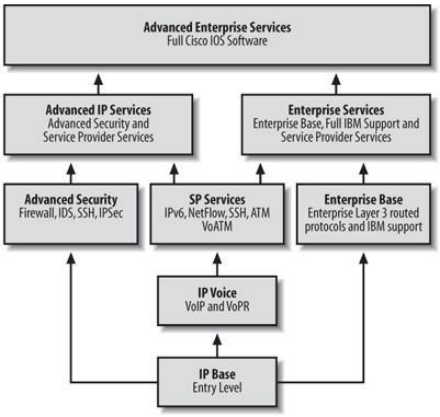


Figura 2.2: Nuevos paquetes y sus relaciones. (extraído de [06])

La tabla 2.3 muestra algunos ejemplos de nombres de imágenes CISCO IOS que usan ésta nueva convención:

Nombre	Grupo de características
C2600-ipbase-mz	IP Base
c2600-ipvoice-mz	IP Voice
c2600-advsecurity-mz	Advanced Security
c2600-spservices-mz	Service Provider Services
c2600-entbase-mz	Enterprise Base
c2600-advipservices-mz	Advanced IP Services
c2600-entservices-mz	Enterprise Services
c2600-adventerprise-mz	Advanced Enterprise Services

Figura 2.3: Nombres de imágenes según la nueva convención.

Formato de imagen: Generalmente en este grupo encontramos dos caracteres, el primero nos dice desde donde se ejecuta la imagen, y el segundo como se ha realizado la compresión de la imagen. La tabla 2.4 muestra caracteres más comunes.

Carácter	Location
F	Flash
M	RAM
R	ROM
L	Relocatable
C	Flash card (PCMIA)

Carácter	Compresión
Z	ZIP
X	Mzip
W	Stac

Tabla 2.4: Principales funcionalidades de los routers. [06]

2.1.2 Versiones del Cisco IOS

CISCO constantemente adapta su tecnología a los nuevos requerimientos de redes empresariales y de proveedores de servicios, en este sentido, cada nueva versión del software cisco IOS optimiza las redes IP y otorga avanzados servicios de red. El paso de una versión de IOS a otra nueva no es rápido, puesto que se requiere de constantes arreglos y mejoras hasta llegar a la estabilidad del software.

Las versiones del CISCO IOS constan de números y letras que denotan diferentes características de la misma según como se agrupan. Dichos números se pueden categorizar en los siguientes grupos según su ubicación: [07] [06]

Mayor lanzamiento: Se trata del primer número con punto decimal y sin ningún otro sufijo que aparece. Este número nos indica la versión del software CISCO IOS.

Lanzamiento en mantenimiento: Se denota por un número o letra entre paréntesis al costado del número de *"mayor lanzamiento"*, comienza en 1 y va creciendo con cada revisión sucesiva. Indica que se ha hecho una revisión a su *"mayor lanzamiento"* para arreglar los fallos pero sin incorporar nuevas funcionalidades. Cada lanzamiento en mantenimiento se basa en un mayor lanzamiento y ocurre periódicamente cuando se necesita.

Lanzamiento provisional: Se hace referencia agregando un punto y el número correspondiente después del número de *"lanzamiento en mantenimiento"*. Indica una revisión especial de su *"lanzamiento en mantenimiento"*. Cada lanzamiento provisional se basa en el mayor lanzamiento y ocurre periódicamente cuando se necesita.

Lanzamientos de despliegue temprano (ED): Se tratan de una o dos letras y números adicionales al final de los numero correspondientes al "Lanzamiento provisional" y denotan que nuevas funcionalidades y hardware han sido introducidos en cada *"lanzamientos en mantenimiento"* para su rápida entrega. Existen muchas series de ED cada una de ellas destinada a un grupo de clientes de cisco. La tabla 2.5 muestra las categorías existentes:

Identificador	Early Deployment Release (ED)
A	Tecnología Access server/dial
C	Routers Core
D	Tecnología xDSL
E	Específico-Empresa
F	Específico-Característica
G	Routers gigabyte switch
H	Tecnología SDH/SONET
N	Voz, conferencia, multimedia
P	Características de plataforma

S	Características de proveedores de servicios
T	Tecnología consolidada (siguiente mayor lanzamiento)
W	ATM/LAN o swithing de capa 3
X	Lanzamiento de vida corta
Y	Lanzamiento de vida corta

Tabla 2.5: Series de EDs existentes. [07]

En nuestro ejemplo, la IOS de la figura 2.1 tiene como *"mayor lanzamiento"* al número **12.3** y como *"lanzamiento en mantenimiento"* al **(3h)**, en este caso, no se cuenta con números de *"lanzamiento provisional"* ni de despliegue temprano *(ED)*.

Finalmente, con respecto al estado de la versión tenemos que diferenciar los siguientes cuatro estados: [07] [06]

First Commercial Shipment (FCS): Es una versión del software CISCO IOS que empieza a estar comercialmente disponible para los clientes de CISCO.

Limited deployment (LD): Es una versión del software CISCO IOS que ha sido introducida después de la FCS pero antes de que se alcance la GD. Contiene muchos fallos resueltos pero no incluye nuevas funcionalidades ni plataformas hasta que llegue al estado GD.

General deployment (GD): Es una versión del software CISCO IOS que se considera estable porque ha logrado un alto grado de satisfacción en el mercado y porque ha sido sometida a pruebas exhaustivas.

Early deployment (ED): Es una versión del software CISCO IOS que contiene las más actuales funcionalidades, por lo tanto, se encuentra en fase de pruebas y puede contener fallos.

CISCO proporciona en (*http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp*) una herramienta muy útil para encontrar rápidamente una versión de IOS que más se acerque a las necesidades de nuestras redes, se trata de un navegador que tiene la facilidad de hacer búsquedas de IOS por características deseadas o por versión, incluso permite comparar dos versiones de software o conocer los requerimientos mínimos de memoria para el funcionamiento de una IOS determinada. [06]

2.2 Modos de operación de los dispositivos de red

Los comandos del CISCO IOS están agrupados en los tres (03) modos de operación: modo usuario, modo privilegiado y modo de configuración [01].

Modo **usuario** es el primer modo que todo usuario va acceder. La mejor manera de conocer de que se está en éste modo es observando el símbolo **>** como prompt después del nombre del equipo de red.

Modo **privilegiado** permite a los usuarios ver la configuración del equipo de red (router o switch), restablecer el equipo de red e ingresar al modo de configuración. El modo privilegiado se puede distinguir observando el símbolo **#** como prompt después del nombre del equipo de red. El administrador del equipo de red puede habilitar la solicitud de un *password* antes de ingresar al modo usuario y/o privilegiado.

Modo de **configuración** permite modificar la configuración del sistema. El ingreso al modo de configuración se realiza desde el modo privilegiado ingresando el comando

configure terminal. El modo configuración se puede distinguir observando **(config)#** después del nombre del equipo de red.

Los comandos utilizados por CISCO IOS se detallan en [02] y serán utilizadas durante el desarrollo del presente Proyecto Fin de Carrera.

2.3 Memorias de los dispositivos de red

La arquitectura de los dispositivos de red es similar a un computador. Cada router tiene un CPU que varía en rendimiento y capacidad, dependiendo de la plataforma del router, e interfaces para conectar periféricos. Además del CPU y las interfaces, los routers necesitan de cuatro (04) tipos de memorias para su normal funcionamiento: ROM, Flash, RAM y NVRAM.

Memoria **ROM** es de sólo lectura, por lo que los datos no pueden ser escritos en este tipo de memoria. El software inicial que corre sobre un router CISCO es llamado *bootstrap software* y es almacenado en la memoria ROM; este software es invocado cuando se arranca el router. Los routers emulados con GNS3 ignoran la lectura de esta memoria.

Memoria **Flash** es usado para almacenar uno o más software CISCO IOS. En algunos sistemas, la memoria Flash puede contener el *bootstrap software* e incluso archivos de configuración o información del sistema. Los routers emulados con GNS3 carecen de esta memoria.

Memoria **RAM** ó DRAM es una memoria rápida que pierde la información almacenada cuando la energía del router es desactivada. Esta memoria es usada para mantener tablas y buffers del software CISCO IOS. GNS3 asigna esta memoria de forma virtual y es compartida por varios routers virtuales con la misma IOS, su tamaño es configurable dependiendo de los requerimientos de la plataforma.

Memoria **NVRAM** para almacenar la configuración de arranque, es decir los archivos que el CISCO IOS lee cuando el router se reinicializa. La memoria NVRAM puede ser obviado cuando se reinicializa el router, modificando el valor del registro interno al valor ox2142. Esta acción es necesaria cuando se requiere cambiar/eliminar los passwords definidos en un router. GNS3 usa el disco en vez de esta memoria.

2.4 Cableado en dispositivos de red

2.4.1 Cableado para la configuración de un router real

Para la configuración inicial de routers se necesitan los siguientes elementos:[09]

- Un router CISCO, en cualquiera de sus plataformas.
- Un cable de energía eléctrica.
- Un PC con un programa de emulación de terminal instalado.
- Un cable de *consola* con conectores de RJ45 a DB9 (cable celeste).

Los pasos a seguir son los siguientes:

1) Conectar el router al PC usando el cable de *consola* desde el puerto de consola (RJ45) del router al puerto COM del PC (DB9). Ver la figura 2.4

2) En el PC, cargar el programa de emulación de terminal, como el Hyperterminal de Windows. En la configuración del programa, tener cuidado en seleccionar el puerto que COM que fue usado para conectar el router.

3) Configurar los parámetros necesarios para el inicio del programa de emulación de terminal. Los parámetros se muestran en la figura 2.3. [09]

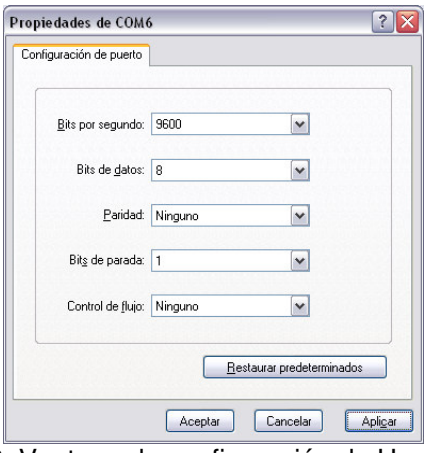


Figura 2.3: Ventana de configuración de Hyperteminal.

4) Encender el router moviendo el interruptor que se encuentra en la parte trasera a la posición de **ON**. Notar que en este momento los LEDs del chasis deben apagarse y encenderse de forma intermitente. [09]

5) A continuación se verán los mensajes típicos de encendido del router en la ventana del Hyperterminal. Finalmente, se nos indicará el encendido satisfactorio del router cuando el LED **SYS PWR** se enciende.[09]

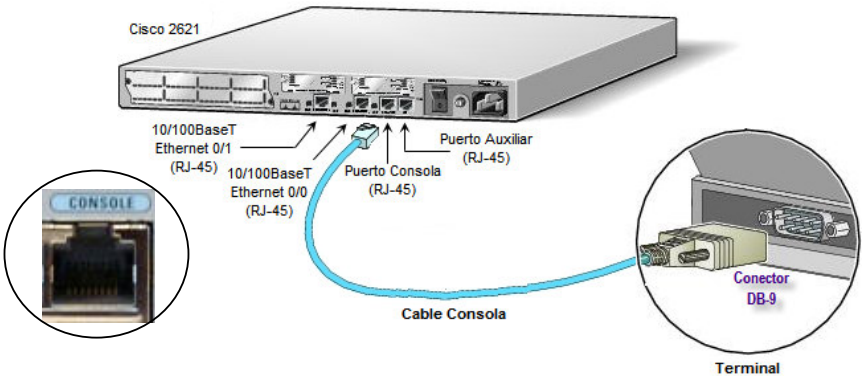


Figura 2.4: Cableado a interfaz de gestión

2.4.2 Cableado para conectar dispositivos reales

Para la conexión del emulador GNS3 a dispositivos de red reales, en este caso routers, se necesitan los siguientes elementos:[09]

- Un router CISCO, en cualquiera de sus plataformas.
- Un cable de energía eléctrica.
- Un PC con el emulador GNS3 instalado.
- Un cable UTP *cruzado* con conectores de RJ45 a RJ45.

Los pasos a seguir son los siguientes:

1) Conectar el router al PC usando el cable UTP *cruzado* desde el puerto FastEthernet (RJ45) del router al puerto FastEthernet de la PC (RJ45). Ver la figura 2.5. [09]

2) Cargar el programa de emulación de redes GNS3 en el PC y crear un nuevo enlace a equipos reales como se muestra en el apartado 3.3.7 de este trabajo. En la configuración del enlace, tener cuidado en seleccionar el nombre del puerto Fastethernet que fue usado para conectar el router.

3) Encender el router moviendo el interruptor que se encuentra en la parte trasera a la posición de **ON**. Esperar hasta que notemos que el LED **SYS PWR** se enciende para indicarnos que el router se inició satisfactoriamente.[09]

4) En el router, **configurar la dirección IP** en la interfaz Fastethernet de acuerdo a la topología creada en el emulador. Para la configuración seguir los pasos del apartado 4.1.2.1.

5) En el PC, **no asignar ninguna dirección IP** a la interfaz Fastethernet, ya que dicha dirección IP será asignada virtualmente tras la configuración del router virtual emulado por GNS3.

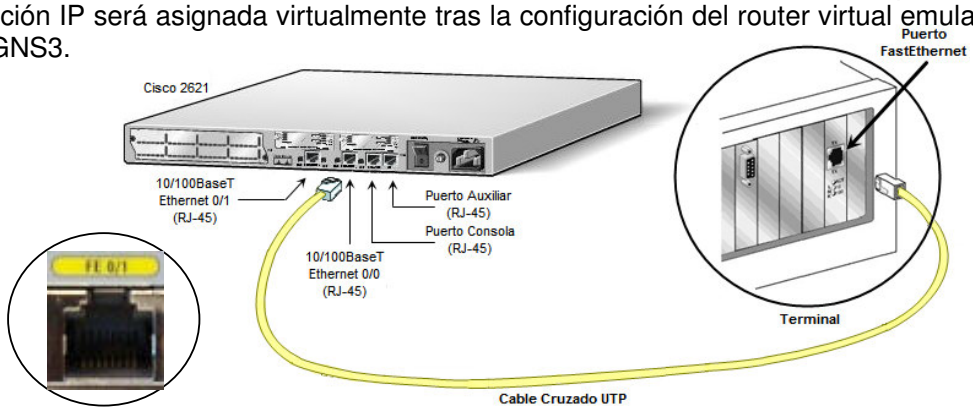


Figura 2.5: Cableado a interfaz de datos

2.5 Arquitectura del emulador

2.5.1 Dynamips

Dynamips[03] es el motor de emulación que permite emular diferentes plataformas hardware usando imágenes de sistemas operativos de CISCO en un mismo host. Entre dichas plataformas se encuentran los Routers 1700, 2600, 3600, 3700 y 7200. Por otro lado, puede emular switches Ethernet, Frame-Relay y ATM con funcionalidades básicas.

En cuando a la emulación de switches, Dynamips no es capaz de emular switches Catalyst sino que provee una versión limitada de un switch virtual, cuyas limitaciones pueden ser resueltas usando métodos alternativos como la emulación de NM-16ESW que el emulador sí soporta. Por otro lado, Dynamips tampoco es capaz de emular Firewalls PIX, para ello se usa el emulador PEMU a través de Dynagen. Los términos NM-16ESW y PEMU se explicarán en las secciones 3.2.2 y 3.1.1.2 respectivamente con más detalle.

Inicialmente Dynamips consume grandes cantidades de CPU del PC emulador, esto se debe principalmente a que realiza la emulación de los routers instrucción por

instrucción y a que no puede saber cuando un router virtual está inactivo, de modo que ejecuta instrucciones como si la imagen del IOS estuviera realizando algún trabajo útil. Para resolver el problema del excesivo uso de CPU se crea un proceso llamado IDLE-PC.

Por otro lado, Dynamips también consume memoria RAM del PC emulador, ya que, en teoría cada router virtual debe tener a su disposición, como mínimo, toda la cantidad de memoria RAM que necesita para poder trabajar, por lo tanto, esta cantidad se hace impráctica si se requieren emular redes con varios routers. Para resolver el problema del excesivo uso de memoria del PC emulador se usan herramientas que permiten compartir la memoria del mismo entre varios routers emulados con la misma IOS, y herramientas que usan el disco en vez de la memoria del emulador.

2.5.1.1 IDLE-PC

Se trata de una herramienta que realiza un análisis en el código de una imagen IOS para determinar los puntos más probables que representen un bucle de inactividad, de modo que, cuando se detecten, haga que los routers virtuales "duerman" durante ese instante. Es decir, IDLE-PC [03] [04] ayuda a Dynamips a emular el estado inactivo de la CPU virtual de un router.

Algunas características adicionales de este proceso son las siguientes:

- La aplicación de un mal valor de IDLE-PC hace que la PC del emulador trabaje entre 60% - 100% cuando emula un solo un router, mientras que un buen valor hace que sólo trabaje entre 1% -10% de la capacidad. Estos valores dependen de lo potente que sea el emulador usado.
- Está ligado a la versión de Dynamips que se usa, si se cambia de versión, es muy probable que se necesita cambiar de valor de IDLE-PC.
- Será diferente para IOS de diferentes versiones y por supuesto para diferentes plataformas. Se aplicará a un router virtual cada vez que use esa IOS.
- No son exclusivos de un PC o sistema operativo, por lo tanto, los archivos "dynagenidledb.ini" pueden ser copiados y compartidos y el valor de seguirá siendo bueno.

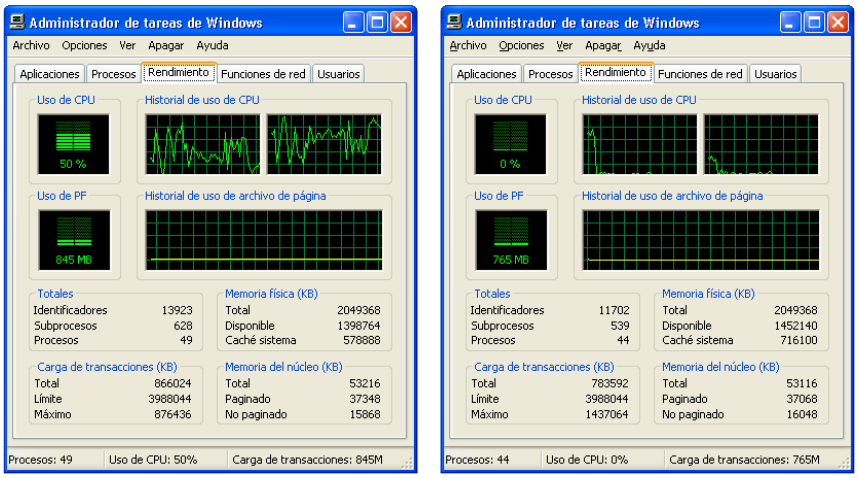


Figura 2.6 Uso de CPU sin y con un valor de IDLE-PC en PC Intel Core2 Duo @ 2.66Ghz – 2 Gb de RAM

2.5.1.2 Herramientas de optimización del uso de memoria

Las herramientas que Dynamips usa para optimizar el uso de memoria [03], tanto real como virtual, del host emulador son las siguientes:

Ghostios: [03] [04] Se encarga de reducir la cantidad de memoria real que se necesita del emulador para crear topologías con routers que corran a la vez, es decir, permite que el emulador comparta una parte de su memoria entre todos los routers que usen una misma imagen IOS de modo que cada router emulado no tenga que almacenar una copia idéntica de un mismo IOS en su memoria virtual. El resultado, en nuestro caso, es un archivo que contiene la región de memoria compartida ubicado en en el directorio "working", llamado c2600-i-mz.123-3h.image.ghost.

Sparsemem: [03] [04] Se encarga de reducir la cantidad de memoria virtual que usa un router emulado, es decir, la memoria necesaria para la ejecución de una IOS, ya que sólo asigna la cantidad de memoria que la IOS va a usar en un momento determinado y no toda la memoria RAM configurada, lo que permite crear más routers virtuales por proceso dynamips. Esta herramienta no está habilitada por defecto.

Mmap: [03] [04] Realiza la correspondencia de archivos temporales del disco con la memoria virtual configurada en los routers emulados, para que cuando se requiera leer estos archivos, el sistema operativo ponga en caché sólo las secciones de los mismos que están siendo utilizados. Estos archivos tienen la extensión "ram", el tamaño de la memoria RAM configurada y se encuentran en el directorio "working" creado por GNS3 en cada simulación.

2.5.2 Dynagen

Dynagen[03] es una interfaz escrita en Python que provee la gestión, mediante línea de comando (CLI), de las plataformas emuladas por Dynamips haciendo más fácil su uso. Usa el modo "Hypervisor" para comunicarse con Dynamips y ambas pueden correr en la misma o en diferente PC. También simplifica la gestión de las redes virtuales ya que implementa comandos para listar, iniciar, parar, reiniciar, suspender, reanudar los diferentes dispositivos emulados, además determina los valores de IDLE-PC y realiza capturas de paquetes.

A partir de sus últimas versiones, Dynagen es capaz de trabajar con el emulador de firewalls PEMU, el cual viene integrado en GNS3 dotando al emulador de capacidad de añadir Firewalls CISCO en las topologías. Además es capaz de conectar de forma transparente a Dynamips los diferentes dispositivos virtuales como switches Ethernet, Frame-Relay y ATM soportados por Dynamips.

Dynagen[03] usa un archivo de texto de fácil interpretación llamado "*Network File*", con extensión ".net", para conocer todas las características de hardware de los dispositivos de red a emular y realizar las interconexiones entre ellos.

2.5.2.1 Network File

Se trata de un archivo, escrito usando sintaxis INI (INI file syntax), que almacena la configuración de todos los dispositivos de red de la topología virtual a simular, como son los routers, switches y las interconexiones entre ellos. Este archivo puede especificar valores tan concretos como los descriptores de los adaptadores de red (NIO) que se encargan de la conexión con equipos reales o los puertos en los que trabajan dichos adaptadores de red de red, etc.

2.5.3 GNS3

GNS3[04] es una aplicación también realizada en Python que usa las librerías de Dynagen para crearle una interfaz gráfica (GUI). Sus principales funciones son editar el archivo de texto .net y realizar las operaciones del CLI hechas por Dynagen y Dynamips. Adicionalmente incorpora la capacidad de simular PCs.

La unión de Dynamips-Dynagen-GNS3, como se observa en la figura 2.7, crea una plataforma que permite el fácil diseño de topologías de red complejas ya que se realizan tan sólo arrastrando los componentes y dibujando líneas entre routers de forma intuitiva. Por lo tanto, GNS3 es idóneo para el entrenamiento de estudiantes que desean familiarizarse con dispositivos de red.

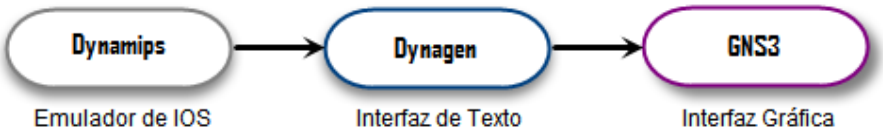


Figura 2.7 Plataforma GNS3

Las capacidades más resaltantes que podemos obtener de GNS3 y que han servido como punto de partida para tomar la decisión de estudiar más a fondo este simulador son las siguientes:

- Se encuentra disponible de forma gratuita en la red.
- Es fácil de instalar ya que todos los programas que necesita para funcionar se encuentran en un solo paquete de instalación.
- Está en constante actualización y periódicamente se puede encontrar versiones de la aplicación más robustas y con nuevas funcionalidades.
- Permite la conexión Telnet a la consola de un router virtual, de forma fácil directamente desde la interfaz gráfica.
- Alternativamente también permite trabajar directamente desde consola de gestión de Dynagen.
- Permite la comunicación entre redes virtuales con redes del mundo real.
- Es apropiado para simular redes de grandes tamaños ya que permite que un cliente GNS3 pueda correr en una máquina diferente a la que contiene al emulador Dynamips, repartiendo el procesamiento entre diferentes PCs.
- Puede capturar los paquetes que pasan por enlaces virtuales y escribir los resultados de la captura en archivos que pueden ser interpretados por aplicaciones como Wireshark o tcpdumps.
- Los foros de Internet evidencian que es una aplicación ampliamente utilizada.

GNS3 no es la única aplicación que brinda una GUI a Dynamips, existe otra con el nombre de *Dynagui* que realiza la misma tarea pero que se encuentra actualmente en fase de desarrollo y que no llega a implementar todas las funcionalidades que posee GNS3.

2.6 Herramientas y protocolos adicionales usados en este trabajo

2.6.1 Protocolo ICMP

Es un protocolo [11] usado para informar a la fuente acerca del procesamiento de los datagramas IP que envía, con el fin de saber si se han producido errores en la comunicación. En este sentido, ICMP no se usa estrictamente para dar fiabilidad a IP, ya que ésta debe ser implementada por los protocolos de nivel superior que usen IP. En la práctica, los routers generan mensajes ICMP para reportar errores, mientras que los hosts de destino sólo envían aquellos mensajes que pueden implementar.

Los mensajes ICMP se encapsulan en datagramas IP, por lo tanto, ICMP es parte de IP y debe ser implementado por él. Es decir, la cabecera IP siempre contendrá un número de protocolo de 1 (ICMP) y los datos IP serán los auténticos mensajes ICMP. Dichos mensajes ICMP poseen una estructura específica donde el tipo de mensaje que se transporta se representa por números 0, 8, 9, 10 y 13 al 18.

Unos tipos de mensajes ICMP muy conocidos son los llamados “Echo” usados principalmente para detectar si otro host de la red está activo aunque también puede medir la latencia de un paquete IP o auto-comprobar que la interfaz de red de nuestro host esta activa. La herramienta que implementa estos tipos de mensajes ICMP se llama **Ping**, en donde la fuente envía un mensaje *“echo request” (8)* al host de destino y el receptor cambia el tipo del mensaje a *"echo reply" (0)* para devolver el datagrama al host fuente.

2.6.2 Protocolo de enrutamiento RIP versión 2

El protocolo RIPv2 (**R**outing **I**nformation **P**rotocol) [06] [11] es un protocolo de enrutamiento basado en el algoritmo Bellman-Ford o vector distancia. Entre sus características se puede indicar que su criterio de selección de la mejor ruta está basado en determinar el menor número de saltos, provee mecanismos de autenticación, tiene una distancia administrativa de 120 (este parámetro indica el grado de confiabilidad del protocolo RIPv2 respecto a otros protocolos de enrutamiento) y va encapsulado en el protocolo UDP. RIPv2 es usado hasta un máximo de 15 saltos.

2.6.2.1 Configuración de RIP versión 2

La configuración de RIP [06] [12] en su versión 2 en cada uno de los router para el escenario de pruebas de las figuras 5.1 y 5.17 del capítulo 5 y 6 se produjeron usando los comandos que se muestran en la figura 2.8. Notar que para que este protocolo pueda funcionar, es necesario que cada router de la topología “anuncie” las redes que tiene directamente conectadas, de modo que después de unos segundos los routers de la red puedan aprender todas las redes existentes en la topología.

```
Barcelona#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Barcelona(config)#router rip
Barcelona(config-router)#version 2
Barcelona(config-router)#network 10.10.10.0
Barcelona(config-router)#network 10.10.10.4
Barcelona(config-router)#exit
Barcelona(config)#_
```

Figura 2.8: Configuración de RIPv2 en el router Barcelona.

Para visualizar las nuevas tablas de enrutamiento creadas después del aprendizaje de los routers usamos el comando mostrado en la figura 2.9 y 2.10. La tabla de enrutamiento convergente del router Madrid de la topología de la figura 5.1 es:

```
Madrid#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/30 is subnetted, 2 subnets
C       10.10.10.0 is directly connected, FastEthernet0/0
R       10.10.10.4 [120/1] via 10.10.10.2, 00:00:15, FastEthernet0/0
R    192.168.1.0/24 [120/2] via 10.10.10.2, 00:00:15, FastEthernet0/0
C    192.168.2.0/24 is directly connected, FastEthernet0/1
Madrid#
```

Figura 2.9: Tabla de enrutamiento convergente en Madrid.

La configuración del enrutamiento en el escenario 6.1 se hizo usando los mismos comandos que se mostraron en la figura 2.8. La tabla de enrutamiento final convergente en uno de los routers (Madrid_1) resultó de la siguiente forma:

```
Madrid_1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/30 is subnetted, 3 subnets
R       10.10.10.8 [120/2] via 10.10.10.2, 00:00:01, FastEthernet0/0
C       10.10.10.0 is directly connected, FastEthernet0/0
R       10.10.10.4 [120/1] via 10.10.10.2, 00:00:01, FastEthernet0/0
C    192.168.1.0/24 is directly connected, FastEthernet0/1
R    192.168.2.0/24 [120/3] via 10.10.10.2, 00:00:01, FastEthernet0/0
Madrid_1#
```

Figura 2.10: Tabla de enrutamiento convergente en Madrid_1.

2.6.3 Protocolo SNMP

Se trata de un estándar que define un conjunto de aplicaciones de gestión de red, utilizando los servicios ofrecidos por TCP/IP en la Internet. [13] [14]

La red SNMP (**S**imple **N**etwork **M**anagement **P**rotocol) la forman dos tipos de elementos: Los agentes, que son dispositivos pasivos ubicados en los host, routers, modems, multiplexores, etc., éstos envían información, por iniciativa propia o al ser interrogados de forma secuencial a otro elemento central; las estaciones administrativas ó equipos de gestión son los dispositivos que recolectarán la

información de gestión proveniente de los agentes. La principal desventaja de este protocolo es el exceso de tráfico que se genera.

Cada agente contiene una MIB (**M**anagement **I**nformation **B**ase) que es una base de datos compleja, estructurada en forma de árbol, con todos los objetos a gestionarse (información sobre variables/valores que se pueden adoptar). La versión 1 de la MIB puede manejar 114 objetos mientras que la versión 2 define 185 objetos, dichos objetos están agrupados de acuerdo al tipo de información que transportan.

2.6.3.1 Configuración de protocolo SNMP

La configuración de SNMP en los routers del escenario de pruebas del capítulo 6 (figura 6.1) se produjo siguiendo los siguientes pasos: [15]

1) En primer lugar configuramos el nombre que le daremos a nuestra comunidad SNMP, con la cual todos los routers que tengan configurado este nombre podrán leer o escribir datos SNMP entre ellos. Repetir el comando de la figura 2.11 en todos los routers de la topología.

```
Barcelona(config)#snmp-server community red-gns3 rw
```

Figura 2.11: Configuración del nombre de comunidad SNMP.

2) Conseguimos que el router envíe indicadores SNMP con sus datos recolectados a la estación de gestión con IP 192.168.1.2, usando la cadena indicadora de comunidad "public". Notar que no es necesario usar la verdadera cadena de comunidad, anteriormente configurada. Repetir el comando de la figura 2.12 en todos los routers de la topología.

```
Barcelona(config)#snmp-server host 192.168.1.2 public
```

Figura 2.12: Configuración de estación de gestión.

El comando anterior provoca que el servidor reciba indicadores SNMP que no necesita, desperdiciando el ancho de banda de la red; en este sentido, existe una forma de restringir la información que cada router envía al servidor pero esta solución implica configurar cada router de la red.

3) Una vez configurado quien recibe los indicadores, encendemos el envío de los mismos por parte del router. Repetir el comando de la figura 2.13 en todos los routers de la topología.

```
Barcelona(config)#snmp-server enable traps
```

Figura 2.13: Habilitamos el envío de indicadores.

El comando anterior enciende el envío de todos los indicadores existentes, en este caso también es posible encender solo indicadores específicos (ver bibliografía).

2.6.4 IPERF

Iperf [16] [17] es una herramienta de distribución libre, capaz de medir el ancho de banda, el retardo, el Jitter e incluso la pérdida de paquetes de un enlace entre dos PCs. Sus principales característica recaen en la facilidad con la que permite configurar sus diferentes parámetros, ya que posee una interfaz gráfica escrita en Java llamada

Jperf, y en la capacidad de poder enviar datos que usan tanto el protocolo TCP, como el protocolo UDP.

Iperf trabaja bajo el modelo de cliente-servidor, en donde el PC cliente inventa información sin sentido que intenta mandar a otro PC que actúa como servidor durante un tiempo determinado, para ello el cliente necesita saber la dirección IP del PC servidor. Básicamente el cliente es el que envía la información y el servidor es el que registra los datos que obtiene cuando le llegan los paquetes y los muestra.

2.6.4.1 Configuración e instalación de IPERF

Iperf se usará en los capítulos 4, 5 y 6. Los pasos que se siguieron para su instalación y configuración fueron:

1) Descargar jperf-2.0.2 desde http://iperf.sourceforge.net/ y descomprimir el archivo en cualquier ubicación.

2) Ir a la carpeta descomprimida llamada “jperf2.0.2” y hacer doble clic en "jperf.bat". Después de unos segundos aparecerá la ventana de Jperf. Cabe resaltar que la herramienta Iperf se encuentra en la carpeta /bin.

3) Repetir los pasos 1 y 2 tanto en el cliente como en el servidor.

4) A cada nodo del enlace le asignamos su papel que desempeñará durante la medida, en el caso de la configuración del cliente se debe indicar la dirección IP del servidor. Ver figura 2.14a y b.

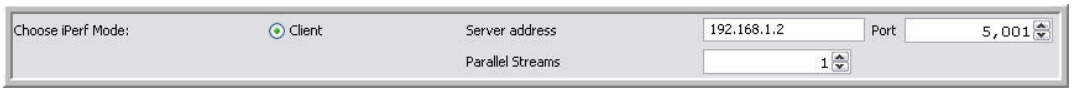


Figura 2.14a: Configuración del nodo como cliente.

Figura 2.14b: Configuración del nodo como servidor.

2.6.5 MRTG: Multi Router Traffic Grapher

Se trata de un monitor de dispositivos de red [18] que tiene la capacidad de mostrar, en tiempo real, todo el tráfico que pasa por cada una de sus interfaces. Los gráficos generados poseen la extensión PNG y están contenidos en páginas HTML que pueden ser vistas en cualquier navegador. MRTG puede contener los datos concernientes al tráfico visto durante semanas, meses e incluso años con un máximo de 2 años.

La única condición para que cualquier dispositivo que se encuentre en una red pueda ser monitorizado por esta aplicación es que debe soportar el protocolo SNMP ya que es éste protocolo el que obtiene los datos de tráfico de cada dispositivo de red que después serán leídos para crear los gráficos.

MRTG es una herramienta escrita usando el lenguaje Perl, es de distribución libre y puede correr tanto en Linux como en Windows.

2.6.5.1 Configuración e instalación de MRTG

A continuación se explican los pasos realizados para la instalación y configuración de la herramienta MRTG usada para las pruebas del capítulo 6. [18]

1) Descargar e Instalar Perl (ActivePerl-5.10.0.1005) desde http://www.activestate .com/activeperl/

2) Descargar y descomprimir MRTG (mrtg-2.16.2.zip) desde http://ftp.idilis.ro/mirrors/ mrtg/

3) Ir a la ubicación de …/mrtg-2.16.2/bin desde una ventana de comandos de Windows para comprobar que la instalación se ha realizado con éxito. Usar el siguiente comando:

```
C:\Programas\mrtg-2.16.2\bin>perl mrtg
Usage: mrtg <config-file>

mrtg-2.16.2 - Multi Router Traffic Grapher

Copyright 1995-2006 by Tobias Oetiker
Licensed under the Gnu GPL.

If you want to know more about this tool, you might want
to read the docs. You can find everything on the
mrtg website:

http://oss.oetiker.ch/mrtg/
```

Figura 2.15: Habilitamos el envío de indicadores.

4) Ahora debemos crear un archivo de configuración de MRTG para cada router de la topología, ingresando el respectivo nombre de la comunidad, la dirección IP del host, la ubicación del directorio de trabajo y el nombre del archivo de configuración. Si nos encontramos ante algún aviso de error, volver a lanzar el comando hasta un máximo de tres veces. Ver figura 2.16

```
C:\Programas\mrtg-2.16.2\bin>perl cfgmaker red-gns3@10.10.10.2 --global "WorkDir
: c:\Programas\mrtg-workDir" --output Barcelona.cfg
```

Figura 2.16: Creamos el archivo de configuración de MRTG.

Después de realizar este comando, encontraremos que se ha creado un archivo .cfg para cada uno de los routers en C:\Programas\mrtg-2.16.2\bin.

5) Empezaremos a recolectar datos de los routers escribiendo el siguiente comando. Repetir para todos los routers de la topología.

```
C:\Programas\mrtg-2.16.2\bin>perl mrtg Barcelona.cfg
```

Figura 2.17: Comando para recolectar datos con MRTG.

Con los comandos anteriores conseguiremos que se creen los archivos .html para cada interfaz de los routers, si los observamos notaremos que no aparecen datos, esto se debe a que el proceso de recolección de datos debe repetirse manualmente después de cierto tiempo. Podemos ver los archivos .html si navegamos hasta el directorio de trabajo, en nuestro caso mrtg-workDir.

6) Activar la recolección de datos de forma constante cada 5 minutos, para ello haremos que MRTG corra como un demonio. Abrir cada uno de los archivos .cfg creados y añadir la siguiente línea:

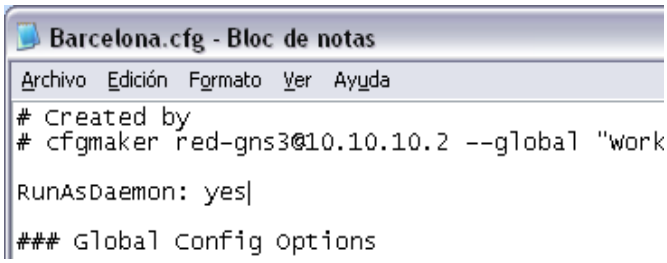


Figura 2.18: Archivo de configuración de Barcelona.

7) Volver a correr el archivo .cfg modificado desde una ventana de DOS. Usando el comando de la figura 2.17. En caso de cerrar la ventana, volver a reiniciar MRTG.

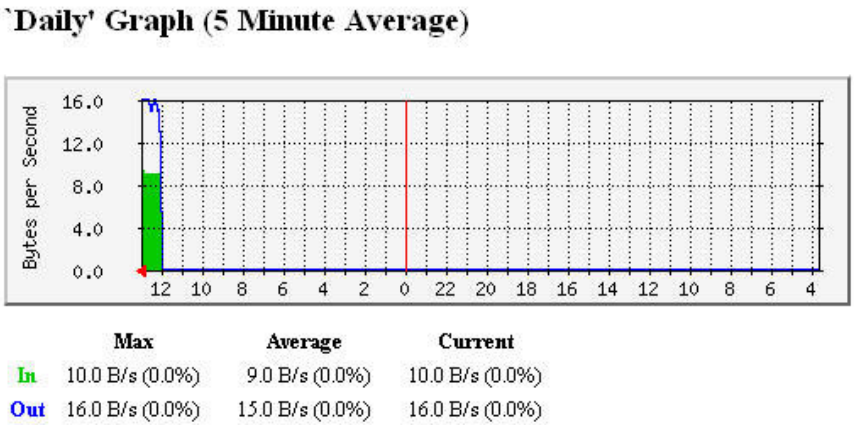


Figura 2.19: Tráfico inicial en el router Barcelona.

2.6.6 RTR: Response Time Reporter

Es una herramienta [19] [20] presente en los routers CISCO, incorporada desde la versión 12.0 del IOS hasta las versiones actuales, que permite que dichos dispositivos puedan responder a servicios que usualmente no soportan como, por ejemplo, diferentes tipos de tráfico IP como TCP ó UDP de modo que sea posible medir los tiempos de respuesta.

Con esta herramienta es posible analizar problemas de una red por medio de la supervisión de los tiempos de respuesta entre dispositivos de red y los retrasos sufridos en un camino entre un router origen y otro de destino, además la recolección de los tiempos de respuesta obtenidos en las diferentes pruebas en forma de estadísticas acumuladas tan detalladas permiten conocer a fondo el comportamiento de la red.

2.6.6.1 Configuración de RTR

Dado que para las diferentes pruebas realizadas en los capítulos 5 y 6 de este trabajo se usan configuraciones diferentes de RTR, explicaremos los pasos seguidos dentro del apartado correspondiente a cada una de ellas.

CAPÍTULO 3: HERRAMIENTA GNS3

3.1 Requerimientos del sistema

3.1.1 Requerimientos del sistema en Windows XP

Memoria RAM: Dynamips [03] asigna por defecto 16MB de memoria RAM al compilador JIT para que realice la compilación del código del simulador en sistema Windows. Además, cada imagen IOS de un router real requiere una cantidad determinada de memoria RAM para funcionar, por lo tanto, inicialmente, la suma de los valores anteriores sería la cantidad de memoria RAM real necesaria para la simulación de un router. En la práctica este valor es mucho menor debido a que Dynamips implementa herramientas que permiten una optimización del uso de la memoria del simulador. Dichas herramientas ya fueron explicadas en el apartado 2.5.1.2 de este trabajo.

CPU: En un principio, Dynamips usará mucha cantidad de CPU porque no sabe cuando el CPU virtual del router está inactivo, por lo tanto, ejecuta todas las instrucciones de las rutinas de inactividad de la IOS como si fueran instrucciones que realizan un trabajo real [03]. El cálculo del valor de IDLE-PC hará que el consumo de CPU del emulador baje drásticamente. Si se elige un buen valor, la utilización de CPU por cada router será baja con lo cual el funcionamiento de del emulador sea óptimo.

Disco: Se necesita 39,65 MB de espacio de disco para almacenar a la aplicación GNS3 y a sus dependencias y emuladores asociados. Además se necesita 0,19 MB para almacenar *WinPCAP*, lo que hace un aproximado de 40 MB de disco necesario. [03] Este parámetro no es determinante a la hora de escoger un buen host donde montar nuestra red virtual debido a que para la mayoría de PCs estos valores resultan fácilmente alcanzables.

Para poder estimar las capacidades recomendables para el buen funcionamiento de un equipo emulador, haremos un análisis comparativo donde se refleje el consumo de recursos cuando se emulen ciertos números de routers en equipos con características de procesamiento diferentes. Las gráficas 3.1 y 3.2 muestran las características del escenario usado para la prueba y los resultados finales obtenidos en 2 emuladores diferentes.

Características

OS: Windows XP
RAM: 0,99 GB
PROCESADOR: Intel(R)
CPU: 1,66 Ghz
PLATAFORMA: c2621
IOS: c2600-i-mz.123-3h
IOS RAM: 48MB
VALOR IDLE-PC: 0x80501884

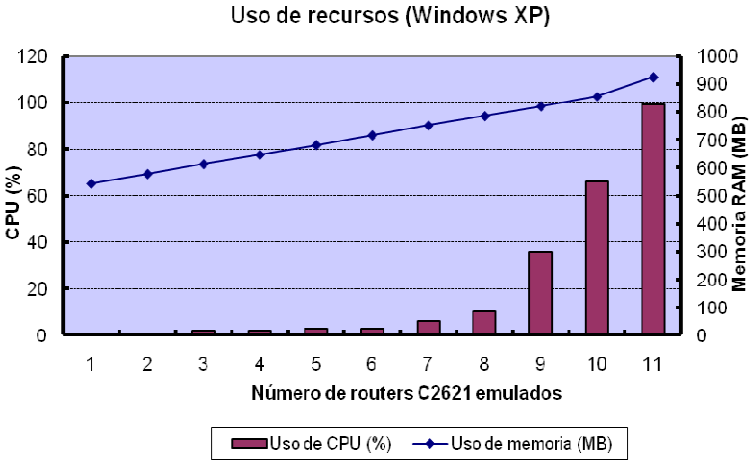


Figura 3.1: Primer Escenario y resultados de pruebas de uso de recursos.

Características

OS: Windows XP
RAM: 1,95 GB
PROCESADOR: Intel(R) Core 2 Duo
CPU: 2,66Ghz
PLATAFORMA: c2621
IOS: c2600-i-mz.123-3h
IOS RAM: 48MB
VALOR IDLE-PC: 0x80501884

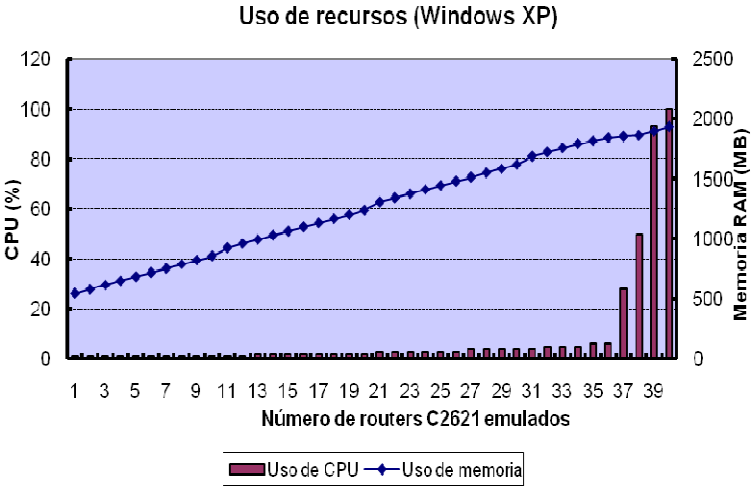


Figura 3.2: Segundo Escenario y resultados de pruebas de uso de recursos.

Las gráficas anteriores comprueban que el número de routers que puede soportar un emulador es directamente proporcional a sus capacidades, así un PC con procesador de 1,6 Ghz podría emular como máximo 11 routers y otro de 2 procesadores de aproximadamente 2,6 Ghz podría emular 40 routers, siempre y cuando se emule la misma plataforma, el mismo IOS con la misma RAM y se use el mismo valor de IDLE-PC.

En la prueba se observa que la RAM usada aumenta de forma progresiva a pasos de 35MB por router aproximadamente, este valor obtenido es inferior al configurado en los routers (48MB) con lo cual se confirma la eficiencia de las herramientas de optimación de recursos de Dynamips. Por otro lado, el hecho de que el uso de la CPU aumente de forma drástica cuando se ha usado casi toda la memoria RAM disponible nos indica que se está realizando “swapping”, es decir, se empieza a remplazar la memoria RAM por espacio en el disco.

Con respecto al comportamiento de GNS3 durante la prueba es importante resaltar que, pese a que la utilización de CPU parece que ha sido mínima durante gran parte de la prueba, esto no ha sido así, cada vez que se añadía un router o se abría una ventana de Telnet, la *utilización de CPU subía de forma abrupta* y fluctuaba por unos instantes, después se estabilizaba y descendía a valores mínimos nuevamente.

Cabe resaltar que todas las pruebas fueron realizadas cuando el emulador sólo tenía corriendo la aplicación GNS3 y el monitor de sistema de Windows para efectuar las mediciones. Además los routers no tenían configuración alguna.

3.1.2 Requerimientos del sistema en Linux (Ubuntu 9.4)

Memoria RAM: En Linux, la memoria RAM requerida teórica para la emulación de un router sería la que Dynamips asigna por defecto al compilador JIT (64MB) y la cantidad de RAM que cada imagen IOS requiere para funcionar en un equipo real, aunque, como ya hemos explicado, en la práctica se necesitan valores inferiores. [03]

CPU: En Linux también se toma en cuenta el valor de IDLE-PC para estimar los requerimientos de CPU del emulador. [03]

Disco: El espacio total necesario en disco para la instalación de GNS3 en Linux es de aproximadamente 117,2MB, valor que es mayor al requerido en Windows debido a la

necesidad de instalación adicional de dependencias. Este parámetro no es determinante a la hora de escoger un buen host de trabajo. [03]

Repetimos las mismas pruebas realizadas en el apartado anterior con dos equipos de características de procesamiento diferentes. Las gráficas 3.3 y 3.4 muestran los resultados obtenidos.

Características

OS: Ubuntu 9.10
RAM: 1,2 GiB
PROCESADOR: Intel(R)
CPU: 1,86 Ghz
PLATAFORMA: c2621
IOS: c2600-i-mz.123-3h
IOS RAM: 48MB
VALOR IDLE-PC: 0x80501884

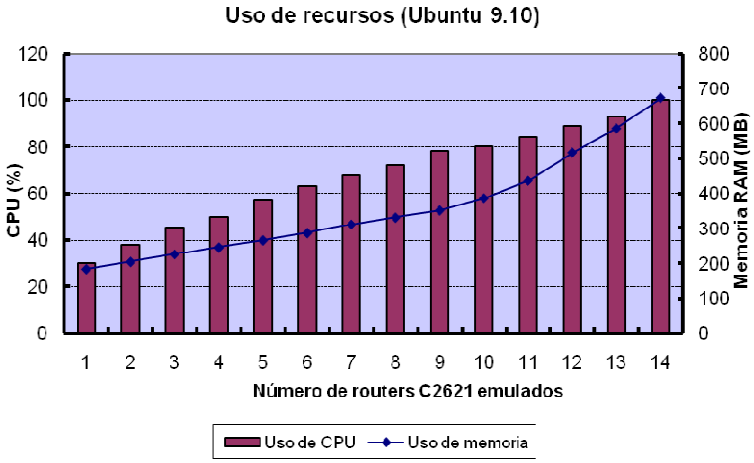


Figura 3.3: Primer Escenario y resultados de pruebas de uso de recursos.

Características

OS: Ubuntu 9.10
RAM: 1,9 GiB
PROCESADOR: Intel(R) Core 2 Duo
CPU: 2,66Ghz
PLATAFORMA: c2621
IOS: c2600-i-mz.123-3h
IOS RAM: 48MB
VALOR IDLE-PC: 0x80501884

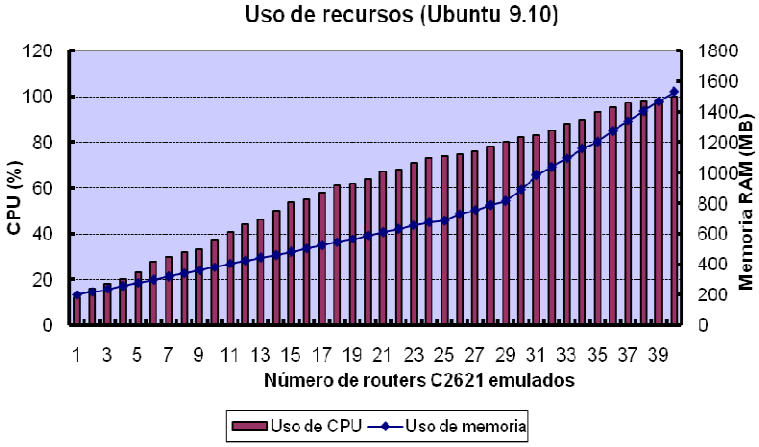


Figura 3.4: Segundo Escenario y resultados de pruebas de uso de recursos.

Los resultados obtenidos en Linux nos muestran un aumento progresivo del uso de la CPU cuando se añade un nuevo router, de modo que, podemos notar más claramente la estrecha relación entre la velocidad de la CPU de un emulador y la cantidad de routers que puede emular. Un emulador con algo menos de 2Ghz de CPU puede emular como máximo 14 routers y otro con 2 procesadores de 2Ghz cada uno aproximadamente podría emular 40 routers, siempre y cuando se emule la misma plataforma, el mismo IOS con la misma RAM y se use el mismo valor de IDLE-PC.

En esta prueba también se comprueba la eficiencia del uso de las herramientas de optimación de recursos de Dynamips, ya que al agregar un router a la topología, se aumenta aproximadamente en un 22 MB la RAM usada, en vez de los 48 MB configurados. Cabe resaltar que cuando la utilización de la CPU del emulador es aproximadamente 80%, la memoria RAM empieza a subir más rápidamente, por lo cual, éste podría ser un valor limitador de rendimiento. Este sistema es *más estable*

puesto que no se observaron cambios buscos en el uso de la CPU cuando se añadían nuevos elementos, sino que su aumento fue progresivo.

Cabe resaltar, otra vez, que todas las pruebas fueron realizadas cuando el emulador sólo tenía corriendo la aplicación GNS3 y el monitor de sistema de Linux para efectuar las mediciones. Además los routers no tenían configuración alguna.

3.1.3 Observaciones y Recomendaciones

Un emulador con sistema operativo Windows requerirá más memoria RAM (~35MB por router) que un emulador con sistema operativo Linux (~22MB por router) para un mismo número de routers emulados. Por otro lado, teniendo en cuenta nuestras pruebas, no es fácil estimar el número de routers que puede soportar eficientemente un emulador, con determinadas capacidades de procesamiento, que usa Windows. Sin embargo, para Linux podemos decir que un emulador con procesador Intel de 2 Ghz y 1 GB de RAM puede emular **10 routers** y otro con procesador Intel Core 2 Duo de 2,6 Ghz y 2 GB de RAM puede emular **28 routers**, para tener como máximo un 80% de CPU ocupada.

El PC que debemos elegir para la instalación de nuestro simulador de red ***GNS3*** debe tener un buen balance entre CPU y Memoria. Algunas recomendaciones adicionales serían:

- Elegir un buen valor de IDLE-PC. Un valor aceptable es aquel que hace que el CPU use entre 0% y 50% para un sólo router y un buen valor es aquel que sólo usa 10%.

- Usar una IOS que requiera las más bajas cantidades de RAM para funcionar pero que cuente con las funcionalidades que se necesita.

- No crear topologías que usan más del 80% del CPU del emulador, ya que el uso de capacidades de los routers, como MPLS, OSPF ó IS-IS, hará que este valor aumente.

- Usar el sistema operativo Linux en los casos en los que se requieran probar “testbeds” que necesiten estar activos por mucho tiempo, ya que es más estable.

3.2 Instalación y configuración

3.2.1 Instalación y configuración en Windows XP

3.2.1.1 Descargar el archivo de instalación

El primer paso para la instalación es descargar el archivo, GNS3-0.6-win32-all-in-one.exe (ocupa aproximadamente 8MB), que se encuentra en la página web *http://www.gns3.net/download*. El archivo anterior contendrá la versión binaria de los siguientes programas:

Dynamips 0.2.8 – Dynagen 0.11.0: Ambos programas son la base para el funcionamiento de GNS3.

Pemu 0.2.3: Es un emulador de firewalls PIX de Cisco basado en QEMU que no es más que una máquina emuladora y virtualizadora de código libre.

WinPcap 4.0.2: Permite la comunicación de redes virtuales con redes reales, ya que se encarga de detectar las interfaces reales del PC de trabajo para que el simulador pueda asignarlas como extremo de un enlace hacia un router virtual.

3.2.1.2 Instalar GNS3

En esta sección, una vez se haya dado doble clic al archivo que acaba de descargar, los sucesivos cuadros de diálogo lo guiarán durante el proceso de instalación de forma habitual.

La mayoría de los valores que aparecen por defecto son los que aceptaremos en la instalación, a no ser que se desee cambiar el directorio donde se instalará el simulador GNS3. [04]

Los pasos para la instalación son:

1) Dar *doble clic* al archivo de instalación descargado anteriormente. Nos aparecerá una ventana como la que se muestra en la figura 3.5ª. Hacer clic en "*Next*".

2) Aceptar la licencia haciendo clic en el botón "*I Agree*" como se observa en la figura 3.5b.

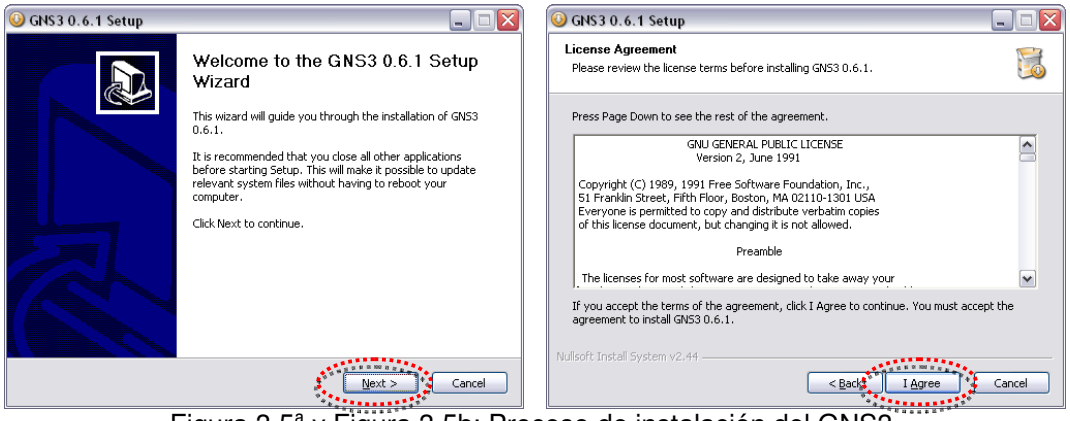


Figura 3.5ª y Figura 3.5b: Proceso de instalación del GNS3.

3) Indicar el nombre del directorio de inicio de GNS3. Seguidamente hacer clic en "*Next*". Ver figura 3.5c.

4) Aceptar todos los componentes que se instalarán por defecto. Hacer clic en "*Next*". Ver figura 3.5d.

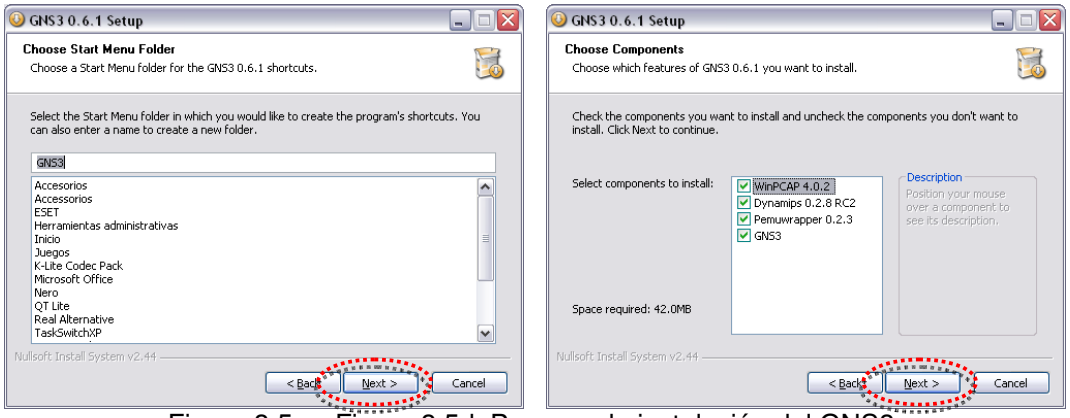


Figura 3.5c y Figura 3.5d: Proceso de instalación del GNS3.

5) En la figura figura 3.5e, Indicar la ubicación del directorio donde se instalará al simulador. Seguidamente hacer clic en "*install*".

6) Antes de concluir la instalación de GNS3, aparecerá la ventana que da inicio a la instalación de WinPcap como se muestra en la figura 3.5f. Hacer clic en "*Next*".

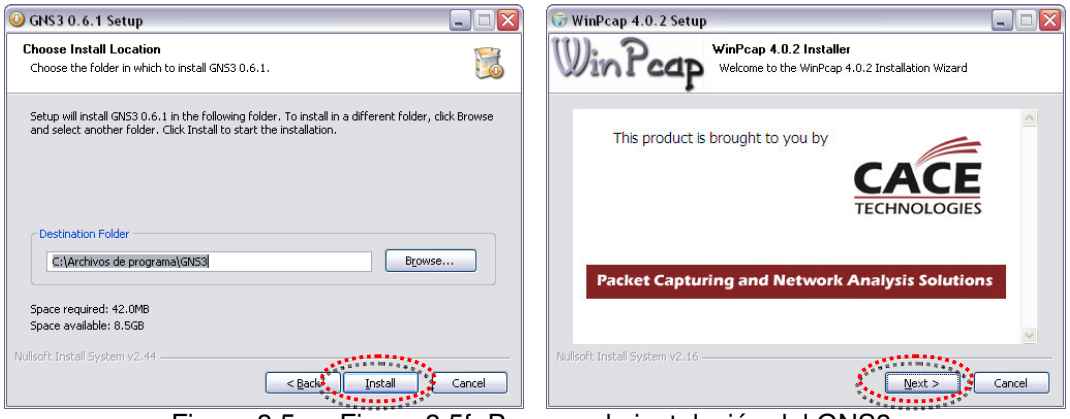


Figura 3.5e y Figura 3.5f: Proceso de instalación del GNS3.

7) Aceptar la licencia de WinPcap haciendo clic en "*I Agree*". Ver figura 3.5g.

8) Después de la instalación de WinPcap, se retomará la instalación de GNS3 y cuando ésta termine aparecerá una ventana como en la figura 3.5h. Hacer clic en "*Finish*" para terminar la instalación.

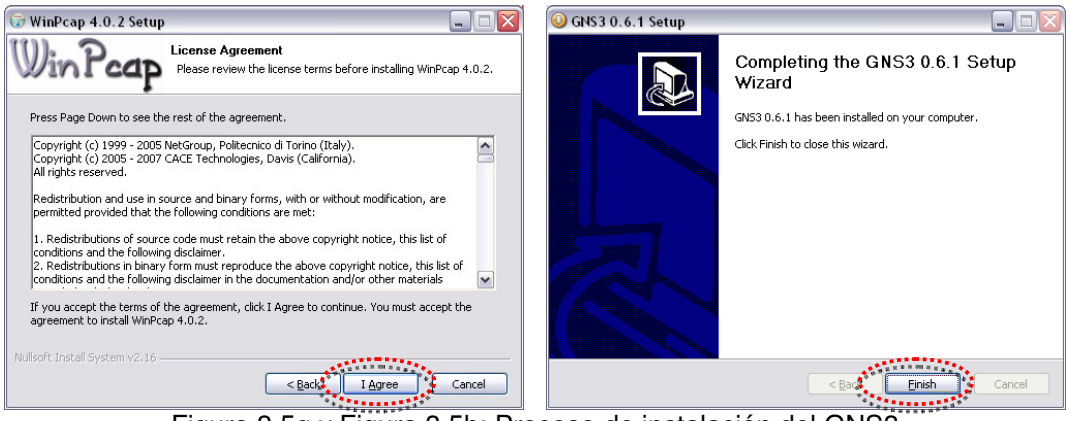


Figura 3.5g y Figura 3.5h: Proceso de instalación del GNS3

9) Tras la finalización de la instalación, podemos ejecutar la aplicación desde "Programas" en el menú de "*Inicio*" o haciendo doble click en el icono correspondiente del escritorio, y aparecerá la pantalla de la figura 3.6.

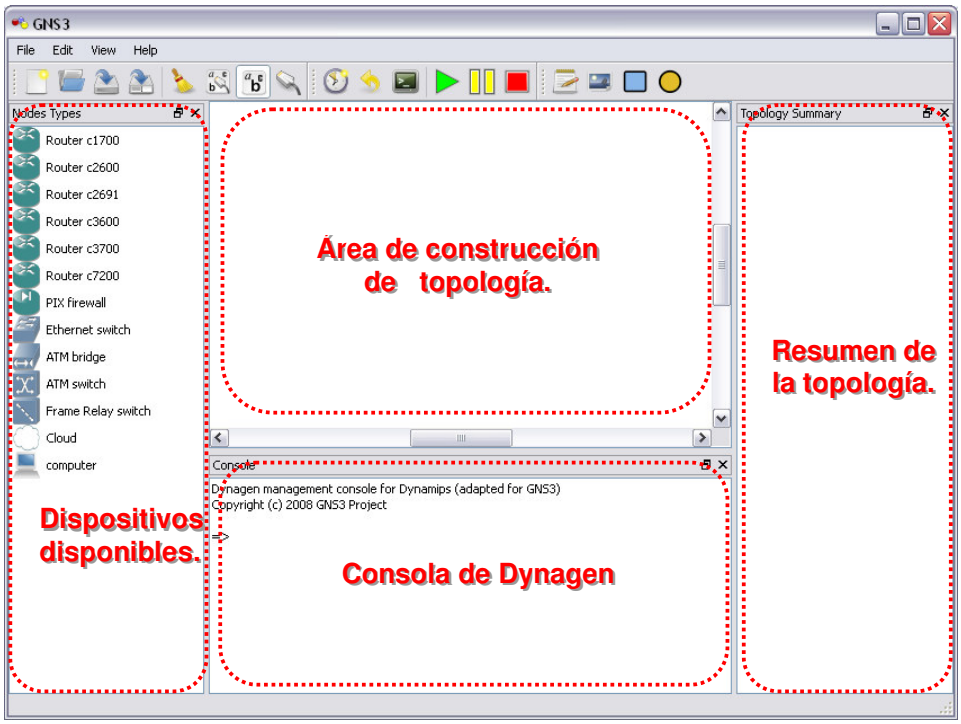


Figura 3.6 Ventana de GNS3 en Windows.

Al momento de ejecutar el programa aparecerá una ventana como la que se muestra en la figura 3.7, la cual podemos obviar ya que su función es reemplazada siguiendo los pasos de los 2 siguientes apartados.

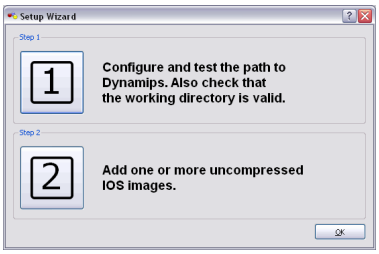


Figura 3.7 Complemento de instalación del GNS3

3.2.1.3 Cargar de los CISCO IOS

El siguiente paso es la carga de la imagen IOS que usarán los routers virtuales de nuestra topología, para lo cual realizaremos los siguientes pasos:

1) Ejecutar la aplicación y seleccionar "*IOS images and hypervisors*" en el menú Edit, como se muestra en la figura 3.8.

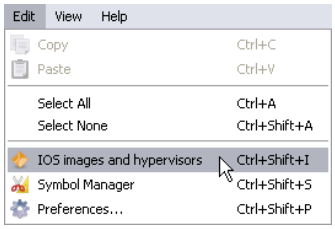


Figura 3.8 Carga de los CISCO IOS en el GNS3

2) En la ventana que aparece, hacer un clic sobre ... para buscar la ubicación de la imagen IOS en el PC. Ver la figura 3.9

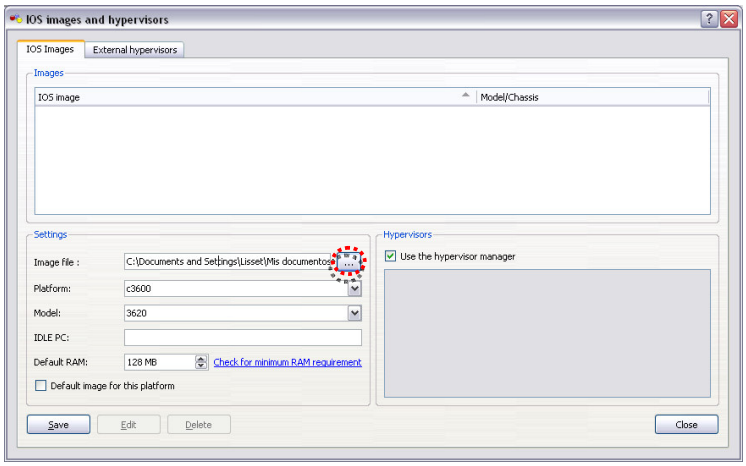


Figura 3.9 Ubicación de CISCO IOS en el GNS3

3) Seguidamente elegiremos la plataforma y el modelo que corresponde con la imagen IOS que usaremos para simular, para ello hacer clic sobre Test Más adelante se mostrará la manera de asignar valores de IDLE PC al router (apartado 3.3.1) y de cambiar los valores de memoria RAM del mismo (apartado 4.4.3.1).

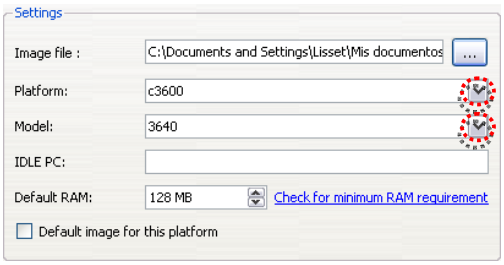


Figura 3.10 Selección de CISCO IOS en el GNS3

4) Finalmente guardamos los cambios haciendo clic en "*Save*". Notar que el valor de IDLE-PC se encuentra vacío por el momento.

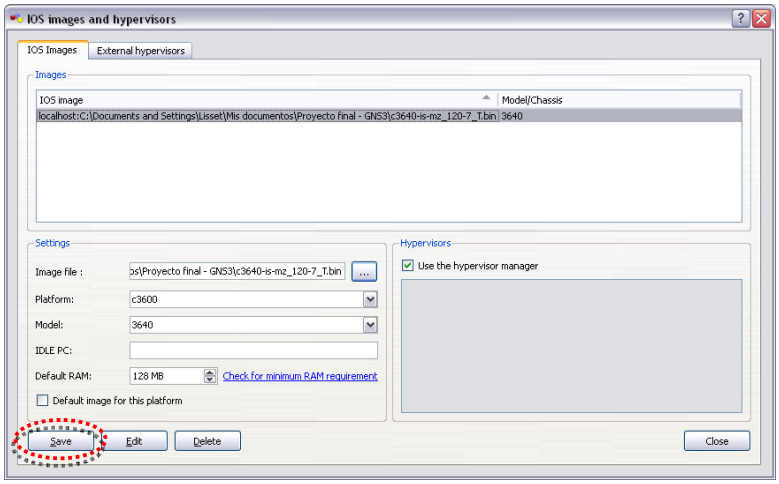


Figura 3.11 Almacenamiento de CISCO IOS en el GNS3

3.2.1.4 Comprobar el *path* hacia Dynamips

Una vez instalado GNS3 es importante comprobar si el simulador ha podido reconocer de forma eficaz el *path* donde se encuentra instalado Dynamips para que pueda usarlo correctamente [04]. Los pasos para realizar esta tarea son los siguientes:

1) En la aplicación, seleccionar la opción "*Preferences*" del menú Edit, como se muestra en la figura 3.12.

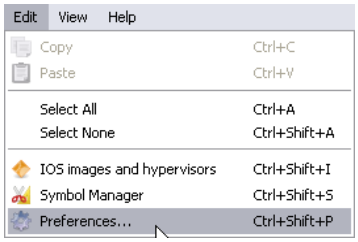


Figura 3.12 Inicio para la comprobación del *path* del Dynamips

2) En la ventana que aparece hacer un clic sobre "*Dynamips*" para obtener la pestaña donde se muestra la ubicación de Dynamips. Comprobar que el *path* que se muestran es correcto haciendo clic en Test si obtenemos algún error buscar la verdadera ubicación Dynamips haciendo clic sobre ... . No olvidar comprobar que se encuentran habilitadas las funciones de Ghostios, Sparsemem y MMAP.

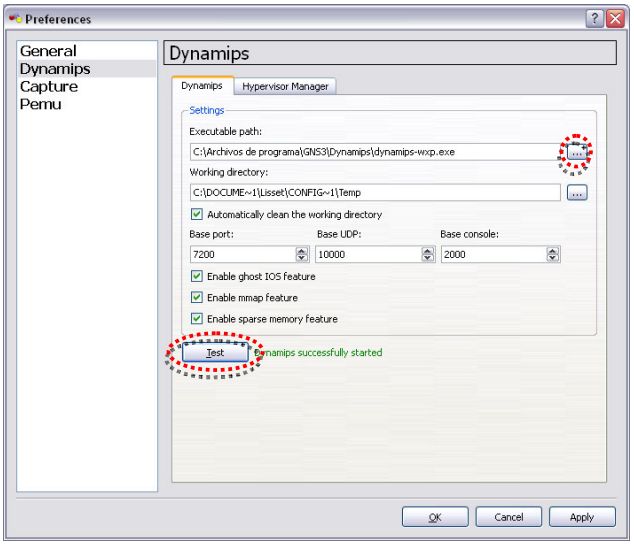


Figura 3.13 Comprobación del *path* del Dynamips

3) Hacer clic en "*Apply*" para guardar los datos.

3.2.2 Instalación y configuración en Linux (Ubuntu 9.4)

3.2.2.1 Descargar el archivo de instalación

La descarga de los archivos necesarios para la instalación en Linux será más fácil, ya que se hará uso del gestor de paquetes APT (**A**dvanced **P**ackaging **T**ool) mediante el comando ***apt-get install gns3***, como se muestra en la figura 3.14. Se puede apreciar como dicho gestor de paquetes se encarga de buscar automáticamente las dependencias que necesita GNS3 para su instalación en Linux. Dichas dependencias, que no son más que programas adicionales de los cuales GNS3 hace uso para poder funcionar, son las siguientes:

- Qt 4.3 (o mayor)
- Sip 4.5 (o mayor)
- PyQt 4.1 (o mayor)
- Python 2.4 (o mayor)

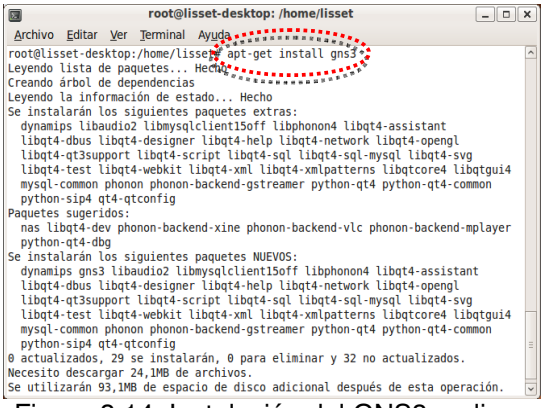


Figura 3.14 Instalación del GNS3 en linux

3.2.2.2 Ejecutar el emulador

Una vez que se ha descargado e instalado todo lo necesario para el funcionamiento de GNS3, se deberá ejecutar y configurar el emulador siguiendo los siguientes pasos:

1) Abrir una ventana de *línea de comando* y escribir el comando que se muestra en la figura 3.15.

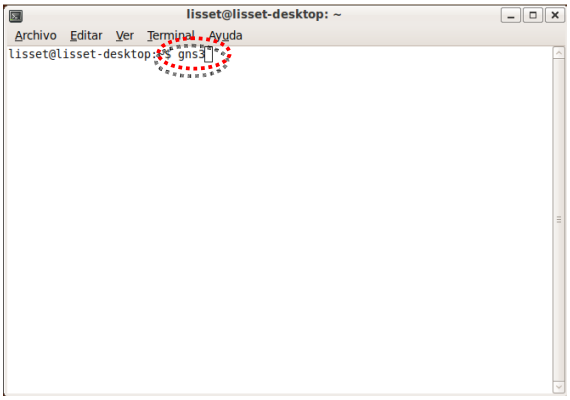


Figura 3.15 Ejecutar el GNS3 en linux

2) Seguidamente nos aparecerá la ventana de trabajo de GNS3 como en la figura 3.16.

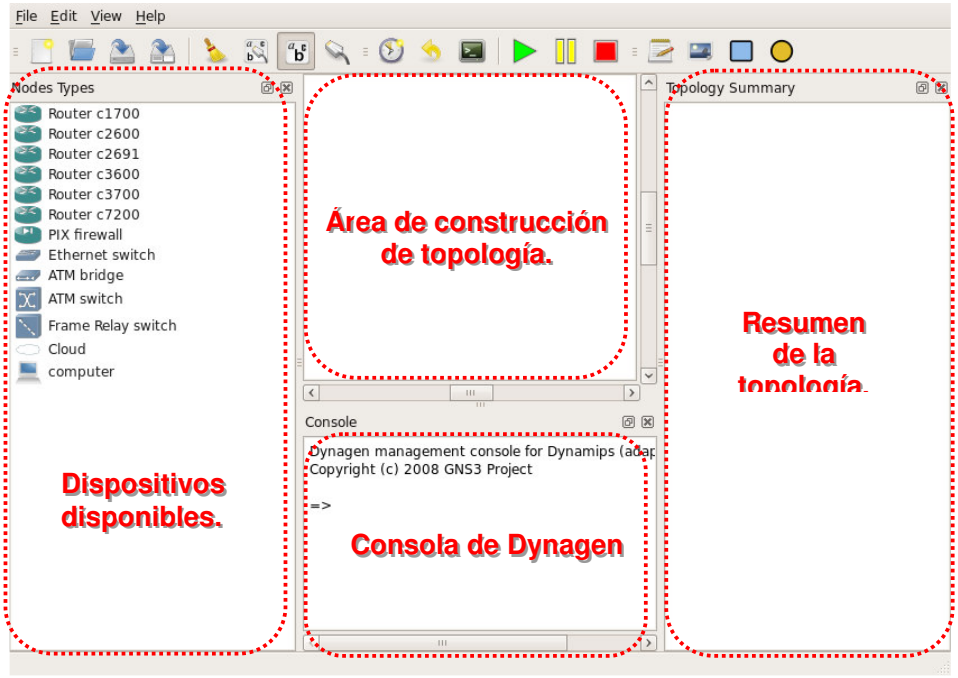


Figura 3.16 Ventana de GNS3 en Linux.

Junto con la ventana de trabajo también aparecerá una pequeña ventana, como la que se muestra seguidamente, la cual podemos ignorar ya que en los pasos siguientes reemplazaremos su función.

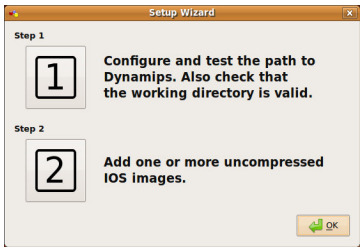


Figura 3.17 Complemento de instalación del GNS3 en linux

3.2.2.3 Cargar la Cisco IOS

En este apartado se seguirán los mismos pasos que en Windows para ubicar el CISCO IOS que será usado en la topología. Al final se obtendrá una ventana como la siguiente:

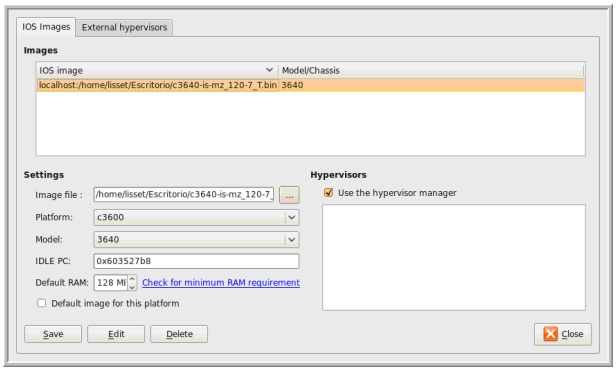


Figura 3.18 Ubicación de CISCO IOS en el GNS3

No olvidar guardar los cambios realizados haciendo un clic en "*Save*".

3.2.2.4 Comprobar el path hacia Dynamips

Comprobamos el path hacia Dynamips de la misma forma que lo hicimos en Windows, de forma que obtendremos una ventana como la que se muestra en la figura 3.19:

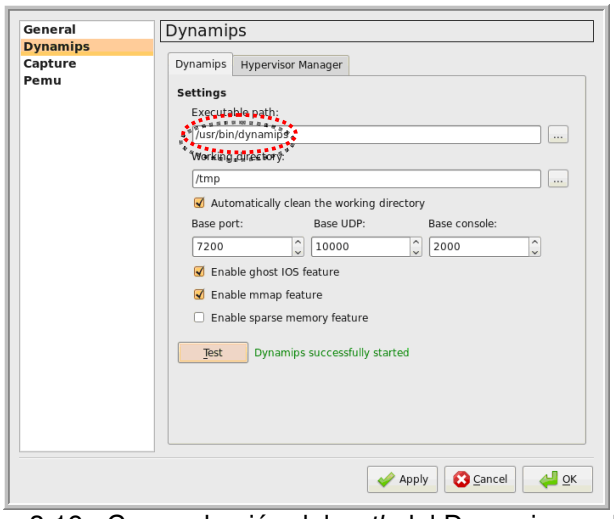


Figura 3.19 Comprobación del *path* del Dynamips en linux

Si la configuración resulto correcta debemos recibir un mensaje de éxito y finalmente guardar los cambios realizados haciendo clic en "*OK*". No olvidar comprobar que se encuentran habilitadas las funciones de Ghostios, Sparsemem y MMAP.

3.3 Uso de GNS3

A continuación se tratará de explicar claramente cómo hacer uso de **GNS 3**, detallando los pasos que hay que seguir para la emulación de las diferentes plataformas CISCO soportadas, así como también se describirán los diferentes métodos existentes para la simulación de PCs, se explicará cómo usar los switches Ethernet, y se detallará

brevemente la utilización de switches Ethernet, ATM y Frame Relay, y firewalls. Asimismo se mostrará como realizar los procesos básicos de GNS3, como por ejemplo, la creación de enlaces, la interconexión de redes reales con virtuales, la captura de datos, la operación en modo hipervisor o el almacenamiento de la topología.

3.3.1 Emulación de Routers CISCO

Como ya se ha indicado anteriormente, para emular routers CISCO reales se necesita una imagen CISCO IOS perteneciente al router que contiene las características que queremos *"clonar"*. En este sentido, el emulador habilitará un número de ranuras o *"slots"* dependiendo del tipo de plataforma que se emula y en cada una de esas ranuras se podrán colocar solo ciertos tipos de adaptadores de interfaces. Por lo tanto, si se quiere añadir *"capacidades de hardware"* en nuestro router virtual, se debe seleccionar el tipo de adaptador de red que éste puede soportan en la configuración del router virtual [04]. Ver figura 3.22 para más detalle.

	Adaptadores de Interfaces Disponibles	
Routers	**Nombre**	**Descripción**
1700s	WIC-1T	1 puerto serie
	WIC-2T	2 puertos serie
	WIC-1ENET	1 puerto Ethernet
2600s	NM-1E	1 puerto Ethernet
	NM-4E	4 puertos Ethernet
	NM-1FE-TX	1 puerto FastEthernet
	NM-16ESW	Módulo de switch Ethernet (16 puertos)
	NM-NAM	Conecta el router virtual a un PC virtual
	NM-IDS	Conecta el router virtual a un PC virtual
	WIC-1T	1 puerto serie
	WIC-2T	2 puertos serie
3600s	NM-1E	1 puerto Ethernet
	NM-4E	4 puertos Ethernet
	NM-1FE-TX	1 puerto FastEthernet
	NM-16ESW	Módulo de switch Ethernet (16 puertos)
	NM-4T	4 puertos serie
	Leopard-2FE	Puerto automático FastEthernet en slot 0
3700s	NM-1FE-TX (FastEthernet, 1 port)	1 puerto FastEthernet
	NM-4T	4 puertos serie
	NM-16ESW	Módulo de switch Ethernet (16 puertos)
	GT96100-FE	2 puertos integrados
	NM-NAM	Conecta el router virtual a un PC virtual
	NM-IDS	Conecta el router virtual a un PC virtual
	WIC-1T	1 puerto sere
	WIC-2T	2 puertos serie
7200s	C7200-IO-FE	Solo puerto FastEthernet en slot 0
	C7200-IO-2FE	2 puertos FastEthernet en slot 0
	C7200-IO-GE-E	Sólo Puerto GigabitEthernet en slot 0
	PA-FE-TX	1 puerto FastEthernet
	PA-2FE-TX	2 puertos FastEthernet
	PA-4E	4 puertos Ethernet
	PA-8E	8 puertos Ethernet
	PA-4T+	4 puertos serie
	PA-8T	8 puertos serie
	PA-A1	Puerto ATM
	PA-POS-OC3	Puerto POS
	PA-GE	1 puerto GigabitEthernet

Tabla 3.1: Lista de adaptadores correspondientes a cada tipo de plataforma. [03]

A continuación se listan todas las plataformas que pueden ser emuladas por **GNS3** ***versión 0.6.1***, tanto si está instalado en Linux como en Windows: [04]

- 1710
- 1720
- 1721
- 1750
- 1751
- 1760
- 2610
- 2610XM
- 2611
- 2611XM
- 2620
- 2620XM
- 2621
- 2621XM
- 2650XM
- 2651XM
- 2691
- 3620
- 3640
- 3660
- 3725
- 3745
- 7200

En la tabla 3.1 podemos notar principalmente que Dynamips no puede emular las interfaces de voz VICs y VWICs. Por lo tanto, podemos decir que GNS3 todavía no es capaz de realizar DSP (**D**igital **S**ignal **P**rossesing).

Los pasos a seguir para la emulación y configuración de un router en GNS3 son los siguientes:

1) Hacer clic en el grupo de plataformas que contiene al router que quedemos emular y arrastrar hasta el área de construcción de topologías. Ver figura 3.20.

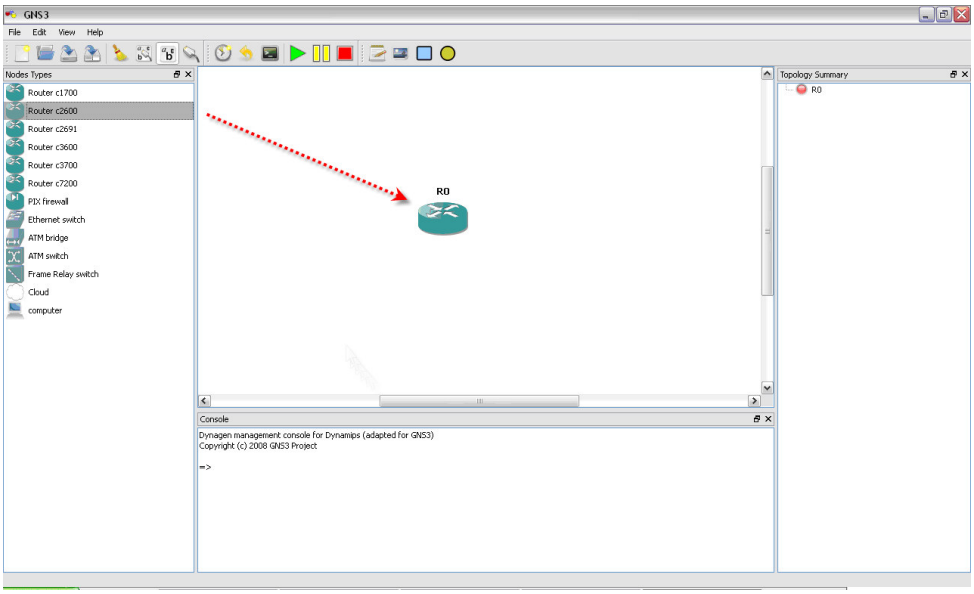


Figura 3.20 Ventana principal de GNS3.

2) Hacer un clic derecho sobre el router y después elegir *"configure".*

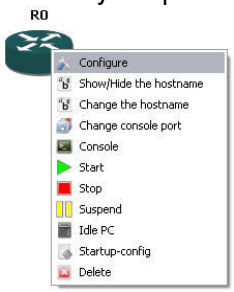


Figura 3.21 Menú de opciones de un router.

3) Hacer clic en el nombre del router, después en la pestaña *"slots"* y finalmente elegir las interfaces que se desean haciendo clic en de cada slot. Guardar los cambios cliqueando en *"OK"*. Ver figura 3.22.

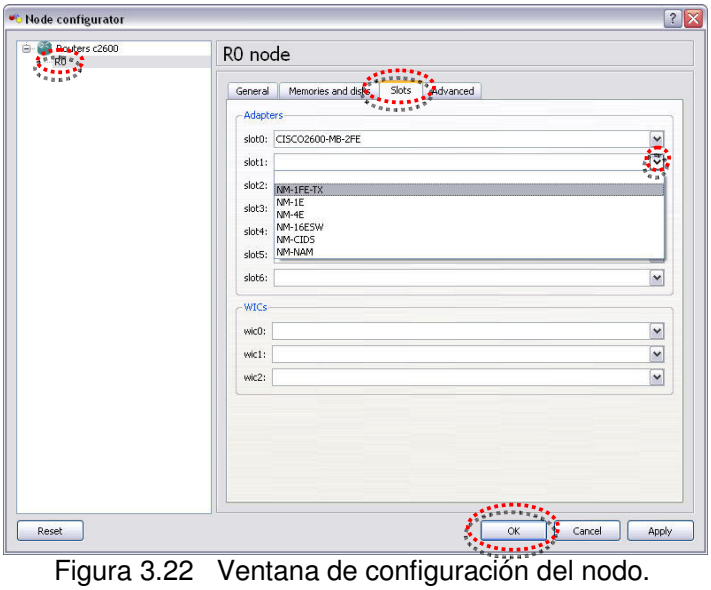


Figura 3.22 Ventana de configuración del nodo.

4) Encender el router con un clic derecho y eligiendo *"start"*. Notar que ahora en el resumen de la topología el indicador del router se encuentra de color verde.

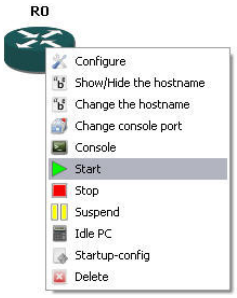


Figura 3.23 Menú de opciones de un router.

5) Calcular el valor de IDLEPC para la imagen de IOS utilizada con un clic derecho en el router y eligiendo *"Idle PC"* del menu desplegado.

Figura 3.24 Menú de opciones de un router.

6) Tras unos instantes se apreciará una ventana como la figura 3.25, hacer clic en para ver todos los posibles valores de IDLE PC calculados, los mejores son los que tiene un * a la izquierda, elegir uno de ellos y hacer clic en *"apply"*.

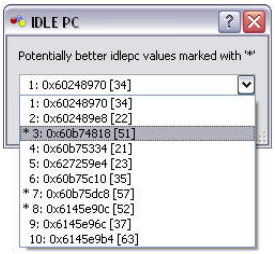


Figura 3.25 Ventana de IDLE PC.

7) Finamente hacer clic derecho sobre el router y elegir *"console"* para obtener una ventana de Telnet con la consola del router que hemos emulado.

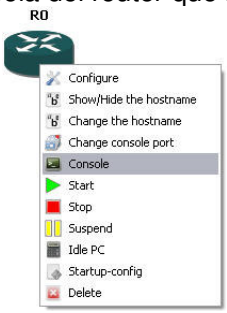


Figura 3.26 Menú de opciones de un router.

3.3.2 Emulación de Switches Ethernet

GNS3 posee integrada la capacidad de emulación de simples switches Ethernet con funcionalidades básicas como la creación de Vlans o el funcionamiento del trunking 802.1q. Por defecto, un switch emulado con GNS3 tiene 8 puertos access configurados en la Vlan1 pero se puede añadir hasta 10.000 puertos, pudiendo ser cada uno de ellos, un puerto de acceso o uno troncal. [04]

En este sentido, si se desea trabajar con switches que poseen más funcionalidades, GNS3 puede emular una tarjeta *"EtherSwitch"* que pude ser soportada sólo por determinadas plataformas CISCO. La tarjeta *"EtherSwitch"* que emula Dynamips es *"NM-16ESW"* y, como se puede apreciar en la tabla 3.1, puede ser incluida en casi todas las plataformas disponibles en GNS3. Las principales capacidades de esta tarjeta son:

- Interfaces Ethernet Layer 2
- Interfaces Switch Virtual (SVI)
- Protocolo VLAN Trunk
- EtherChannel
- Protocolo Spanning Tree
- Protocolo Cisco Discovery
- Switched Port Analyzer (SPAN)
- Calidad de Servicio
- IP Multicast
- Storm-Control
- Seguridad de Puertos
- Stacking
- Control de flujo.

Cabe resaltar que una tarjeta *"NM-16ESW"* no puede soportar todos los comandos existentes en un switch Ethernet, con lo cual, para su uso se recomienda consultar los manuales existentes en CISCO.

La emulación y configuración de un switch Ethernet usando GNS3 se hace de la siguiente manera:

1) Hacer clic en “Ethernet Switch” situado en la parte izquierda de la ventana principal y arrastrar hasta el área de construcción de topologías.

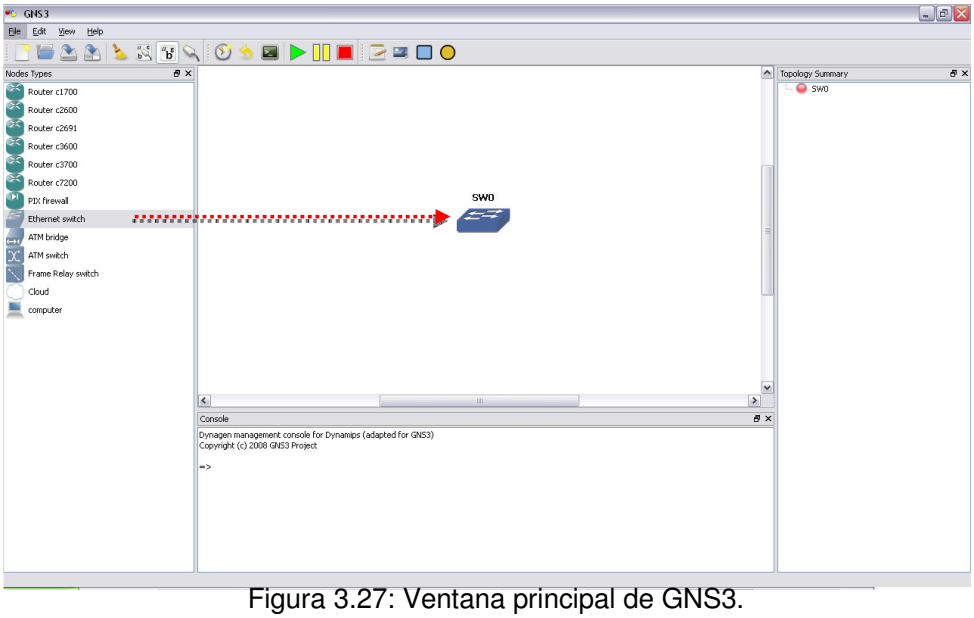


Figura 3.27: Ventana principal de GNS3.

2) Hacer clic derecho sobre el switch y elegir *“configure”* .

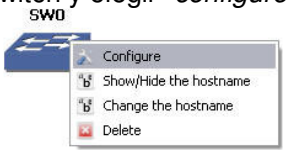


Figura 3.28 Menú de opciones de un switch.

3) En ventana de la figura 3.29, hacer clic sobre el nombre del dispositivo y borrar la configuración inicial del switch Ethernet seleccionando cada puerto y haciendo clic en *“Delete”.*

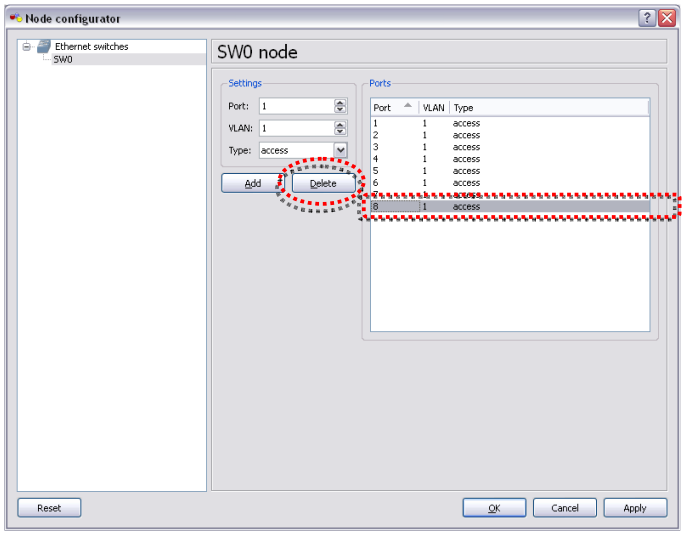


Figura 3.29 Ventana de configuración del nodo.

4) Configurar los nuevos puertos ingresando los parámetros de la sección *"Settings*" y hacer clic en *"Add"*. Finalmente guardar los cambios haciendo clic en *"OK"*.

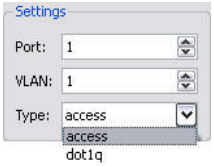


Figura 3.30 Sección "settings" de la ventana de configuración del nodo.

5) Crear enlaces entre los diferentes dispositivos disponibles y cualquiera de las interfaces del switch que acabamos de añadir.

3.3.3 Emulación de Dispositivos ATM y Frame Relay

GNS3 posee la capacidad de emular simples Switches ATM y Frame Relay [04], dichos dispositivos se configuran y se añaden a la topología de forma similar a la que se lleva a cabo en los switches Ethernet convencionales, es decir, arrastrando su imagen al área de creación de topología. No pueden ser encendidos ni apagados, es decir, permanecen activos desde que se crea un enlace a alguno de sus puertos. En cuanto a su configuración podemos decir que debemos indicar los parámetros que se piden en las ventanas de configuración respectivas a cada tipo de switch. No entraremos en más detalle con estos dispositivos puesto que no está en nuestros objetivos profundizar en dispositivos que funcionen con otras tecnologías que no sean la Ethernet.

3.3.4 Simulación de PCs

GNS3 permite, además de los equipos de red, la incorporación de PCs en las topologías creadas, lo que facilita la comprobación y el estudio de las redes simuladas. Las formas existentes de simulación de PCs en GNS3 son las siguientes: [04]

3.3.4.1 Usando Virtual PC

Esta primera forma de simulación de PCs se realiza utilizando un programa llamado *"Virtual Pc Simulator"* ó VPC que usa puertos UDP para la comunicación entre el simulador y cada uno de los PCs simulados. VPC es un programa que corre tanto en Windows como en Linux y que se puede descargar desde Internet de forma gratuita. Las ventajas de usar VPC es que su uso es simple y que no usa grandes cantidades de memoria ni siglos de CPU para su funcionamiento; por otro lado, tiene la desventaja de que tiene funcionalidad limitada, ya que solo permite el uso de comandos como *"ping"* y *"traceroute"*, además soporta un máximo de nueve (09) PCs simulados simultáneamente.

Los pasos a seguir para la simulación de PCs utilizando este método son:

1) Descargar e instalar el programa *"Virtual Pc Simulator"* desde http://wiki.freecode.com.cn/doku.php?id=wiki:vpcs.

2) Escribir el carácter **?** para observar los comandos disponibles, como en la figura 3.31 y *"show"* para ver la configuración de red actual de los PCs simulados. Para cambiar de PC basta con escribir un número del 1 al 9 asignado a cada una de ellas.

```
VPCS 1 >?
show                          Print the net configuration of PCs.
d                             Switch to the PC[d], d is digit, range 1 to 9.
hist                          List the history command, use arrow keys to get
                              recently-executed commands.
ip address CIDR [gateway]     Set the host's ip, gateway's ip and network mask.
                              In the ether mode, the ip of the tapx is the
                              maximum host ID of the subnet. Default IPv4 CIDR
                              is 24, IPv6 is 64. 'ip 10.1.1.70 10.1.1.65 26',
                              set the host ip to 10.1.1.70, the gateway ip to
                              10.1.1.65, the netmask to 255.255.255.192, the
                              tapx ip to 10.1.1.126 (ether mode).
dhcp                          Configure host/gateway address using DHCP.
arp                           Show arp table.
ping address                  Ping the network host.
echo address port [udp|tcp]   Echo the host port with protocol(udp/tcp).
tracert address [maxhops]     Print the route packets take to network host.
                              default maxhops is 16.
conf [lport|rport] port       Set local or remote port. 'conf lport' will close
                              the old port and open the new port. Only udp mode.
?                             Print help.
quit                          Close all the port(udp mode) or the tapx (ether
                              mode), then quit.
```

Figura 3.31 Comandos disponibles en vpcs.

3) Configurar los datos de cada PC ingresando el comando *"ip"* seguido de la dirección IP (192.168.0.2), la puerta de enlace por defecto del PC (192.168.0.1) y la mascara de subred (24). Escribir un número y luego presionar *"enter"* para cambiar de PC.

```
VPCS 1 >ip 192.168.0.2 192.168.0.1 24
PC1 : 192.168.0.2 255.255.255.0 gateway 192.168.0.1

VPCS 1 >2
VPCS 2 >
```

Figura 3.32 Configuración de PCs virtuales.

4) Arrastrar tantas nubes como PCs se quiera integrar al simulador, hacer clic derecho en cada uno de ellos y elegir *"configure*". Hacer clic en C0 debajo de *"clouds"* y elegir la pestaña llamada *"NIO UDP".* Ver figura 3.33.

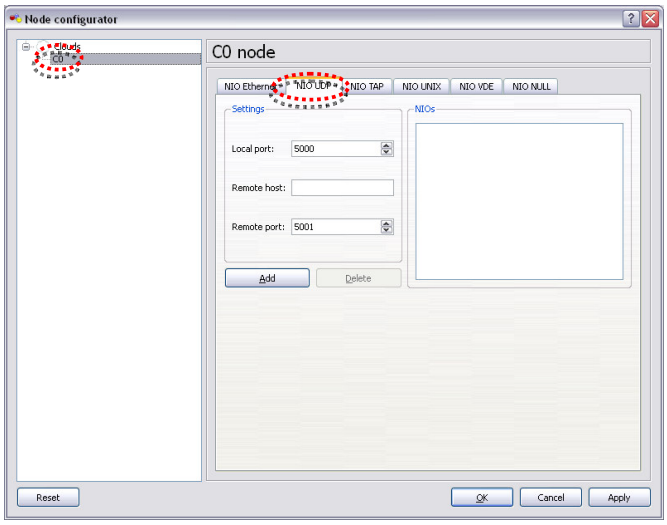


Figura 3.33 Ventana de configuración de nodo.

5) Configurar los parámetros *"Local port"*, *"Remote host"* y *"Remote port"* debajo de *"Settings"* con los datos mostrados en la figura 3.34 que corresponden a los puertos asignados por VPC para un PC virtual. Hacer clic en *"Add"* y finalmente en *"OK"*.

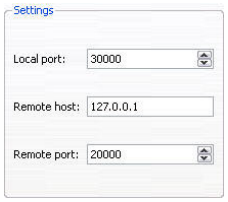
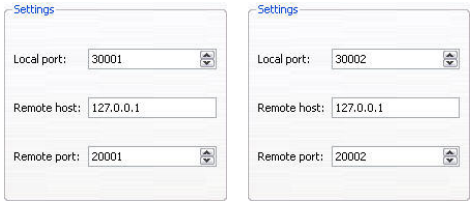


Figura 3.34 Configuración de PCs virtuales.

6) Repetir los pasos 4 y 5 en cada una de las otras nubes añadidas pero asignar valores correlativos a los mostrados anteriormente, tanto para el puerto local como para el remoto.



… ETC.

Figura 3.35 Configuración de PCs virtuales.

7) Añadir enlaces entre PCs y otros dispositivos con normalidad.

3.3.4.2 Routers que actúan como PC

En este método, la incorporación de un PC a la topología es la más simple, ya que solo se trata de añadir un router y configurarlo de forma que realice funciones limitadas propias de un PC, la desventaja de este método es que utiliza demasiados recursos de memoria y procesamiento.

Los comandos que debemos usar para configurar cada router básicamente asignarán una dirección IP y un Gateway a una interfaz del mismo, en nuestro ejemplo, se usará la interfaz FastEthernet 0/0, además se deshabilitarán las funciones de enrutamiento del dispositivo. Ver figura 3.36

```
Router(config)#no ip routing
Router(config)#interface fa0/0
Router(config-if)#ip address 192.168.0.2 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
*Mar  1 00:09:47.963: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state t
o up
*Mar  1 00:09:48.963: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
et0/0, changed state to up
Router(config)#ip default-gateway 192.168.0.1
```

Figura 3.36 Configuración de router como PC.

3.3.4.3 Usando un PC real

En este caso usaremos la capacidad que tiene el simulador de poder conectar dispositivos reales a una topología simulada. Debemos arrastrar una nube, asignarle cualquier adaptador real de red, detectado en el host donde se encuentra el simulador, crear un enlace virtual que la conecte con el resto de la topología y finalmente otro físico entre el simulador y el PC externo. Estos pasos se explicarán con más detalle en el apartado 3.3.7.

3.3.5 Emulación de Firewall PIX

GNS3 está diseñado para poder soportar la emulación de Firewalls PIX de CISCO pero, al igual que con la emulación de routers, se necesita de una imagen CISCO PIX, además si se desea trabajar con capacidades adicionales es necesario contar con números de serie y códigos de activación específicos.

Estos dispositivos poseen interfaces Ethernet y no pueden conectarse con equipos reales directamente a través de una nube, sino que tiene que hacerlo teniendo en medio un switch emulado. Por otro lado, estos dispositivos no cuentan con parámetros “idlepc” disponibles, por lo tanto, cada firewall emulado consumirá el 100% de la capacidad del CPU, para solucionar este problema existen aplicaciones tanto en Linux como en Windows que se encargan de limitar el uso de CPU a determinados

procesos, estos programas son: BES y cpulimit para Windows y Linux respectivamente. [04]

Nosotros tampoco nos centraremos en el estudio de este tipo de dispositivos ya que se trata de uno generalmente sólo estudiado en exámenes de certificación de CCNP y CCIE, con lo cual, no es muy usado en redes reales, además de que su emulación requiere de muchas licencias adicionales.

3.3.6 Creación de Enlaces

En GNS3 los enlaces son usados para unir dos dispositivos de red de la topología creada, pueden ser GigaEthernet, ATM, POS, FastEthernet, Ethernet o Serie y su tipo depende de los adaptadores de interfaz que tengan los dos dispositivos a los que une en sus extremos. [04]

Para simular un enlace se siguen los siguientes pasos:

1) Hacer clic en situado en la parte superior de la ventana principal de GNS3 y elegir *"Manual"* para que se asigne un tipo de enlace automáticamente.

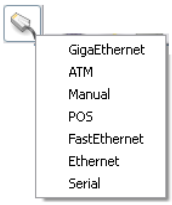


Figura 3.37 Tipos de enlaces disponibles.

2) Hacer clic en el router, aparecerá un cuadro con todas las interfaces disponibles (indicador en rojo), elegir una de ellas y repetir el proceso para el router del otro extremo del enlace. Ver figura 3.38.

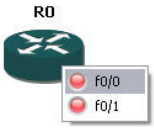


Figura 3.38 Interfaces disponibles en el router.

3) Cuando el enlace se ha terminado, hacer clic sobre para deshabilitar el proceso de creación de enlaces.

3.3.7 Creación de enlaces a equipos reales

Hemos hablado anteriormente, uno de los principales logros de Dynamips esta relacionado con la comunicación de redes virtuales con el mundo real, dicha proeza se logra por medio de enlaces que tienen la capacidad de comunicar interfaces de los routers virtuales con las interfaces de red reales del simulador, de modo que, se puede agregar a la topología, tantos equipos reales como adaptadores de red tenga el simulador. [04] Por lo tanto, los paquetes que salen de un router virtual son colocados en red real a través del correspondiente adaptador de red del emulador asignado y los paquetes que entran son enviados de vuelta al router virtual por el mismo adaptador.

En GNS3 los enlaces descritos anteriormente son simulados siguiendo los siguientes pasos:

1) Hacer clic en la nube y arrastrarla hasta el área de construcción de topologías.

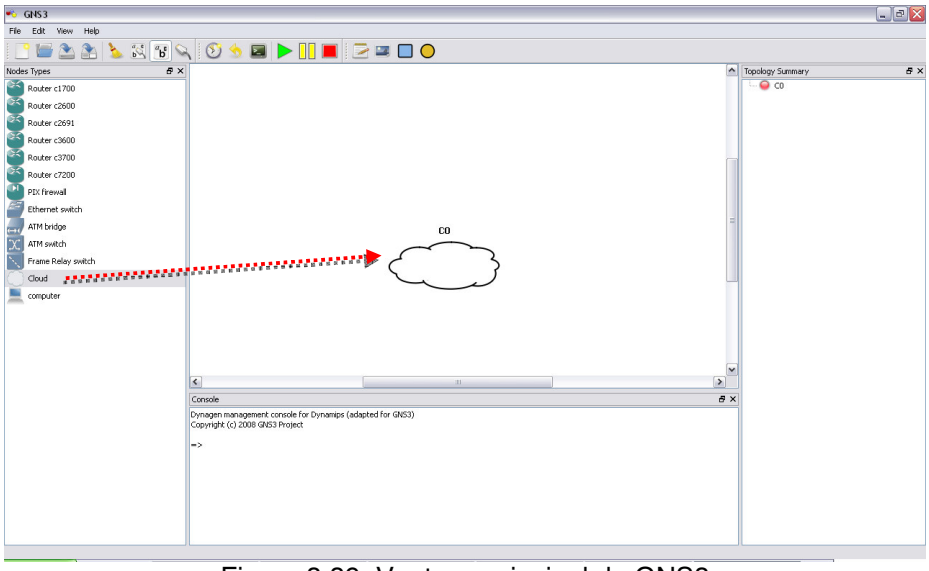


Figura 3.39 Ventana principal de GNS3

2) Hacer clic derecho sobre la nube y elegir *"configure"* para abrir una ventana de configuración.

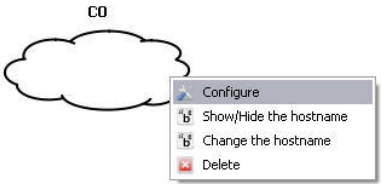


Figura 3.40 Menú de opciones de una nube.

3) Hacer clic en el nombre de la nube, elegir la pestaña *"NIO Ethernet"* y hacer clic en la caja que esta inmediatamente debajo de *"Generic Ethernet NIO"* para observar todos los adaptadores de red reconocidos por el simulador. Seleccionar el que queremos usar para conectar el dispositivo externo y añadirlo a la topología haciendo clic en *"Add"*, finalmente aceptar los cambios con un clic en *"OK"*. Ver figura 3.41.

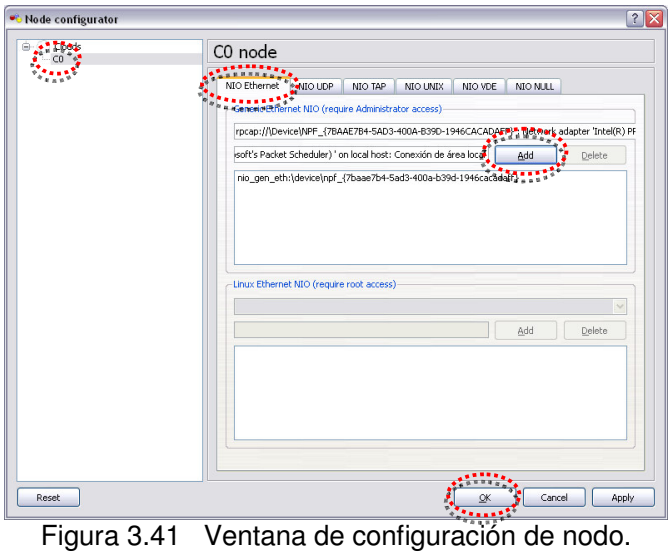


Figura 3.41 Ventana de configuración de nodo.

4) En el otro extremo externo del enlace, configurar la dirección IP necesaria para la conexión. Si el equipo externo es un PC configurar una IP en su adaptador de red que se va a usar, por otro lado, si se trata de un router, configurar la dirección IP desde la consola del equipo en la interfaz que se conectará al simulador. No olvidar elegir direcciones IP que mantengan concordancia con la topología a simular.

5) Realizar una conexión física entre el adaptador de red que ha sido elegido anteriormente en el simulador y el correspondiente al equipo externo. Ver figura 2.5.

6) Crear un enlace virtual, como se explicó en el apartado 3.3.6, entre un router virtual y la nube creada.

3.3.8 Almacenamiento de topología

GNS3 permite el almacenamiento de la topología y de la configuración efectuada en los routers emulados en archivos diferentes; la topología se guarda, en forma de texto, en un archivo .net que es interpretado por Dynagen y mostrado gráficamente por GNS3; mientras que la configuración de los routers emulados se guardan en archivos .cfg, los cuales abiertos con un editor de texto, muestran los comandos realizados en el router de la misma forma que se pueden encontrar en un router real. [04]

Figura 3.42: Archivos con la topología y la configuración creados por GNS3.

El almacenamiento se realiza de la siguiente forma:

1) Al iniciar el programa aparecerá una ventana la figura 3.43, en ella, habilitar el almacenamiento de las NVRAMs y archivos adicionales (logfiles y bootfiles) y de los archivos de configuración de todos los routers, además asignar un nombre al proyecto. Hacer clic en *"OK"*.

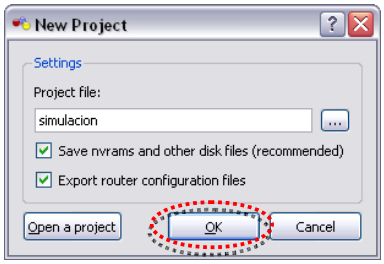


Figura 3.43: Ventana de nuevo proyecto.

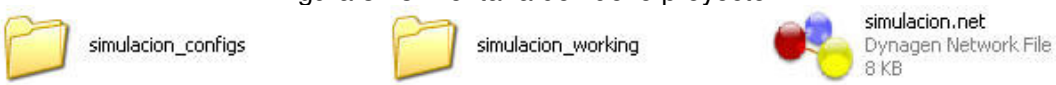


Figura 3.44: Archivos creados después de aceptar la creación de un proyecto.

3) Arrastrar un router al área de creación de topologías, hacer clic derecho sobre él y elegir *"Change the hostname"*, a continuación escribir el nuevo nombre y seleccionar *"OK"*.

Figura 3.45: Ventana de cambio de nombre de host.

4) Guardar los cambios de configuración realizados en el router usando el comando **wr** para almacenar los datos que se encuentran en la RAM a la NVRAM.

```
Router(config)#hostname Barcelona
Barcelona(config)#exit
Barcelona#
*Mar  1 00:01:14.343: %SYS-5-CONFIG_I: Configured from console by console
Barcelona#wr
Building configuration...
[OK]
```

Figura 3.46: Comandos para guardar la configuración de routers.

5) Hacer clic sobre de la barra principal de GNS3 y aparecerá una ventana como en la figura 3.47, elegir *"Extracting to a directory"* para guardar los archivos de configuración de todos los routers existentes en la topología. Hacer clic en "Ok", buscar la ubicación de la carpeta simulación_config y aceptar el destino.

Figura 3.47: Ventana de almacenamiento configuraciones.

Cabe resaltar que en la figura 3.47 existe la opción *"Importing from a directory"* que nos permite recuperar las configuraciones de los routers, así como también, importar configuraciones de routers de otras topologías a la nuestra siempre y cuando ambos routers tengan el mismo *Hostname*.

6) Seguidamente podemos comprobar que en la carpeta simulación_config se ha creado un archivo llamado Barcelona.cfg que contiene la configuración del router. No olvidar escribir el comando **wr** antes de volver a extraer la configuración.

7) Para guardar la topología dibujada, conexiones y el escenario hacer clic en *"File"* y elegir *"Save"*.

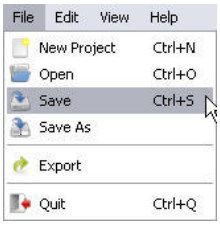


Figura 3.48: Opciones de archivo.

3.3.9 Captura de datos

GNS3 integra la capacidad de capturar los paquetes que pasan por interfaces Ethernet o Serie y almacenarlos en archivos con formato libpcap para que puedan ser interpretados por aplicaciones como Wireshark, tcpdump, etc. [03][04]

Para realizar la captura de datos se deben seguir los siguientes pasos:

1) Descargar e instalar el programa Wireshark de la siguiente URL: http://www.wireshark.org en esta página se puede encontrar la versión tanto para Windows como para Linux.

2) En el menú principal ir a *"Edit"* y seleccionar "*Preferences*".

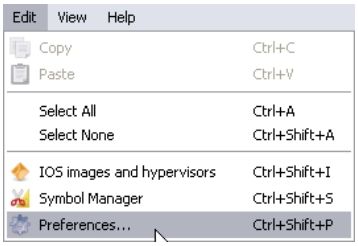


Figura 3.49: Opciones de edit.

3) Hacer un clic sobre "*Capture*" para obtener la pestaña relacionada con la configuración del proceso de captura del simulador. Comprobar que el *path* indicado en *settings* es el correcto para el programa de captura que se va a usar, en este caso se trata de Wireshark. Para terminar hacer clic en "OK". Ver figura 3.50.

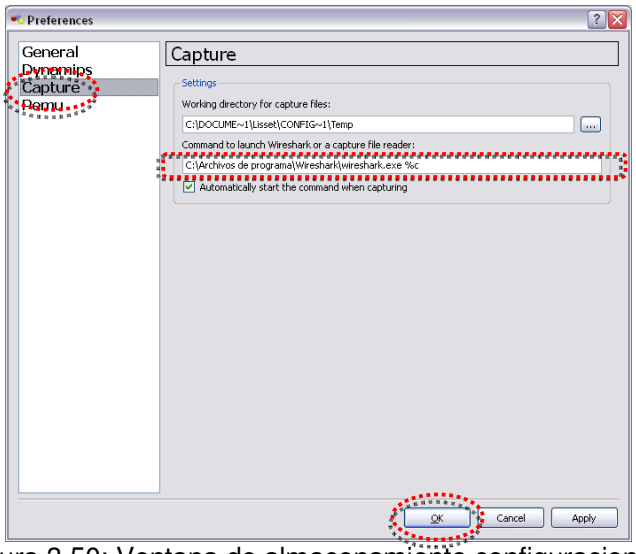


Figura 3.50: Ventana de almacenamiento configuraciones.

4) Hacer clic derecho sobre cualquier parte del enlace del que se desea obtener una captura y seleccionar *"Capture"*. Los enlaces pueden ser Ethernet o serie y pueden unir dos equipos Ethernet o uno Ethernet y otro FrameRelay. Ver figuras 3.51a, 3.51b y 3.51c.

Figura 3.51 a: Enlace Ethernet entre routers.

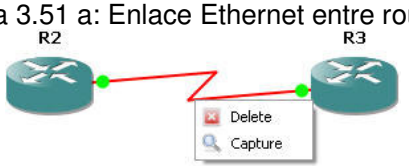


Figura 3.51 b: Enlace Serie entre routers.

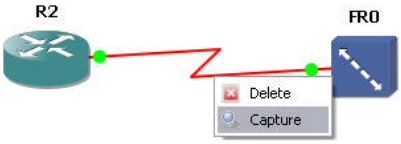


Figura 3.51 c: Enlace Serie entre router y switch FrameRelay.

5) A continuación aparecerá una ventana de captura. Hacer clic en y elegir el router que actuará como fuente de los paquetes enviados y la encapsulación que realiza dicho router. Hacer clic en *"OK"* para finalizar.

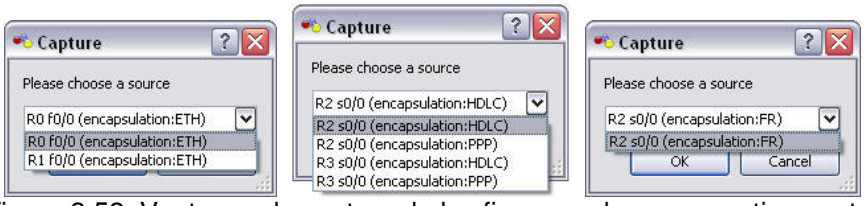


Figura 3.52: Ventanas de captura de las figuras a, b y c respectivamente.

Como se puede apreciar en las figuras anteriores, en el caso de un enlace Ethernet la encapsulación siempre será Ethernet mientras que si el enlace es serie debemos elegir entre *HDLC, PPP o FR*.

6) Al cabo de unos segundos tendremos una ventana de Wireshark con los datos de la captura realizada.

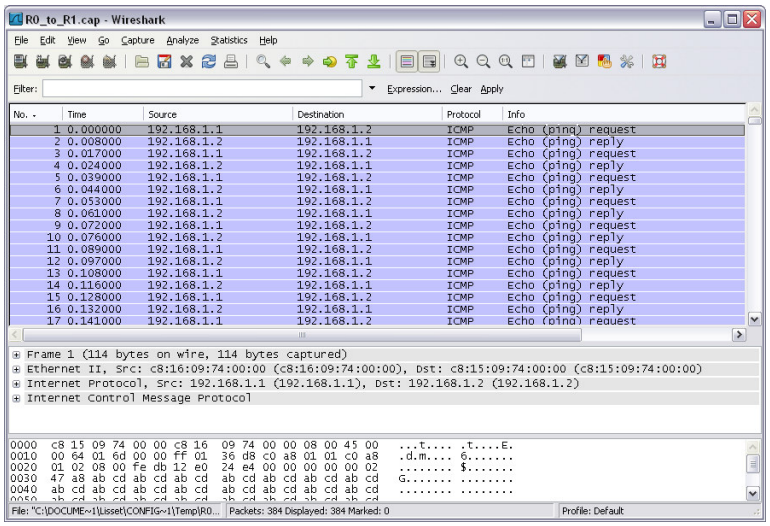


Figura 3.53: Captura de tráfico.

3.3.10 Operación en modo Hipervisor

GNS3 posee la capacidad de permitir que un host cliente con Dynagen se comunique con otro host externo que contiene al emulador Dynamips por medio de conexiones TCP/IP, es decir, el host con Dynagen escuchará al emulador por un puerto TCP determinado. A este modo de trabajo del simulador se le conoce como Hipervisor y tiene como principal ventaja que permite la repartición del uso de recursos de procesamiento y memoria en más de un host de la red, con lo cual, es posible simular topologías de gran tamaño. [04]

Cabe resaltar que los equipos externos que contendrán al emulador Dynamips (*External Hipervisor*) pueden trabajar tanto en Windows como en Linux, lo único que hay que tener en cuenta es que se use correctamente la nomenclatura para la definición de los *path* hacia los directorios de trabajo, es decir, los directorios donde se guardarán los archivos generados.

1) En la ventana principal del emulador, elegir *"Edit"* y después hacer clic en *"IOS images and hipervisor"*.

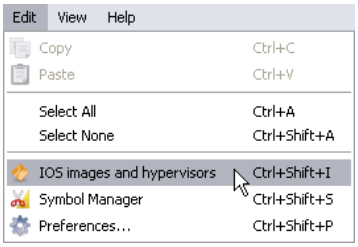


Figura 3.54: Opciones de edit.

2) Elegir la pestaña *"External hipervisors"* y añadir la dirección IP, el puerto TCP, los puertos Base UDP y el de consola para la comunicación con el host remoto. Por otro lado, crear la carpeta de trabajo en el emulador Dynamips y colocar dicho path en la configuración. Finalizar haciendo clic en *"Save"* y después en *"Close."*

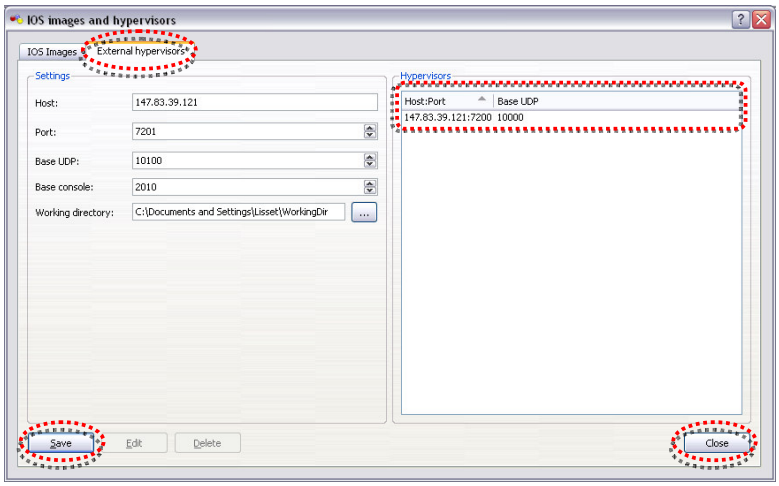


Figura 3.55: Ventana de configuración de Hipervidor.

En la figura anterior vemos como se indica a nuestro host con GNS3 (*host cliente*) que se comunique con el *External hipervisor* de dirección IP 147.83.39.121 por medio del puerto 7200 aunque dicho valor de puerto puede ser modificado. También se indica el *path* hacia la carpeta del Hipervisor que se usará para guardar sus datos propios de la simulación. Para la elección de los puertos UDP, GNS3 asigna por defecto 100 puertos para cada host remoto a partir del 10000, éstos puertos son usados para realizar las conexiones entre nodos de nuestra topología, es decir, habrá uno por cada enlace creado. Finalmente el puerto de consola puede ser elegido de forma arbitraria y es usado cuando se abre una ventana de consola en un router.

3) Seleccionar la pestaña "IOS Images" para unir los host remotos con una imagen de CISCO IOS correspondiente.

4) Copiar la imagen a simular en el host remoto.

5) Escribir la ubicación de la IOS en la sección *"Image file"* junto con sus demás parámetros. Seguidamente deshabilitar la opción *"Use the Hipervisor manager"* y elegir un host de la lista. Finalizar deshabilitando *"Default image for this plataform"* y haciendo clic en *"Save"* y después en *"Close"*. Ver imagen 3.56.

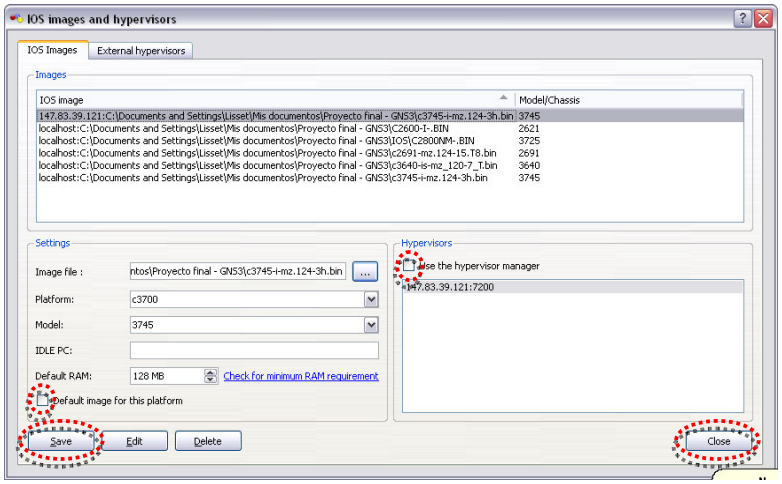


Figura 3.56: Creación de un Hipervisor.

Lo que hemos logrado hasta ahora es que un router con una determinada IOS se emule en un PC externo. También es posible conseguir que el consumo de memoria que produzca un router se reparta entre varios PC externos simultáneamente dependiendo del nivel de dicho consumo en cada hipervisor externo, provocando un balanceo de carga, esto se realiza haciendo clic sobre más de un hipervisor de la lista presionando la tecla *Ctrl* simultáneamente.

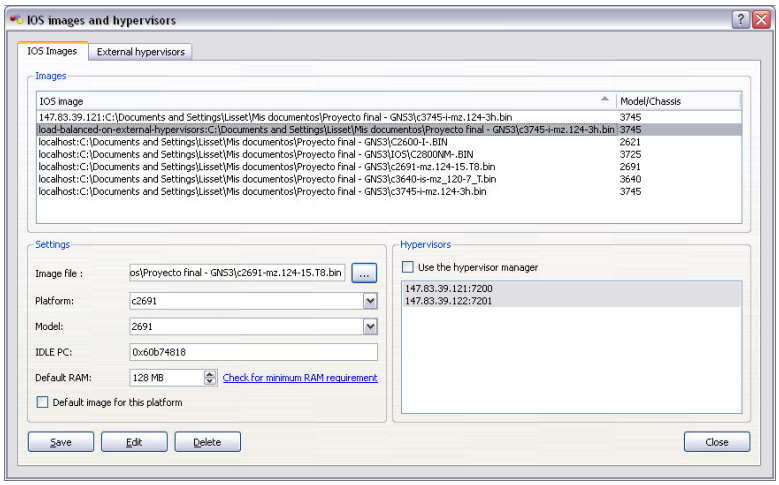


Figura 3.57: Balanceo de carga usando hipervisores.

6) Activar Dynamips en el host remoto desde GNS3 o desde la ventana de comandos. Como se dijo, por defecto el host escuchará el puerto 7200, si se usan varios Hipervisores, es necesario cambiar este puerto para cada uno de ellos.

```
dynamips -H 7201 &
```

Figura 3.58a: Comando para activar Dynamips desde Linux.

```
C:\Archivos de programa\GNS3\Dynamips>dynamips-wxp -H 7201
```

Figura 3.58b: Comando para activar Dynamips desde Windows.

7) Esta vez cuando arrastremos un router, el emulador nos pedirá elegir qué imagen nos gustaría usar.

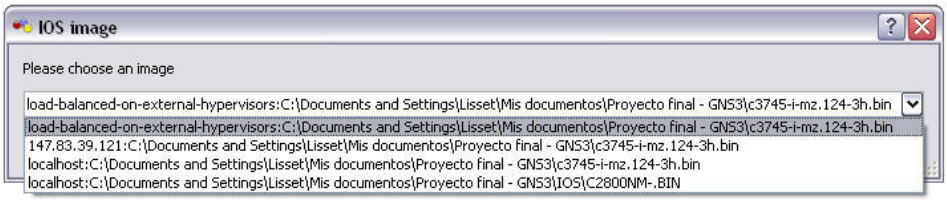


Figura 3.59: Ventana de elección de IOS.

CAPÍTULO 4: ANÁLISIS DEL RENDIMIENTO ENTRE EQUIPO REAL Y VIRTUAL

En este capítulo nos centraremos en medir el rendimiento de un router real y un router emulado con GNS3, trataremos de estudiar cómo se comportan estos elementos en las diferentes pruebas realizadas. Dicho estudio contrastará los valores de rendimiento obtenidos con dispositivos virtuales y reales, y nos dará una idea de la eficacia que tendría el incorporar dispositivos emulados en redes reales.

Para medir el rendimiento de los dispositivos realizaremos 3 pruebas: la primera tiene como objetivo cuantificar el tiempo de respuesta de los equipos, la segunda trata de medir la velocidad máxima de transmisión de datos alcanzada en el enlace sin que se produzcan pérdidas de paquetes, y la última prueba se centrará en mostrar el Jitter de un enlace; para las dos últimas pruebas utilizaremos un programa llamado "Iperf" que ya fue descrito en el capítulo 2.

4.1 Latencia

DEFINICIÓN: La Latencia o tiempo de reacción se define como el tiempo que tarda en transferirse un paquete de información desde un punto a otro de la red. En dispositivos de red como routers, podemos detallar esta definición diciendo que es el intervalo de tiempo que comienza cuando el final del primer bit de la trama entrante llega a la interfaz de entrada y termina cuando el comienzo del primer bit de la misma trama es visto en la interfaz de salida. Este valor es de suma importancia en la medida del rendimiento de una red ya que pase a que la velocidad de transferencia de datos en un momento sea alta, es posible que la transmisión de datos sea deficiente debido a que los nodos tarden mucho tiempo en conectarse. Ver punto 3.8 de RFC 1242 [22]

4.1.1 Objetivo

El objetivo de esta prueba es medir, teniendo como base el procedimiento descrito en el punto 26.2 de la RFC 2544 [23], y comparar, el tiempo de respuesta de un router emulado y de un router real por medio del valor promedio de RTT (**R**ound-**T**rip-**T**ime) proporcionado por el comando "ping". También se comentarán los cambios que se producen en la Latencia al modificar la capacidad de procesamiento del PC que contiene al emulador GNS3.

4.1.2 Escenario de prueba

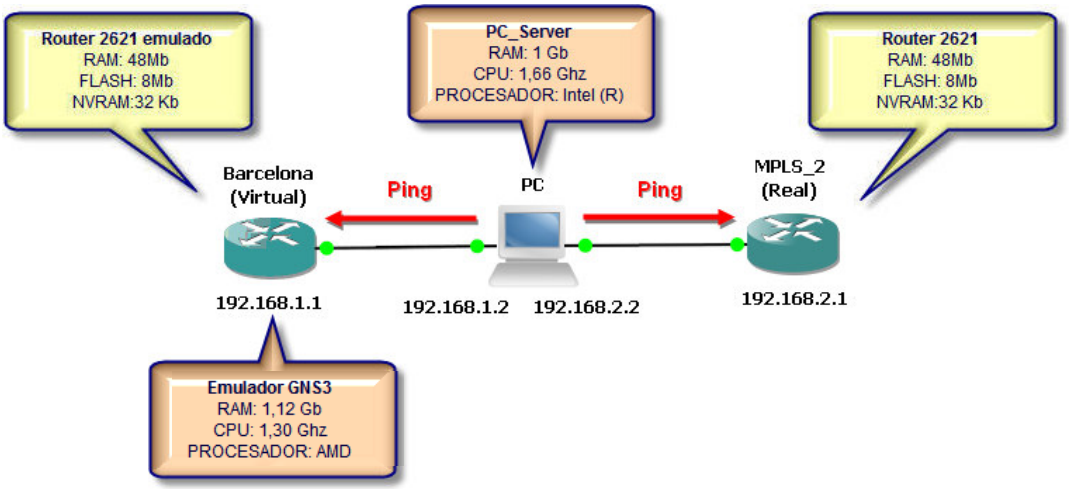


Figura 4.1: Primer Escenario de prueba.

En la tabla siguiente se muestran todos los elementos involucrados en el escenario con sus nombres y principales características:

	Nombre	Tipo	Memoria	CPU	Observaciones
Router	Barcelona	Virtual	RAM: 48Mb FLASH: 8Mb NVRAM:32 Kb	-	Router 2621 emulado
	MPLS_2	Real	RAM: 48Mb FLASH: 8Mb NVRAM:32 Kb	-	Router 2621
PC	Simulador GNS3	Real	RAM: 1,12 Gb	1,30 Ghz	Con procesador AMD
	PC_Server	Real	RAM: 1Gb	1,66 Ghz	Con procesador Intel(R)
Interfaces	Fastethernet 0/1	Virtual	IP: 192.168.1.1		Perteneciente a Barcelona
	Fastethernet 0/1	Real	IP: 192.168.2.1		Perteneciente a MPLS_2
	Adaptador de red 1	Real	IP: 192.168.1.2		Perteneciente al servidor FTP
	Adaptador de red 2	Real	IP: 192.168.2.2		Perteneciente a servidor FTP

Tabla 4.1: Elementos que aparecen en el escenario.

Notar que para la prueba se tomó especial cuidado en asignar al router emulado la misma IOS (***c2600-i-mz.123-3h***) y los mismos parámetros de memoria con los que cuenta el router real. Por otro lado, la elección del nombre que aparece para el router emulado se realizó por razones didácticas.

4.1.2.1 Configuración del escenario

En esta primera prueba, comenzaremos configurando el router Barcelona al que asignamos la dirección ip correspondiente a la interfaz fastethernet 0/1 e habilitamos el interfaz.

```
Barcelona#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Barcelona(config)#interface fastethernet 0/1
Barcelona(config-if)#ip address 192.168.1.1 255.255.255.0
Barcelona(config-if)#no shut
```

Figura 4.2: Configuración de la interfaz 0/1 de Barcelona. [12]

Por otro lado, repetimos los comandos para el router MPLS_2, asignando la ip que le corresponde a la interfaz fastethernet 0/1 e habilitando dicha interfaz.

```
MPLS_2#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
MPLS_2(config)#interface fastethernet 0/1
MPLS_2(config-if)#ip address 192.168.2.1 255.255.255.0
MPLS_2(config-if)#no shut
```

Figura 4.3: Configuración de la interfaz 0/1 de MPLS_2. [12]

Finalmente, asignamos 2 direcciones ips a los dos adaptadores de red con los que cuenta el ordenador PC_Server, teniendo en cuenta que pertenezcan a la red que deben.

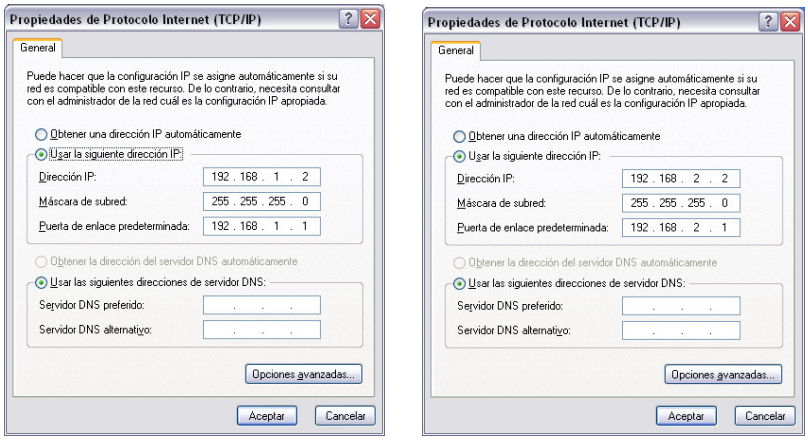


Figura 4.4: Configuración de parámetros en PCs.

4.1.3 Medición y tratamiento de datos

PROCEDIMIENTO: Se tomarán **10** muestras sucesivas de **100 ping** cada una, en donde se enviarán **paquetes de longitudes diferentes**. Las muestras se realizarán en sentido que va desde el ordenador PC_Server al router emulado Barcelona, y desde el mismo al router real MPLS_2. Las longitudes de tramas recomendadas, en la RFC 2544 [23], para la realización de mediciones en un medio Ethernet son: 64, 128, 256, 512, 1024, 1280 y 1518 octetos.

El comando utilizado en el PC_Server para la medición de la capacidad respuesta del router virtual Barcelona es:

```
ping 192.168.1.1 -l 22 -n 100
```

Figura 4.5: Comando usado en el PC de pruebas.

La tabla 4.2 muestra los datos promedios finales obtenidos en el router Barcelona después de tomar las muestras con diferentes longitudes de tramas Ethernet.

Longitud de trama (octetos)	64	128	256	512	1024	1280	1512
Round-trip min (ms)	3	5	6	7	8	9	11
Round-trip avg (ms)	**7**	**9**	**9**	**10**	**11**	**13**	**14**
Round-trip max (ms)	38	38	41	42	42	45	45

Tabla 4.2: Datos obtenidos para el router virtual Barcelona.

Seguidamente, el comando utilizado para la medición de la capacidad respuesta del router MPLS_2:

```
ping 192.168.2.1 -l 22 -n 100
```

Figura 4.5: Comando usado en el PC de pruebas.

La tabla 4.3 muestra los datos de Latencia promedio obtenidos en el router MPLS_2 para las diferentes longitudes de trama.

Longitud de trama (octetos)	64	128	256	512	1024	1280	1512
Round-trip min (ms)	<1	1	1	1	1	1	1
Round-trip avg (ms)	**<1**	**1**	**1**	**1**	**1**	**1**	**1**
Round-trip max (ms)	1	2	2	2	2	2	2

Tabla 4.3: Datos obtenidos para el router MPLS_2.

Finalmente repetimos todo el proceso anterior usando un escenario modificado, esta vez, los PCs involucrados poseen capacidades de procesamiento diferentes. Ver figura 4.6.

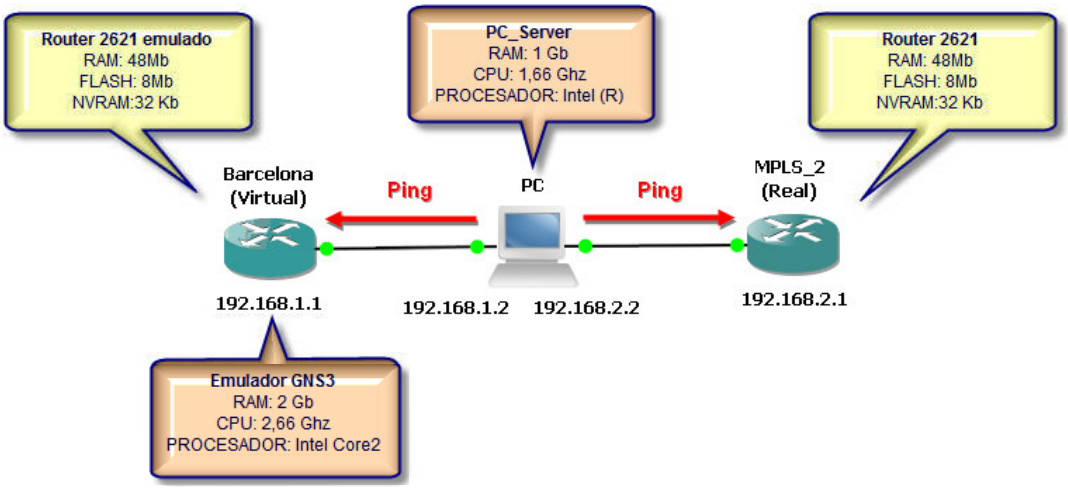


Figura 4.6: Escenario de prueba modificado.

La tabla 4.4 nuestra los promedios de Round-Trip obtenidos para el router Barcelona en el escenario modificado.

Longitud de trama (octetos)	64	128	256	512	1024	1280	1512
Round-trip min (ms)	1	1	2	2	3	4	5
Round-trip avg (ms)	**2**	**3**	**3**	**4**	**4**	**5**	**5**
Round-trip max (ms)	5	6	6	7	13	16	21

Tabla 4.4: Nuevos datos obtenidos para el router virtual Barcelona.

4.1.4 Captura de tráfico

Usaremos la capacidad de GNS3 para capturar las tramas que pasan por el enlace que va hacia el router Barcelona en la figura 4.6. La longitud de los datos ICMP es variable. Ver figura 4.7.

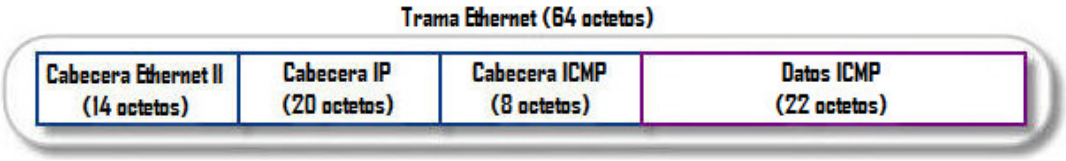
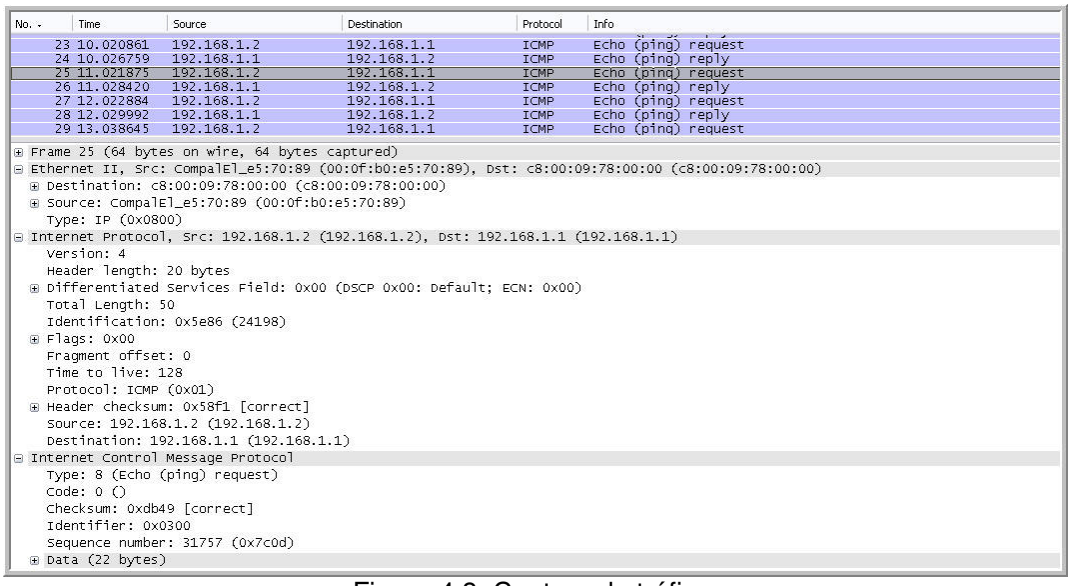


Figura 4.7: Trama más pequeña enviada para la medición de Latencia.

A continuación se muestra la captura realizada usando Wireshark (figura 4.8), en ella apreciamos la organización de los bits en los paquetes enviados, así como también, las direcciones MAC (las cuales cuentan con una nomenglatura particular que se explicará en el apartado 5.6) e IP de origen y destino configuradas. Por otro lado, observamos claramente la transmisión y recepción de paquetes ICMP debido al uso del comando *ping*.



No.	Time	Source	Destination	Protocol	Info
23	10.020861	192.168.1.2	192.168.1.1	ICMP	Echo (ping) request
24	10.026759	192.168.1.1	192.168.1.2	ICMP	Echo (ping) reply
25	11.021875	192.168.1.2	192.168.1.1	ICMP	Echo (ping) request
26	11.028420	192.168.1.1	192.168.1.2	ICMP	Echo (ping) reply
27	12.022884	192.168.1.2	192.168.1.1	ICMP	Echo (ping) request
28	12.029992	192.168.1.1	192.168.1.2	ICMP	Echo (ping) reply
29	13.038645	192.168.1.2	192.168.1.1	ICMP	Echo (ping) request

```
⊞ Frame 25 (64 bytes on wire, 64 bytes captured)
⊟ Ethernet II, Src: CompalEl_e5:70:89 (00:0f:b0:e5:70:89), Dst: c8:00:09:78:00:00 (c8:00:09:78:00:00)
  ⊞ Destination: c8:00:09:78:00:00 (c8:00:09:78:00:00)
  ⊞ Source: CompalEl_e5:70:89 (00:0f:b0:e5:70:89)
    Type: IP (0x0800)
⊟ Internet Protocol, Src: 192.168.1.2 (192.168.1.2), Dst: 192.168.1.1 (192.168.1.1)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 50
    Identification: 0x5e86 (24198)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 128
    Protocol: ICMP (0x01)
  ⊞ Header checksum: 0x58f1 [correct]
    Source: 192.168.1.2 (192.168.1.2)
    Destination: 192.168.1.1 (192.168.1.1)
⊟ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0 ()
    Checksum: 0xdb49 [correct]
    Identifier: 0x0300
    Sequence number: 31757 (0x7c0d)
  ⊞ Data (22 bytes)
```

Figura 4.8: Captura de tráfico.

4.1.5 Resultados

En la figura 4.9 se muestra el tiempo de ida y vuelta promedio de un ping para el router MPLS_2, el router Barcelona y el router Barcelona cuando se mejoran las capacidades de procesamiento del emulador. Se aprecia que se mejora notablemente la Latencia en el enlace que contiene al router virtual simulado en un PC con mejor capacidad de procesamiento, registrando valores más cercanos al casi un milisegundo obtenido con el router real.

La Latencia típica de una red LAN ronda en torno a los 1 – 2 ms pudiendo aceptarse, una Latencia máxima de 10ms en redes generalmente grandes; del mismo modo, tenemos que para redes WAN la máxima Latencia con la que muchas aplicaciones pueden trabajar antes de que se produzcan problemas es de 380 ms. Vemos en la figura 4.9 que los valores obtenidos en las pruebas anteriores se encuentran dentro del rango típico (datos obtenidos del foro "experts-exchange.com").

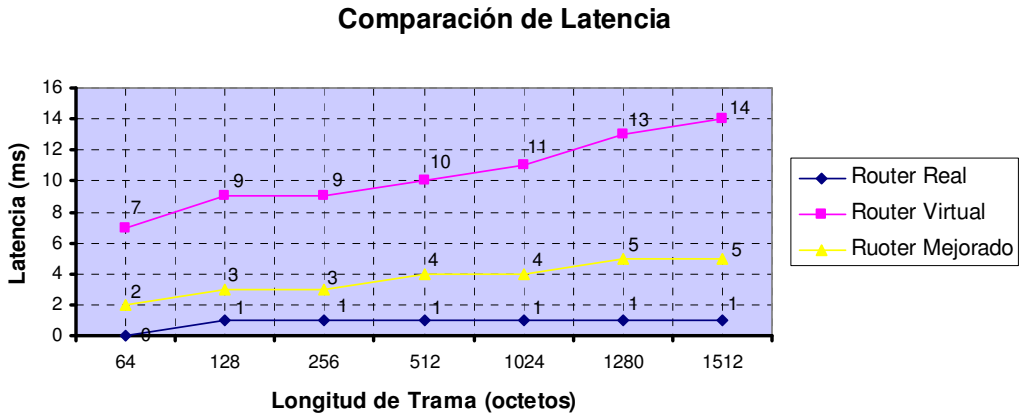


Figura 4.9: Comparación de Latencia.

4.1.6 Conclusiones y observaciones parciales

Después de la realización de esta prueba y bajo el escenario descrito en este capítulo, hemos podido llegar a las siguientes conclusiones y observaciones:

- Un router real nos ofrece retardos constantes y prácticamente nulos para cualquier longitud de paquete.
- El tiempo de respuesta del dispositivo de red emulado con GNS3 es más lento que el dispositivo de red real.
- Los valores de latencia de un router virtual aumentan conforme aumenta el tamaño de la trama que se envía.
- Al duplicar la capacidad de procesamiento del PC emulador del router se consiguen mejorar notablemente los valores de latencia.
- Se comprueba que enlace virtual es más lento y fácilmente congestionable que uno real.

4.2 Throughput (Caudal Eficaz)

DEFINICIÓN: El Throughput es la máxima velocidad de transmisión lograda en un enlace sin que se produzcan descartes de tramas en el dispositivo de red. Se mide en paquetes por segundo (pps) y su valor depende del tamaño de la trama de pruebas que se envíe por la red. Si una trama es larga tomará más tiempo en ser transmitida, como resultado, el router verá pocos paquetes en el intervalo de medida y tendrá más

tiempo para recuperarse de cada trama antes de que llegue la siguiente. Otros factores que afectan al Throughput son la forma en la que fluye el tráfico dentro del dispositivo de red, la realización de tareas adicionales en el router al momento de las pruebas, o el tipo de medio presentes en el router, en este caso, contamos con interfaces FastEthernet que pueden alcanzar teóricamente una velocidad de 100 Mbps. Ver el punto 3.17 de la RFC 1242. [22]

4.2.1 Objetivo

El objetivo de esta prueba es medir los valores de throughput alcanzados, teniendo como base el procedimiento descrito en el punto 26.1 de la RFC 2544 [23], en dos enlaces fastethernet, uno que contiene a un router emulado con GNS3 y el otro que contiene a un router real. También tendremos en cuenta los efectos producidos en las mediciones cuando se mejora la capacidad de procesamiento del emulador. Las medidas se obtendrán usando Iperf en los extremos del enlace.

4.2.2 Escenario de prueba

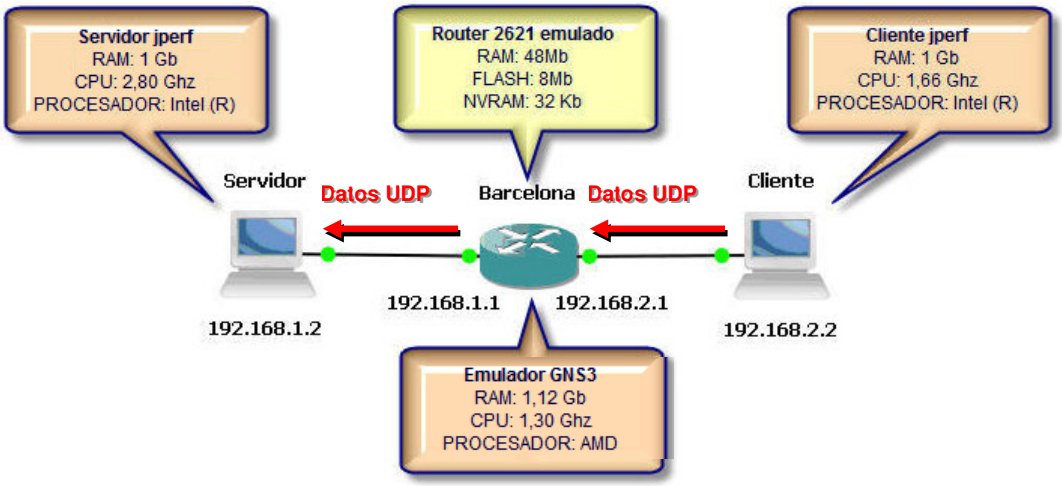


Figura 4.10a: Escenario de prueba para el router emulado con GNS3.

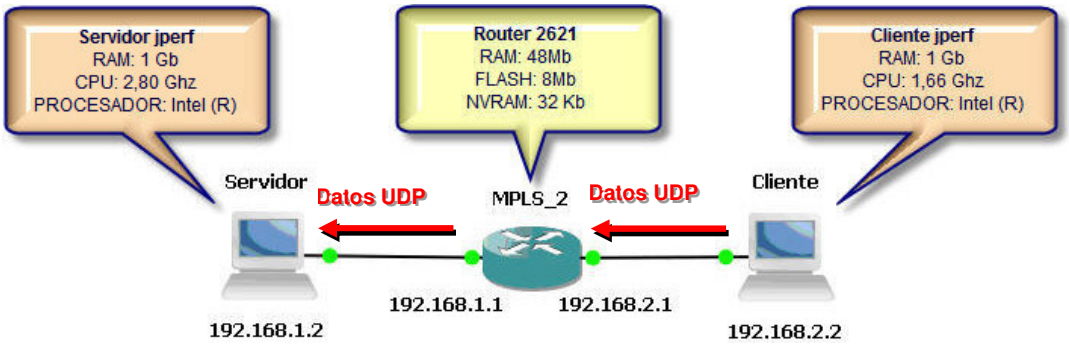


Figura 4.10b: Escenario de prueba para el router real.

En esta ocasión el escenario consta de dos partes separadas con el objetivo de obtener información individual que poder contrastar más adelante, en este sentido, cabe resaltar que se usaron las mismas direcciones IP y los mismos nombres de equipos por cuestiones de comodidad.

A continuación se muestran todos los elementos involucrados en este escenario con sus nombres y principales características:

	Nombre	Tipo	Memoria	CPU	Observaciones
Router	Barcelona	Virtual	RAM: 48Mb FLASH: 8Mb NVRAM:32 Kb	-	Router 2621 emulado
	MLPS	Real	RAM: 48Mb FLASH: 8Mb NVRAM:32 Kb	-	Router 2621
PC	Simulador GNS3	Real	RAM: 1,12 Mb	1,30 Mhz	Con procesador AMD
	Cliente JPerf	Real	RAM: 1Gb	1,66 Ghz	Con procesador Intel(R)
	Servidor JPerf	Real	RAM:1 Gb	2,80 Ghz	Con procesador Intel(R)
Interfaces	Fastethernet 0/0	Virtual	IP: 192.168.1.1		Perteneciente a Barcelona
	Fastethernet 0/1	Virtual	IP: 192.168.2.1		Perteneciente a Barcelona
	Fastethernet 0/0	Real	IP: 192.168.1.1		Perteneciente a MPLS_2
	Fastethernet 0/1	Real	IP: 192.168.2.1		Perteneciente a MPLS_2
	Adaptador de red 1	Real	IP: 192.168.1.2		Perteneciente al servidor iperf
	Adaptador de red 2	Real	IP: 192.168.2.2		Perteneciente al cliente iperf

Tabla 4.5: Elementos que aparecen en el escenario.

Para esta prueba también debemos indicar que se ha asignado al router emulado la misma IOS (***c2600-i-mz.123-3h***) y los mismos parámetros de memoria que los que tiene el router real.

4.2.2.1 Configuración del escenario

El escenario de prueba consta de dos partes, empezaremos explicando la configuración del router emulado, después la del router real y finalmente con la configuración de las interfaces de los PCs involucrados.

Para la primera parte de la prueba, asignamos, al router Barcelona, la dirección ip correspondiente a cada una de sus dos interfaces de red que vamos a usar y las habilitamos.

```
Barcelona#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Barcelona(config)#interface fastethernet 0/0
Barcelona(config-if)#ip address 192.168.1.1 255.255.255.0
Barcelona(config-if)#no shutdown
```

Figura 4.11: configuración de la interfaz 0/0 de Barcelona. [12]

```
Barcelona(config-if)#interface fastethernet 0/1
Barcelona(config-if)#ip address 192.168.2.1 255.255.255.0
Barcelona(config-if)#no shutdown
```

Figura 4.12: Configuración de la interfaz 0/1 de Barcelona. [12]

Después, para la segunda parte del escenario, realizaremos los mismos pasos anteriores pero en las interfaces del router MPLS_2, sin olvidar habilitarlas.

```
MPLS_2#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
MPLS_2(config)#interface fastethernet 0/0
MPLS_2(config-if)#ip address 192.168.1.1 255.255.255.0
MPLS_2(config-if)#no shutdown
```

Figura 4.13: Configuración de la interfaz 0/0 de MPLS_2. [12]

```
MPLS_2(config-if)#
MPLS_2(config-if)#interface fastethernet 0/1
MPLS_2(config-if)#ip address 192.168.2.1 255.255.255.0
MPLS_2(config-if)#no shutdown
```

Figura 4.14: Configuración de la interfaz 0/1 de MPLS_2. [12]

Acto seguido nos centraremos en la configuración de las interfaces de red de los dos PCs que serán usados en ambas partes del escenario. Ver figura 4.15.

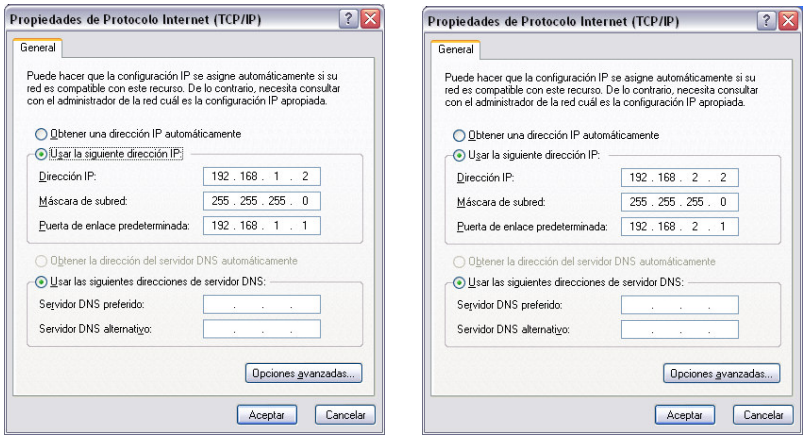


Figura 4.15: Configuración de parámetros en PCs.

Antes terminar toda la configuración de los elementos vamos a realizar una prueba de conectividad entre los dos PCs en ambas partes del escenario.

```
Haciendo ping a 192.168.2.2 con 32 bytes de datos:

Respuesta desde 192.168.2.2: bytes=32 tiempo=11ms TTL=127
Respuesta desde 192.168.2.2: bytes=32 tiempo=10ms TTL=127
Respuesta desde 192.168.2.2: bytes=32 tiempo=11ms TTL=127
Respuesta desde 192.168.2.2: bytes=32 tiempo=10ms TTL=127

Estadísticas de ping para 192.168.2.2:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 10ms, Máximo = 11ms, Media = 10ms
```

Figura 4.16: Ping desde el servidor al cliente iperf por medio de Barcelona.

```
Haciendo ping a 192.168.1.2 con 32 bytes de datos:

Respuesta desde 192.168.1.2: bytes=32 tiempo=1ms TTL=127
Respuesta desde 192.168.1.2: bytes=32 tiempo<1m TTL=127
Respuesta desde 192.168.1.2: bytes=32 tiempo<1m TTL=127
Respuesta desde 192.168.1.2: bytes=32 tiempo<1m TTL=127

Estadísticas de ping para 192.168.1.2:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 1ms, Media = 0ms
```

Figura 4.17: Ping desde el cliente al servidor iperf por medio de MPLS_2.

4.2.3 Medición y tratamiento de datos

PROCEDIMIENTO: Enviaremos un número específico de tramas a velocidades específicas hasta que encontremos la máxima cantidad de tramas recibidas en el destino sin que se produzcan perdidas de las mismas. Cada una de las ráfagas enviadas dura **60 segundos** y en ellas se transporta **datagramas UDP de longitud variable**. En las pruebas el cliente será el encargado de enviar los paquetes de datos, mientras que el servidor será el que los reciba y nos muestre la cantidad de paquetes que llegaron correctamente. Las longitudes de tramas recomendadas, en la RFC 2544 [23], para la realización de mediciones en un medio Ethernet son: 64, 128, 256, 512, 1024, 1280 y 1518 octetos.

Se elige usar el protocolo UDP en las pruebas de Throughput debido a que no implementa ventanas que otorgan control de flujo a la transmisión de bytes, de forma que no se limita el ancho de banda de la red que está disponible. Además, UDP tampoco realiza retransmisiones de bytes, ya que no espera un reconocimiento afirmativo (ACK) por parte del receptor de cada byte que envía, para enviar el siguiente.

El tamaño de los datagramas UDP considera que cada datagrama involucrado en la prueba se encapsula en un sólo paquete IP, de modo que se pueden obtener resultados en paquete por segundo (pps) y no en datagramas por segundo.

La figura 4.18 muestra los parámetros configurados en el cliente Iperf usando la interfaz gráfica jperf. Tanto el valor de *"UDP Bandwidth"* (en Mbits/seg) como el de *"UDP Packet Size"* irán cambiando al transcurrir la prueba.

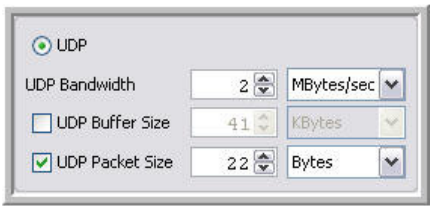


Figura 4.18: Parámetros para la medición de Throughput.

Cabe resaltar que Iperf muestra los paquetes recibidos en cada segundo y también los recibidos durante toda la prueba, lo cual facilitó el proceso de toma de medidas. Además, los resultados que se muestran en las tablas siguientes son valores típicos obtenidos después de la repetición sucesiva de cada una de las pruebas.

La tabla 4.6 muestra los resultados obtenidos tras la realización de las pruebas en el enlace que involucra al router virtual Barcelona.

Longitud de trama (octetos)	64	128	256	512	1024	1280	1512
Velocidad de Tx (Mbps)	0,12	0,4	0,8	1,2	1,5	1,6	1,6
Velocidad de Rx (Mbps)	0,12	0,39	0,78	1,2	1,48	1,59	1,6
Tramas Transmitidas	40273	33864	27193	18717	11314	9641	8149
Tramas Recibidas	40273	33864	27193	18717	11314	9641	8149
Tramas Perdidas	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
Tramas Recibidas (pps)	**671,2**	**564,4**	**453,2**	**311,9**	**188,5**	**160,7**	**135,8**

Tabla 4.6: Datos obtenidos en el enlace que contiene a Barcelona.

La tabla 4.7 muestra los resultados obtenidos tras la realización de las pruebas en el enlace que involucra al router real MPLS_2.

Longitud de trama (octetos)	64	128	256	512	1024	1280	1512
Velocidad de Tx (Mbps)	2,7	3,6	4,2	4,9	5	5,1	5,2
Velocidad de Rx (Mbps)	2,69	3,6	4,18	4,89	4,97	5,1	5,19
Tramas Transmitidas	924406	309207	146221	77643	37934	31190	26104
Tramas Recibidas	924406	309207	146221	77643	37934	31190	26104
Tramas Perdidas	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
Tramas Recibidas (pps)	**15407**	**5153,4**	**2437**	**1294,1**	**632,2**	**519,8**	**435**

Tabla 4.7: Datos obtenidos en el enlace que contiene a MPLS_2.

Finalmente repetimos el proceso de medición de Throughput para el enlace que contiene al router Barcelona pero esta vez modificamos la capacidad de procesamiento del PC con el emulador GNS3. Ver figura 4.19.

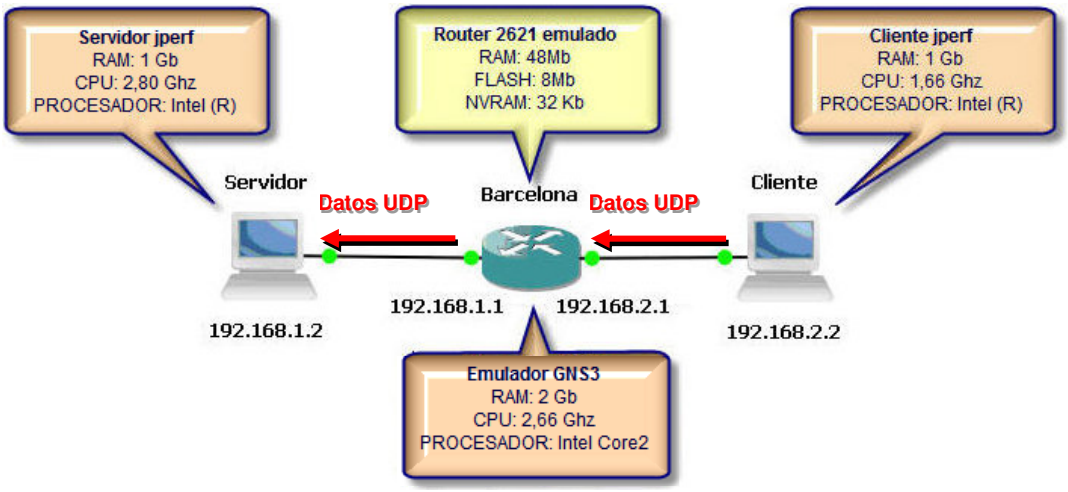


Figura 4.19: Escenario de prueba para el router emulado modificado.

Los nuevos resultados obtenidos se muestran en la tabla 4.8.

Longitud de trama (octetos)	64	128	256	512	1024	1280	1512
Velocidad de Tx (Mbps)	0,18	0,5	0,9	1,35	1,48	1,49	1,65
Velocidad de Rx (Mbps)	0,18	0,5	0,9	1,35	1,48	1,49	1,65
Tramas Transmitidas	61414	43595	31547	21545	11305	9029	8420
Tramas Recibidas	61414	43595	31547	21545	11305	9029	8420
Tramas Perdidas	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
Tramas Recibidas (pps)	**1023,5**	**726,6**	**525,8**	**359,1**	**188,4**	**150,5**	**140,3**

Tabla 4.8: Datos obtenidos para el escenario modificado.

4.2.4 Captura de tráfico

Usaremos la capacidad de GNS3 para capturar las tramas que pasan por uno de los enlaces simulados de la figura 4.10a. Ver figura 4.20.

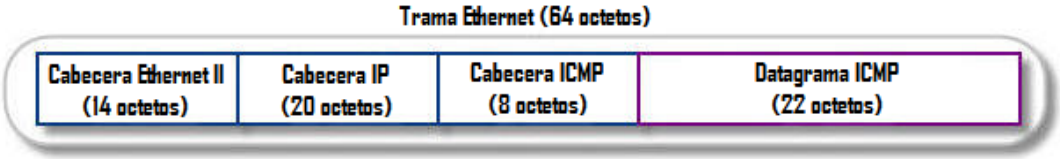
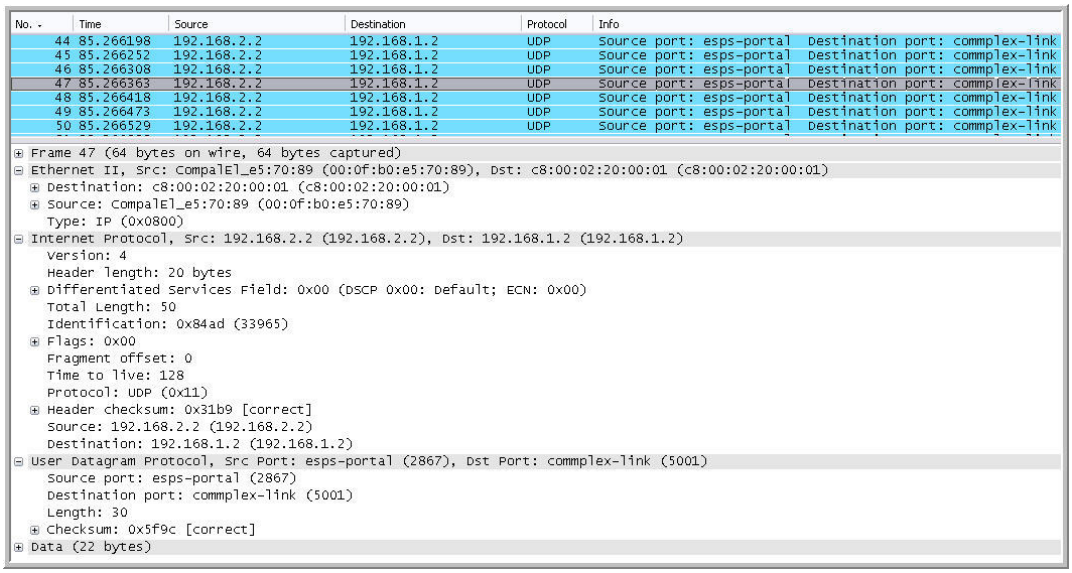


Figura 4.20: Trama más pequeña enviada para la medición del throughput.

La figura 4.21 muestra la captura realizada usando Wireshark, en ella apreciamos principalmente la disposición exacta de los bits en los paquetes enviados, así como también, las direcciones MAC (ver apartado 5.6.1) e IP de los dispositivos de origen y destino, el protocolo que transportan (en este caso UDP) y como se encapsulan para ser transportados.



No.	Time	Source	Destination	Protocol	Info
44	85.266198	192.168.2.2	192.168.1.2	UDP	Source port: esps-portal Destination port: commplex-link
45	85.266252	192.168.2.2	192.168.1.2	UDP	Source port: esps-portal Destination port: commplex-link
46	85.266308	192.168.2.2	192.168.1.2	UDP	Source port: esps-portal Destination port: commplex-link
47	85.266363	192.168.2.2	192.168.1.2	UDP	Source port: esps-portal Destination port: commplex-link
48	85.266418	192.168.2.2	192.168.1.2	UDP	Source port: esps-portal Destination port: commplex-link
49	85.266473	192.168.2.2	192.168.1.2	UDP	Source port: esps-portal Destination port: commplex-link
50	85.266529	192.168.2.2	192.168.1.2	UDP	Source port: esps-portal Destination port: commplex-link

```
⊞ Frame 47 (64 bytes on wire, 64 bytes captured)
⊟ Ethernet II, Src: CompalEl_e5:70:89 (00:0f:b0:e5:70:89), Dst: c8:00:02:20:00:01 (c8:00:02:20:00:01)
  ⊞ Destination: c8:00:02:20:00:01 (c8:00:02:20:00:01)
  ⊞ Source: CompalEl_e5:70:89 (00:0f:b0:e5:70:89)
    Type: IP (0x0800)
⊟ Internet Protocol, Src: 192.168.2.2 (192.168.2.2), Dst: 192.168.1.2 (192.168.1.2)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 50
    Identification: 0x84ad (33965)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 128
    Protocol: UDP (0x11)
  ⊞ Header checksum: 0x31b9 [correct]
    Source: 192.168.2.2 (192.168.2.2)
    Destination: 192.168.1.2 (192.168.1.2)
⊟ User Datagram Protocol, Src Port: esps-portal (2867), Dst Port: commplex-link (5001)
    Source port: esps-portal (2867)
    Destination port: commplex-link (5001)
    Length: 30
  ⊞ Checksum: 0x5f9c [correct]
⊞ Data (22 bytes)
```

Figura 4.21: Captura de tráfico.

4.2.5 Resultados

En el apartado anterior hemos obtenido valores de throughput para diferentes longitudes de trama, un grupo corresponde al enlace que contiene al router Barcelona, otro al mismo router Barcelona pero simulado en un PC con mejores prestaciones y por último uno que contiene al router MPL_2, dichos valores están representados en paquetes por segundo (ver figura 4.22).

Las especificaciones de CISCO acerca del rendimiento de un router 2621 [24] nos indican que puede alcanzar un throughput de **1500 pps** (0,768Mbps) cuando trabaja en modo Process-switching y **25000 pps** (12,80 Mbps) cuando lo hace en el modo Fast-switching ó CEF-switching; siempre y cuando se tenga en cuenta las mejores condiciones, es decir, asumiendo que sólo se envían paquetes IP y que todos están en su mínimo tamaño, es decir, **64 octetos**. Notamos que el throughput varía mucho dependiendo del modo de envío de paquetes que se use. En el modo *Process-switching* el router no conoce la dirección de red del paquete que acaba de llegar, con lo cual, invoca el uso de la CPU y no envía un paquete hasta que le llegue la tarea explicita para hacerlo. En el modo *Fast-switching* el router ya tiene la dirección de red del paquete en el caché, con lo cual para lo que esté haciendo y envía el paquete inmediatamente haciendo de este modo más rápido y eficiente. El modo *CEF (CISCO Express Fordwarding)* trata de solucionar las deficiencias de los dos modos anteriores por medio de la creación de una tabla adicional para el envío de paquetes.

Sabemos que los datos anteriores son puramente orientativos y nos sirven para darnos una idea de los valores que nos podemos encontrar, en nuestro caso, obtenemos un throughput de **15407 pps** (2,7 Mbps) en el router real MPLS_2 cuando enviamos paquetes de **64 octetos**, este valor esta dentro del intervalo esperado ya que el router no tenía su cache previamente configurada ni se configuró el modo CEF, pero siempre se envió paquetes a la misma dirección.

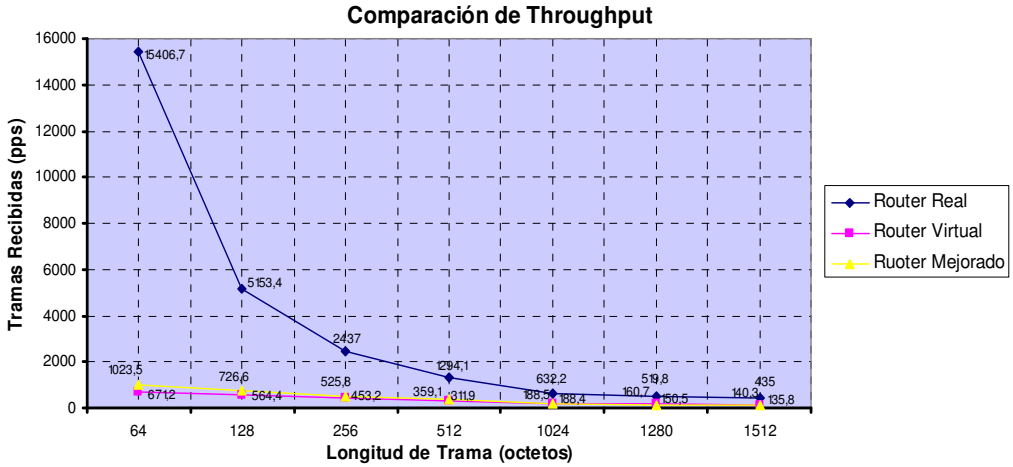


Figura 4.22: Resultados finales.

4.2.6 Conclusiones y observaciones parciales

Después de la realización de esta prueba y bajo el escenario descrito en este capítulo, hemos podido llegar a las siguientes conclusiones y observaciones:

- Según especificaciones de CISCO [24] los datos obtenidos para el Throughput de un router real son aceptables.
- Se sabe que toma menos tiempo transmitir un paquete corto que uno largo, es por eso que todos los dispositivos envían mas paquetes cortos durante la prueba.
- Los valores de Throughput de un router virtual y un router virtual emulado en un PC con el doble de capacidad de procesamiento no son aceptables pues que están muy por debajo del esperado.
- Duplicar las características del emulador no ayuda a mejorar el Throughput del enlace.
- Se comprueba que enlace virtual es más lento y fácilmente congestionable que uno real.

4.3 Jitter

DEFINICIÓN: El Jitter es la variación en el tiempo de la llegada de los paquetes en redes basadas en la conmutación de paquetes y que no son orientadas a conexión. Se produce debido a la congestión de la red, pérdida de sincronización debido a cambios mecánicos o eléctricos, o por que los paquetes siguen diferentes rutas para llegar a su destino. Es un efecto que tiene bastante relevancia en comunicaciones en tiempo real como VoIP ya que no depende de la Latencia, es decir, se pueden conseguir altos tiempos de respuesta pero sin embargo muy bajos valores de Jitter.

4.3.1 Objetivo

El objetivo de esta prueba es medir el Jitter en un enlace Fastethernet que contiene a un router emulado con GNS3, con el de otro enlace que contiene a un router real, usando, para la toma de datos, dos PCs con Iperf. Además veremos los efectos producidos en los resultados al mejorar la capacidad de procesamiento del host que contiene al simulador.

4.3.2 Escenario de prueba

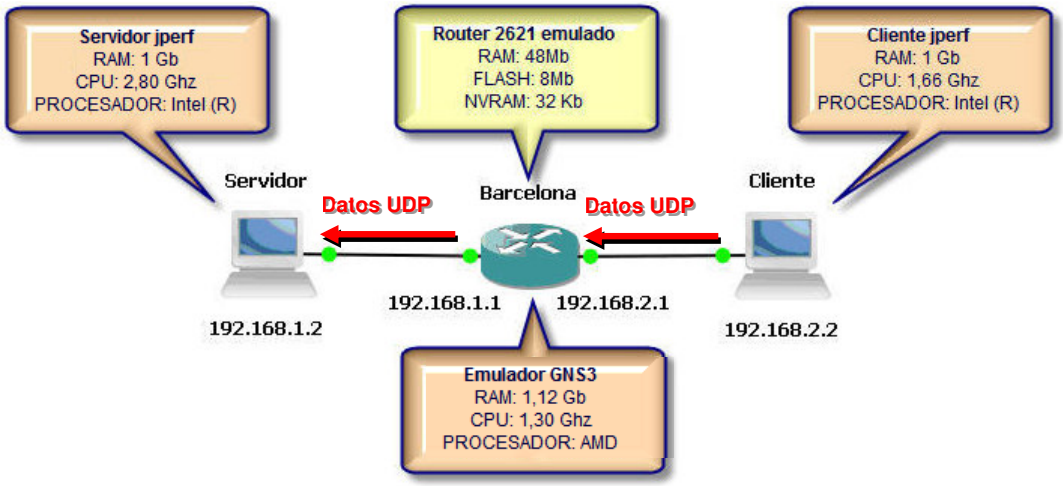


Figura 4.23a: Escenario de prueba para el router emulado con GNS3.

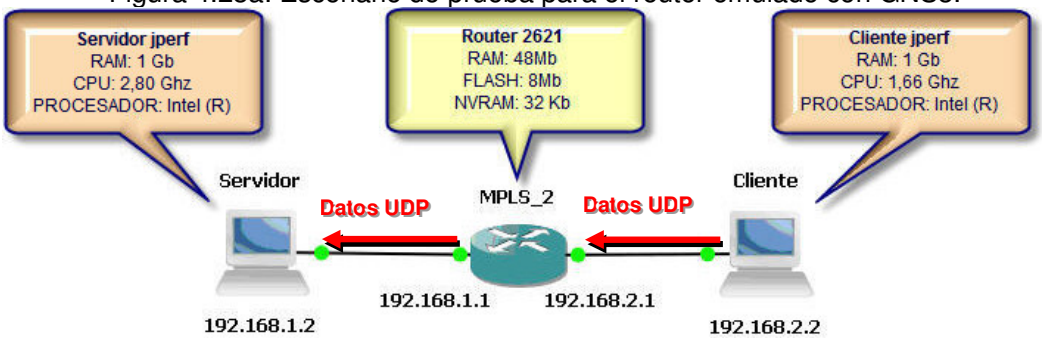


Figura 4.23b: Escenario de prueba para el router real.

En esta prueba, por cuestiones de comodidad, se ha usado el mismo escenario que para la prueba anterior, con lo cual los elementos involucrados y la configuración de los mismos siguen siendo idénticos.

4.3.3 Medición y tratamiento de datos

PROCEDIMIENTO: Enviaremos ráfagas que contienen datagramas UDP a la velocidad máxima de Throughput calculada en el apartado 4.2. La longitud de los **datos UDP es variable** y la duración de la prueba es de **60 segundos**. El origen de los datos será el cliente y el destino será el receptor que nos mostrará los valores de jitter obtenidos durante la transmisión.

Para la realización del cálculo del Jitter por Iperf, el cliente almacena una marca de tiempo en cada paquete que envía y el servidor calcula el tiempo de tránsito relativo como la diferencia entre ***tiempo de recepción del servidor – tiempo de envío del cliente***. Puesto que los relojes del cliente y del servidor no necesitan ser sincronizados, cualquier diferencia presente será reconocida como un nuevo valor de Jitter. Finalmente el Jitter será la media de las diferencias entre tiempos de tránsito consecutivos. [16]

Se ha vuelto a elegir al protocolo UDP en estas nuevas pruebas debido a que no queremos que se produzca un control de flujo en la transmisión de datos, ni que se reciba un reconocimiento afirmativo de la recepción de los datos; con el fin de obtener tráfico simular al que se produce, por ejemplo, en la comunicación de voz. Por lo tanto, los parámetros configurados en el cliente Iperf son los mismos que los que se muestran en la gráfica 4.18.

Por otro lado, los resultados que se muestran en las tablas siguientes son valores típicos obtenidos despues de la repetición sucesiva de cada una de las pruebas.

La tabla 4.9 muestra el valor de Jitter obtenido tras el envío de datos a la velocidad máxima de Throughput en el enlace que involucra al router Barcelona. El Jitter se obtuvo promediando los 60 datos obtenidos durante el minuto que dura la prueba.

Longitud de trama (octetos)	64	128	256	512	1024	1280	1512
Velocidad de Tx (Mbps)	0,12	0,4	0,8	1,2	1,5	1,6	1,6
Tramas Transmitidas	40273	33864	27193	18717	11314	9641	8149
Tramas Perdidas	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
Jitter (ms)	**3,524**	**4,592**	**4,073**	**4,706**	**8,878**	**7,245**	**9,539**

Tabla 4.9: Jitter en el enlace que contiene a Barcelona.

La tabla 4.10 muestra el valor de Jitter obtenido tras el envío de datos a la velocidad máxima de Throughput en el enlace que involucra al router MPLS_2. El Jitter se obtuvo promediando los 60 datos obtenidos durante el minuto que dura la prueba.

Longitud de trama (octetos)	64	128	256	512	1024	1280	1512
Velocidad de Tx (Mbps)	2,7	3,6	4,2	4,9	5	5,1	5,2
Tramas Transmitidas	924406	309207	146221	77643	37934	31190	26104
Tramas Perdidas	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
Jitter (ms)	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Tabla 4.10: Jitter en el enlace que contiene a MPLS_2.

Finalmente repetimos el proceso de medición de Jitter para el enlace que contiene al router Barcelona pero esta vez modificamos la capacidad de procesamiento del PC con el simulador GNS3. Ver figura 4.24.

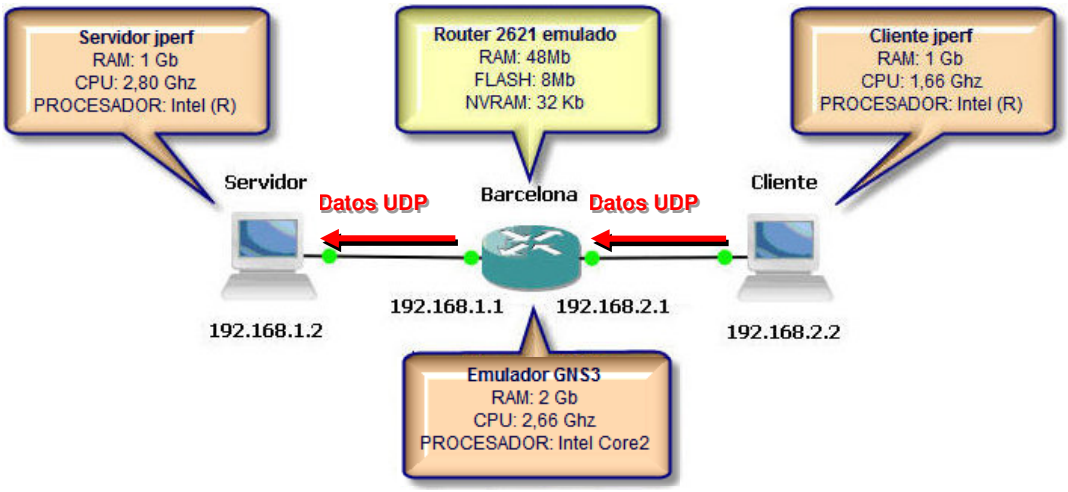


Figura 4.24: Escenario de prueba para el router simulado modificado.

Los nuevos resultados obtenidos se muestran en la tabla 4.11.

Longitud de trama (octetos)	64	128	256	512	1024	1280	1512
Velocidad de Tx (Mbps)	0,18	0,5	0,9	1,35	1,48	1,49	1,65
Tramas Transmitidas	61414	43595	31547	21545	11305	9029	8420
Tramas Perdidas	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
Jitter (ms)	**1,557**	**0**	**0**	**0**	**0**	**0,001**	**0,001**

Tabla 4.11: Datos obtenidos para el escenario modificado.

4.3.4 Captura de tráfico

Dado que para medir el Jitter también se usan tramas que encapsulan datagramas UDP y con la misma longitud que las usadas para la medición de throughput, entonces la estructura de la trama y la captura de datos usando GNS3 son las mismas que las mostradas en las gráficas 4.20 y 4.21 del apartado anterior respectivamente.

4.3.5 Resultados

La gráfica 4.25 muestra una síntesis de los valores de Jitter obtenidos para diferentes longitudes de trama.

En la gráfica 4.25, se aprecia que los valores típicos de Jitter citados anteriormente, resultan posibles de conseguir por un grupo de routers emulados con GNS3, tanto si se han instalado en un host poco eficiente como si se usan PCs más modernos y rápidos.

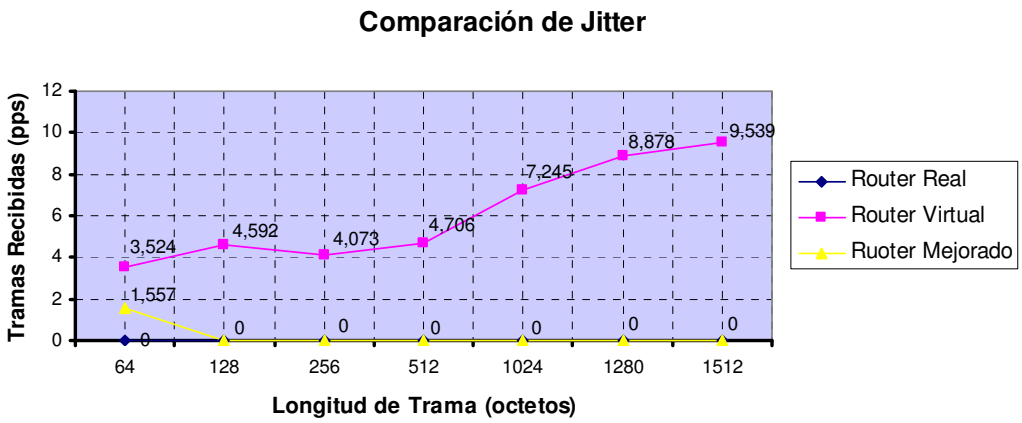


Figura 4.25: Promedio de datos obtenidos de Jitter.

4.3.6 Conclusiones y observaciones parciales

Después de la realización de esta prueba y bajo el escenario descrito en este capítulo, hemos podido llegar a las siguientes conclusiones y observaciones:

- Un router real nos ofrece valores constantes y prácticamente nulos de Jitter.
- Un router emulado con GNS3 registra valores de Jitter muy por encima de los registrados para un router real.
- Los valores de Jitter para el router emulado aumentan conforme se envían paquetes de mayor longitud.
- Un router virtual emulado en un PC con prácticamente el doble de capacidad de procesamiento muestra características muy similares a un router real.
- Se comprueba que enlace virtual es más lento y fácilmente congestionable que uno real.

4.4 Problemas detectados y soluciones

4.4.1 Obtención del IOS C2621

En el laboratorio de trabajo se cuenta con un router Cisco 2621 el cual ha sido destinado a las pruebas relacionadas con este proyecto, por lo tanto, era de suma importancia obtener un IOS con las mismas características del equipo real para que sea simulado usando GNS3 y así poder realizar pruebas que nos permitan sacar conclusiones acerca del rendimiento de ambos dispositivos de red en igualdad de condiciones.

El primer problema al que nos enfrentamos se relacionó con la obtención del IOS 2621 perteneciente al router MPLS_2 ubicado en el laboratorio. Para solucionar dicho percance se optó por instalar un servidor FTP y realizar una copia del IOS que se encontraba en el router MPLS_2 hacia dicho servidor usando el puerto 21.

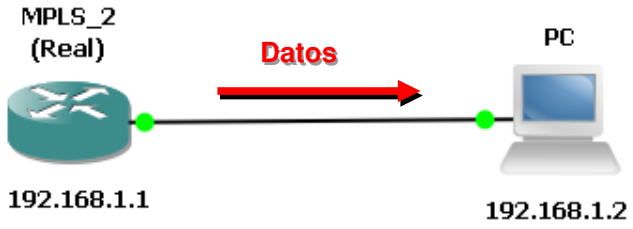


Figura 4.26: Escenario montado para la obtención del IOS del router MPLS_2.

En el PC de trabajo se realizaron principalmente dos cambios, se instaló el software Filezilla para convertirlo en servidor FTP y se configuró su interfaz de red según la figura, los pasos seguidos para los cambios mencionados se describen a continuación:

1) Deshabilitar el cortafuegos de Windows y de cualquier otro antivirus que se active por defecto (Norton).

2) Descargar de Internet el software libre FileZilla Server-0.9.31.

3) Hacer doble clic sobre el icono creado y aceptar los valores que se muestran a continuación que harán que nuestro propio PC actúe como servidor FTP. Finamente hacer clic en "*OK*".

Figura 4.27: Ventana de inicio.

4) Hacer un Clic en y en el menú de la Izquierda seleccionar "*General*". Aparecerá una ventana como se muestra en la figura 4.28.

5) Añadir un usuario anónimo haciendo clic en *"Add"* y escribiendo *"anonymous"*. Finalmente hacer clic en *"OK".*

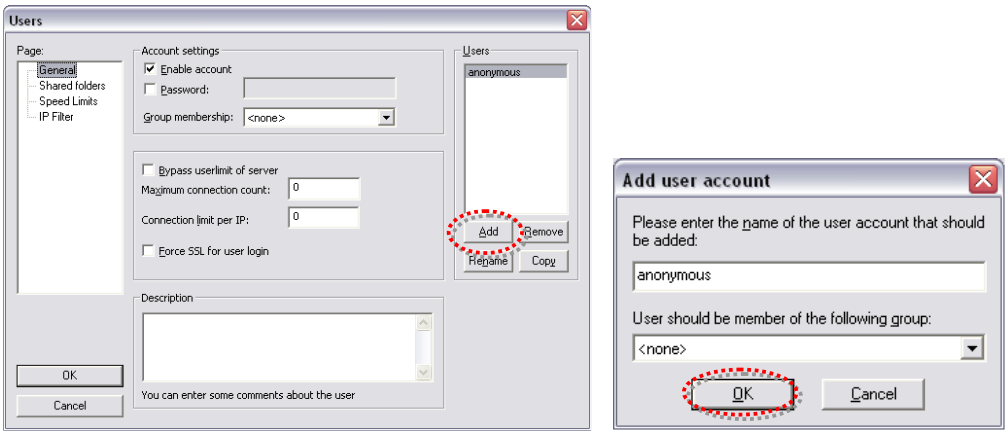


Figura 4.28: Agregación de nuevo usuario.

6) Seguidamente debemos asignar una carpeta "*home*" al nuevo usuario para que sea compartida. Hacer clic en *"Add"* y seleccionar el directorio correspondiente. Para terminar hacer clic en *"Aceptar"*.

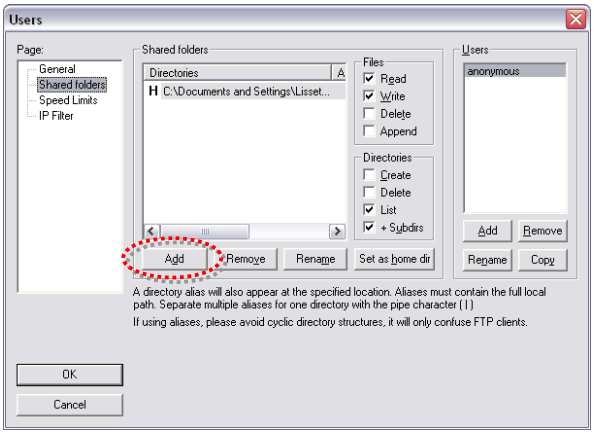


Figura 4.29: Asignación de carpeta al nuevo usuario.

7) No olvidar seleccionar los permisos de lectura y escritura de la carpeta.

Figura 4.30: Cuadro de asignación de permisos a la carpeta.

8) Finalmente configuramos la dirección IP correspondiente en el host.

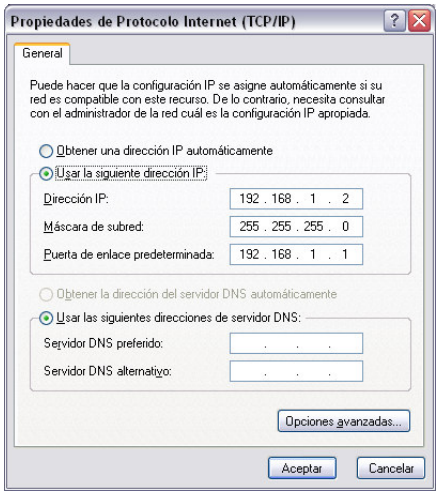


Figura 4.31: Configuración de parámetros en PCs.

Desde el punto de vista del router MPLS_2 se hicieron las siguientes configuraciones:

1) Configurar la dirección IP, de acuerdo al escenario propuesto, en la interfaz Fastethernet 0/1.

```
MPLS_2#conf ter
Enter configuration commands, one per line.  End with CNTL/Z.
MPLS_2(config)#interface fastethernet 0/1
MPLS_2(config-if)#ip address 192.168.1.1 255.255.255.0
MPLS_2(config-if)#no shutdown
```

Figura 4.32: Configuración de la interfaz 0/1 de MPLS_2. [12]

2) Ver la ubicación y el nombre del IOS en el router usando el comando **show flash**. Ver figura 4.33.

```
MPLS_2#show flash

System flash directory:
File  Length   Name/status
  1   7441364  c2600-i-mz.123-3h.bin
[7441428 bytes used, 947180 available, 8388608 total]
8192K bytes of processor board System flash (Read/Write)
```

Figura 4.33: Información de la memoria Flash. [12]

3) Realizar la copia de datos al servidor FTP que corresponde a nuestro PC de trabajo.

4.4.2 Formato de la cisco IOS en el router

Al momento de cargar la imagen IOS obtenida del router real en el simulador GNS3 y abrir una ventana de Telnet para conectarnos al router virtual nos encontramos con un error como el que se muestra en la figura 4.34.

```
Connected to Dynamips VM "R0" (ID 0, type c2600) - Console port

Smart Init is disabled. IOMEM set to: 15
                                        Using iomem percentage: 15
```

Figura 4.34: Error encontrado al cargar una imagen IOS.

Al principio se pensó que el problema se debía a que el router asignaba demasiada cantidad de memoria RAM a las interfaces de entrada y salida, lo que producía que no se pudiera cargar la IOS correspondiente, después se pensó que el problema era producto de deficiencias de Dynamips para cargar ciertas imágenes de la plataforma 2600. Finalmente nos dimos cuenta que Dynamips tiene problemas para descomprimir imágenes IOS de la plataforma 2600 exepto la 2691, con lo cual las imágenes que van a ser simuladas deben estar descomprimidas pese a llevar la extensión .bin al final del nombre del IOS. Para la descompresión debemos usar winrar en Windows y en Linux podemos utilizar el siguiente comando: [03] [27]

```
unzip -p c7200-g6ik8s-mz.124-2.T1.bin > c7200-g6ik8s-mz.124-2.T1.image
```

Figura 4.35: Comando para la descompresión de imagen en Linux. [03]

4.4.3 Igualdad de características en los equipos

Antes de comenzar las pruebas de tiempo de reacción, máxima tasa de transferencia y Jitter en el router MPLS_2 y en el router Barcelona, nos dimos cuenta que la clave

para conseguir unos resultados con coherencia era que ambos dispositivos de red contaran con los mismos parámetros de asignación de memoria.

Después de conocer el estado de ambos router mediante el comando **show version**, nos dimos cuenta que la diferencia entre ellos radicaba principalmente, en la asignación de memoria RAM dinámica o DRAM, y en la asignación memoria RAM no volátil o NVRAM de los mismos. Por otro lado, también notamos rasgos característicos propios de cada tipo de router con respecto a la memoria FLASH y ROM.

4.4.3.1 Memoria Dinámica (DRAM)

La DRAM (comúnmente conocida como memoria RAM) es una memoria que se usa en el momento de ejecución del router, por ejemplo, para ejecutar el software Cisco IOS, sus subsistemas, almacenar las tablas de enrutamiento, el cache de conmutación, la configuración que está corriendo actualmente (running-config), etc. Se divide en Main Processor Memory and Shared Input/Output (I/O) Memory, la Shared I/O Memory se divide entre interfaces de entrada y de salida para almacenar temporalmente los paquetes, mientras que la Process memory se usa para almacenar el software IOS de cisco, las tablas de enrutamiento y los buffers del sistema. [28]

La memoria DRAM que necesita cada router para trabajar depende de los requerimientos de la versión de CISCO IOS que utiliza, de modo que, en nuestro caso, para la versión de IOS **12.3 (3h)** en la plataforma 2600 se requiere como mínimo una cantidad de memoria DRAM de 32MB (según Cisco Systems .Inc), pero como el router real MPLS_2 tiene configurado 48MB de memoria DRAM (ver figura 4.37), nosotros asignaremos la misma cantidad de memoria RAM al router virtual. Para cambiar la cantidad de memoria DRAM asignada a cada router virtual emulado con GNS3, debemos modificar el valor de "*RAM size*" de la ventana de configuración del router Barcelona. Ver figura 4.36.

Por otro lado, dado que los routers serie 2600 poseen la capacidad de reasignar la cantidad de memoria DRAM repartida entre el procesador y las interfaces I/O mediante el comando **memory–size iomem <percent>,** en nuestro caso, usamos el comando anterior para cambiar el porcentaje de memoria RAM repartido, en el router MPLS_2 de un 10% que tenía inicialmente a un 15% que es el que tenía por defecto configurado el router virtual Barcelona.

4.4.3.2 Memoria no volátil (NVRAM)

La NVRAM se encargada de almacenar permanentemente datos como la configuración del router que inicialmente se encuentra en la RAM. Después de cada iniciación del router esta memoria debe enviar su información a la RAM para que se trabaje temporalmente sobre ella para agilizar la configuración del dispositivo. [28]

En este caso, cambiamos el valor de la NVRAM, desde la ventana de configuración del router Barcelona a 32Kb, para mantener la concordancia de configuración con el router MPLS_2.

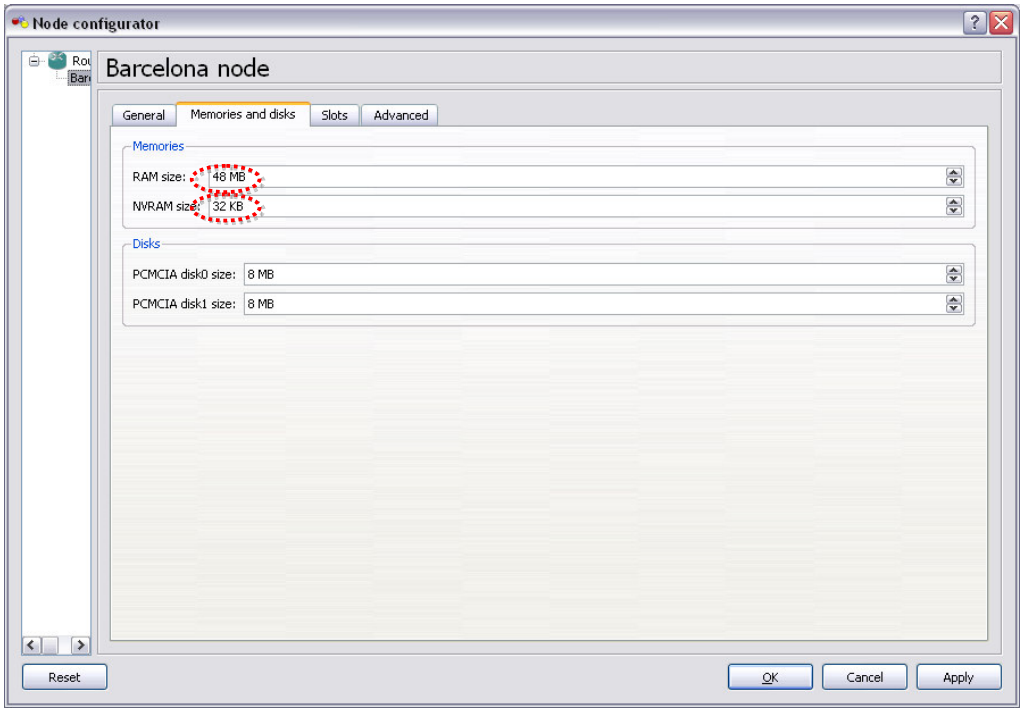


Figura 4.36: Ventana de configuración del nodo.

4.4.3.3 Memoria FLASH

Esta memoria almacena la imagen y microcódigo del sistema operativo que ésta siendo ejecutado por un router. En el router Barcelona, por tratarse de un router emulado, esta memoria no existe ya que el IOS se almacena en el disco. En la figura 4.38 vemos que GNS3 asigna un tamaño fijo de 8Mbytes a la FLASH de Barcelona.

4.4.3.4 Memoria ROM

La memoria ROM, como ya se indicó en el apartado 2.3, almacena un software con el estado mínimo de configuración de los routers (*software bootstrap)*, este software sólo puede ser invocado si se produce algún fallo interno del router ó de forma “fisica” durante el encendido de los routers, por lo tanto, la memoria ROM carece de funcionalidad en los routers emulados puesto que dicho *software* no puede ser invocado. En nuestro caso, la figura 4.38 muestra como, en el router virtual Barcelona, se incorporan unas líneas adicionales con el fin de ignorar la lectura de la ROM.

Las figuras 4.37 y 4.38 muestran la información obtenida después de usar el comando **show version** en el router real MPLS_2 y en el router virtual Barcelona respectivamente:

```
MPLS_2#show version
Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-I-M), Version 12.3(3h), RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2005 by cisco Systems, Inc.
Compiled Fri 01-Apr-05 15:39 by alnguyen
Image text-base: 0x80008098, data-base: 0x80C9A414

ROM: System Bootstrap, Version 11.3(2)XA4, RELEASE SOFTWARE (fc1)
ROM: C2600 Software (C2600-I-M), Version 12.3(3h), RELEASE SOFTWARE (fc2)

MPLS_2 uptime is 1 hour, 50 minutes
System returned to ROM by reload
System image file is "flash:c2600-i-mz.123-3h.bin"

cisco 2621 (MPC860) processor (revision 0x101) with 41984K/7168K bytes of memory
.
Processor board ID JAB0402017R (1642293183)
M860 processor: part number 0, mask 49
Bridging software.
X.25 software, Version 3.0.0.
2 FastEthernet/IEEE 802.3 interface(s)
32K bytes of non-volatile configuration memory.
8192K bytes of processor board System flash (Read/Write)

Configuration register is 0x2102
```

Figura 4.37: Parámetros finales del router MPLS_2.

```
Barcelona#show ver
Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-I-M), Version 12.3(3h), RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2005 by cisco Systems, Inc.
Compiled Fri 01-Apr-05 15:39 by alnguyen
Image text-base: 0x80008098, data-base: 0x80C9A414

ROM: ROMMON Emulation Microcode
ROM: C2600 Software (C2600-I-M), Version 12.3(3h), RELEASE SOFTWARE (fc2)

Barcelona uptime is 17 minutes
System returned to ROM by unknown reload cause - suspect boot_data[BOOT_COUNT] 0
x0, BOOT_COUNT 0, BOOTDATA 19
System image file is "tftp://255.255.255.255/unknown"

cisco 2621 (MPC860) processor (revision 0x202) with 41984K/7168K bytes of memory
.
Processor board ID 00000000000 (1880125456)
M860 processor: part number 0, mask 0
Bridging software.
X.25 software, Version 3.0.0.
2 FastEthernet/IEEE 802.3 interface(s)
32K bytes of non-volatile configuration memory.
8192K bytes of processor board System flash (Read/Write)

Configuration register is 0x2102
```

Figura 4.38: Parámetros finales del router Barcelona.

Vemos que en la figura 4.38, al router Barcelona se le asigna un procesador **CPU** con las mismas características que el real pero con identificador diferente (virtual).

CAPÍTULO 5: ANÁLISIS DE LAS COMUNICACIONES INVOLUCRADAS EN LA SIMULACIÓN

Este capítulo se centra principalmente en mostrar la capacidad del simulador de crear topologías que puedan soportar comunicaciones de diferentes tipos de protocolos de red en distintos tipos de escenarios, por medio de pruebas que nos permitan recoger datos que podremos analizar y comparar. Las pruebas se harán principalmente en dos tipos de escenarios, el primero contará solamente con routers simulados y el segundo incorporará un router real en su topología.

5.1 Pruebas con Comunicaciones Internas

5.1.1 Objetivo

Este apartado tiene el objetivo de mostrar las comunicaciones que se producen en escenarios puramente virtuales, es decir, que no contienen equipos reales. Del mismo modo, mediremos el funcionamiento de la red para contrastar los resultados obtenidos con el escenario que incluye un router real.

5.1.2 Escenario de prueba

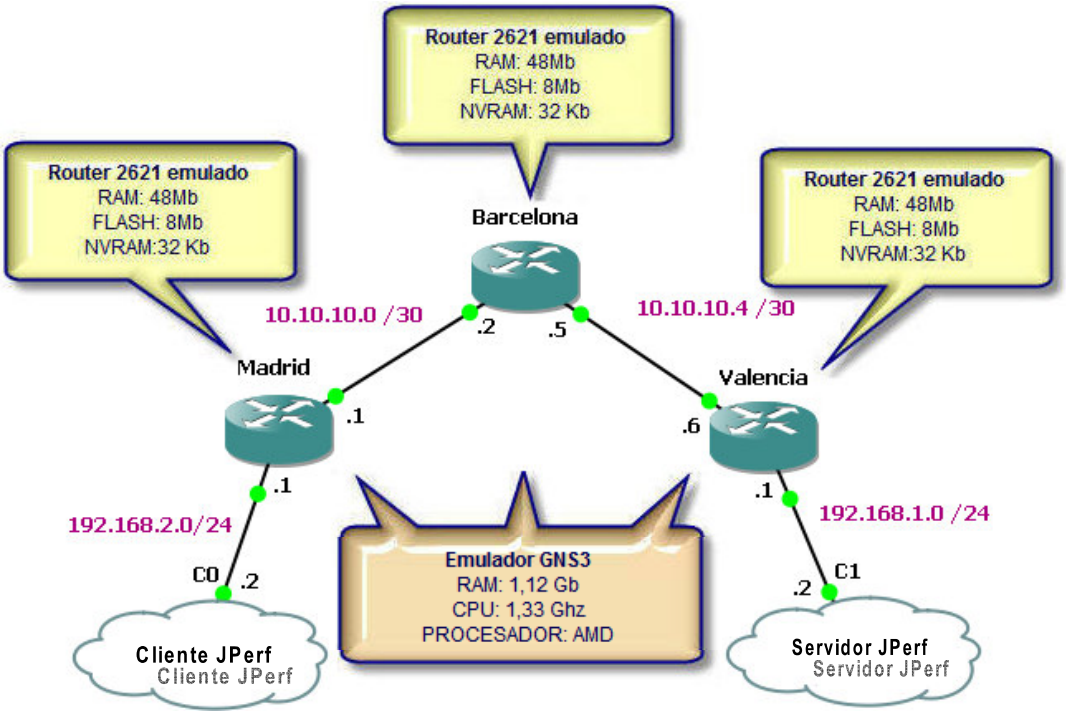


Figura 5.1: Escenario de la prueba puramente virtual.

En la figura 5.1 queda reflejado el escenario simulado, vemos que consta principalmente de tres routers emulados conectados entre ellos mediante enlaces virtuales.

A continuación se menciona, con más detalle, todos los elementos y sus características involucrados en este escenario:

	Nombre	Tipo	Memoria	CPU	Observaciones
Router	Madrid	Virtual	RAM: 48Mb FLASH: 8Mb NVRAM:120 Kb	-	Router 2621 emulado
	Barcelona	Virtual	RAM: 48Mb FLASH: 8Mb NVRAM:120 Kb	-	Router 2621 emulado
	Valencia	Virtual	RAM: 48Mb FLASH: 8Mb NVRAM:120 Kb	-	Router 2621 emulado
PC	Simulador GNS3	Real	RAM: 1,12Gb	1,30 Ghz	Con procesador AMD
	Cliente JPerf	Real	RAM: 1 Gb	1,66 Ghz	Con procesador Intel(R)
	Servidor JPerf	Real	RAM: 1 Gb	2,80 Ghz	Con procesador Intel
Interfaces	Fastethernet 0/0	Virtual	IP: 10.10.10.1		Perteneciente a Madrid
	Fastethernet 0/1	Real	IP: 192.168.2.1		Perteneciente a Madrid
	Fastethernet 0/0	Virtual	IP: 10.10.10.2		Perteneciente a Barcelona
	Fastethernet 0/1	Virtual	IP: 10.10.10.5		Perteneciente a Barcelona
	Fastethernet 0/0	Virtual	IP: 10.10.10.6		Perteneciente a Valencia
	Fastethernet 0/1	Real	IP: 192.168.1.1		Perteneciente a Valencia
	Adaptador de red 1	Real	IP: 192.168.1.2		Perteneciente a Servidor JPerf
	Adaptador de red 2	Real	IP: 192.168.2.2		Perteneciente a Cliente JPerf

Tabla 5.1: Elementos que aparecen en el escenario.

Para esta prueba se tomó especial cuidado en usar la misma IOS (***c2600-i-mz.123-3h***) y los mismos parámetros de memoria tanto en el router virtual como el router real. Por otro lado, la elección del nombre que aparece en los routeres emulados se realizó por razones didácticas.

5.1.2.1 Configuración del escenario

Configuramos las interfaces de los tres routers emulados de nuestra topología, los comandos utilizados se muestran a continuación:

```
Madrid#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Madrid(config)#interface fastethernet 0/0
Madrid(config-if)#ip address 10.10.10.1 255.255.255.252
Madrid(config-if)#no shutdown
Madrid(config-if)#interface fastethernet 0/1
Madrid(config-if)#ip address 192.168.2.1 255.255.255.0
Madrid(config-if)#no shutdown
Madrid(config-if)#exit
Madrid(config)#
```

Figura 5.2: Configuración de interfaces en el router Madrid. [12]

```
Barcelona#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Barcelona(config)#interface fastethernet 0/0
Barcelona(config-if)#ip address 10.10.10.2 255.255.255.252
Barcelona(config-if)#no shutdown
Barcelona(config-if)#interface fastethernet 0/1
Barcelona(config-if)#ip address 10.10.10.5 255.255.255.252
Barcelona(config-if)#no shutdown
Barcelona(config-if)#exit
Barcelona(config)#
```

Figura 5.3: Configuración de interfaces en el router Barcelona. [12]

```
Valencia#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Valencia(config)#interface fastEthernet 0/0
Valencia(config-if)#ip address 10.10.10.6 255.255.255.252
Valencia(config-if)#no shutdown
Valencia(config-if)#interface fastEthernet 0/1
Valencia(config-if)#ip address 192.168.1.1 255.255.255.0
Valencia(config-if)#no shutdown
Valencia(config-if)#exit
Valencia(config)#
```

Figura 5.4: Configuración de interfaces en el router Valencia. [12]

Finalmente se configurarán las interfaces de red de los dos PCs que actuarán como cliente y servidor Jperf en algunas de las pruebas siguientes. Ver figura 4.15.

5.1.3 Medición y tratamiento de datos

5.1.3.1 Prueba 1: Latencia de ICMP

PROCEDIMIENTO: Se enviarán **paquetes ICMP de longitud variable** usando la herramienta *"ping"* desde el host cliente JPerf hasta el host servidor JPerf en cada segundo. Las tramas enviadas tendrán longitudes de 64, 128, 256, 512, 1024, 1280 y 1512 octetos, mientras que los datos de latencia para cada longitud de trama se tomarán en pequeñas pruebas de **150 seg**.

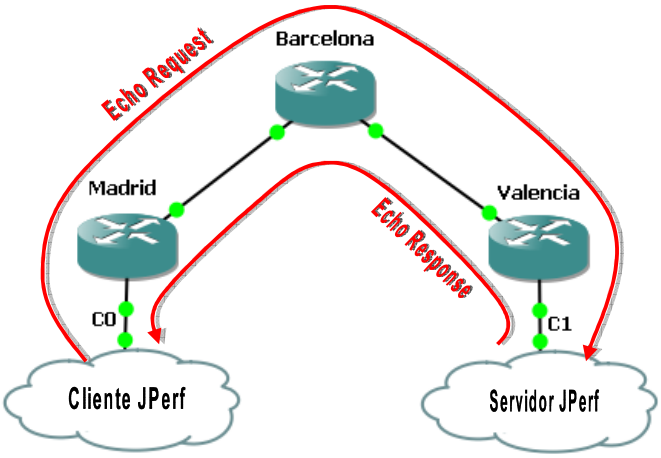


Figura 5.5: Descripción del procedimiento de la prueba.

La tabla 5.2 muestra los datos finales obtenidos en la red después de repetir el proceso de medición de datos **10 veces** para las diferentes longitudes de tramas Ethernet.

Longitud de trama (octetos)	64	128	256	512	1024	1280	1512
Round-trip min (ms)	51	53	53	61	78	84	92
Round-trip avg (ms)	**60**	**61**	**64**	**82**	**98**	**106**	**116**
Round-trip max (ms)	106	92	108	115	132	153	166

Tabla 5.2: Datos de latencia ICMP para la red puramente virtual.

5.1.3.2 Prueba 2: Latencia de TCP

PROCEDIMIENTO: Se enviarán **paquetes TCP** al puerto 23 (Telnet) del router Valencia desde el router Madrid cada minuto. El router Madrid será configurado para recolectar y realizar estadísticas de los tiempos de conexión TCP producidos en el enlace en presencia del protocolo TCP. Toda la prueba dura **60 min**.

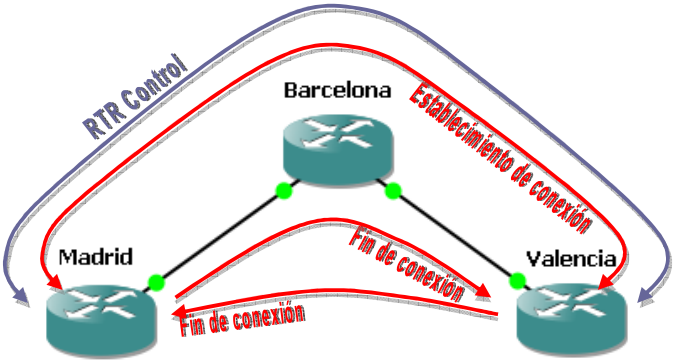


Figura 5.6: Descripción del procedimiento de la prueba.

Esta prueba hace uso de la herramienta RTR, la cual se configuró de la siguiente manera: [19]

1) Habilitar RTR en el router Valencia para activar temporalmente el servicio TCP. Ver figura 5.7

```
Valencia(config)#rtr responder
```

Figura 5.7: Configuración de RTR en Valencia

2) Definir una prueba tipo *"tcpConnect"* en el router recolector de datos (Madrid) para medir el tiempo que se tarda en realizar una conexión TCP desde el router origen hasta el destino. Por defecto, el router origen envía un paquete de control RTR antes de los de TCP para habilitar el puerto 23 en el router destino. En la figura 5.7 se muestran los parámetros adicionales necesarios por el comando, estos son: La dirección IP de destino y el puerto de destino.

```
Madrid(config)#rtr 2
Madrid(config-rtr)#type tcpConnect dest-ipaddr 10.10.10.6 dest-port 23
```

Figura 5.8: Configuración de la segunda prueba.

3) Definir con qué frecuencia se realiza cada conexión TCP durante la prueba. En este caso, el mínimo valor es 60 segundos.

```
Madrid(config-rtr-tcp)#frequency 60
Madrid(config-rtr-tcp)#exit
```

Figura 5.9: Configurar la frecuencia de envío de paquetes.

4) Damos un tiempo de vida a la prueba con el comando *"lifetime"*, en este caso, nuestra prueba correrá por 1 hora (3600 segundos). Finalmente indicamos que la prueba se inicie inmediatamente con el comando *"start-time now"*.

```
Madrid(config)#rtr schedule 2 life 3600 start-time now
```

Figura 5.10: Configuración de tiempo de vida de la prueba.

5) Finamente vemos los datos obtenidos en la prueba.

```
Madrid#show rtr distribution-statistics 2
        Captured Statistics
Entry     = Entry number
StartT    = Start time of entry (hundredths of seconds)
Pth       = Path index
Hop       = Hop in path index
Dst       = Time distribution index
Comps     = Operations completed
OvrTh     = Operations completed over thresholds
SumCmp    = Sum of RTT (milliseconds)
SumCmp2L  = Sum of RTT squared low 32 bits (milliseconds)
SumCmp2H  = Sum of RTT squared high 32 bits (milliseconds)
TMax      = RTT maximum (milliseconds)
TMin      = RTT minimum (milliseconds)

Entry StartT      Pth Hop Dst Comps       OvrTh       SumCmp      SumCmp2L    SumCmp2
H    TMax        TMin
2      8582709     1   1   1   60          0           3676        332080      0
     332         20

Madrid#
```

Figura 5.11: Resultados obtenidos en la segunda prueba.

La Latencia producida durante la prueba se mide utilizando los valores de RTT mostrados en la figura 5.11. [20]

	Operación	Resultado (mseg)
Latencia TCP(promedio)	SumCmp/ Comps	61
Dispersión	SumCmp2L/ Comps	5534
Desviación Estándar	$\sqrt{5534 - 61^2}$	42
» La Latencia TCP está dentro del rango de 61 $\pm$ 42 mseg.		

Tabla 5.3: Cálculo de Latencia de paquetes TCP.

5.1.3.3 Prueba 3: Latencia de UDP

PROCEDIMIENTO: Se enviarán **paquetes UDP** cada 10 segundos al puerto 888 del router Valencia desde el router Madrid. El router Madrid se encargará de recolectar y realizar estadísticas de los valores de Latencia recogidos en el enlace debido a paquetes UDP. La duración de la prueba es de **60 min**.

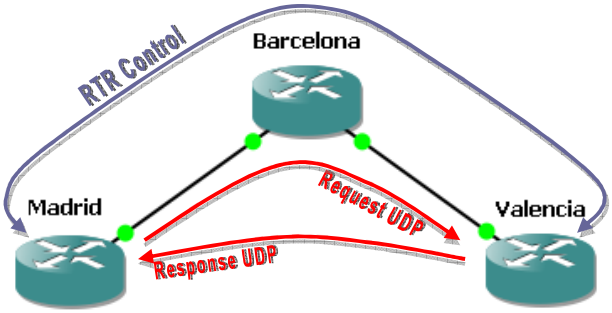


Figura 5.12: Descripción del procedimiento de la prueba.

Para la configuración de esta prueba se tomaron en consideración los siguientes pasos: [19]

1) Habilitar RTR en el router destino, como en la figura 5.7, para activar temporalmente el servicio UDP.

2) Definir una prueba tipo *"udpEcho"* en el router Madrid para medir la Latencia producida por paquetes UDP en el enlace durante la prueba. Al igual que en la prueba anterior, el router origen envía, por defecto, paquetes de control antes de enviar los paquetes UDP para habilitar los puertos en el destino. No olvidar indicar la dirección IP del router destino, el puerto al cual se quiere realizar la conexión, la frecuencia de envío de paquetes y la duración de la prueba. Ver figura 5.12.

```
Madrid(config)#rtr 3
Madrid(config-rtr)#type udpEcho dest-ipaddr 10.10.10.6 dest-port 888
Madrid(config-rtr-udp)#frequency 10
Madrid(config-rtr-udp)#exit
Madrid(config)#rtr schedule 3 life 3600 start-time now
```

Figura 5.13: Configuración de la tercera prueba.

3) Observamos un reporte de los datos obtenidos.

```
Madrid#show rtr distribution-statistics 3
        Captured Statistics
Entry     = Entry number
StartT    = Start time of entry (hundredths of seconds)
Pth       = Path index
Hop       = Hop in path index
Dst       = Time distribution index
Comps     = Operations completed
OvrTh     = Operations completed over thresholds
SumCmp    = Sum of RTT (milliseconds)
SumCmp2L  = Sum of RTT squared low 32 bits (milliseconds)
SumCmp2H  = Sum of RTT squared high 32 bits (milliseconds)
TMax      = RTT maximum (milliseconds)
TMin      = RTT minimum (milliseconds)

Entry StartT      Pth Hop Dst Comps       OvrTh       SumCmp      SumCmp2L    SumCmp2
H    TMax        TMin
3     18617367    1   1   1   360         0           17993       1245643     0
     446         1

Madrid#_
```

Figura 5.14: Resultados obtenidos en la tercera prueba.

La Latencia de la prueba se calcula usando los valores de RTT mostrados en la figura 5.14. [20]

	Operación	Resultado (mseg)
Latencia UDP(promedio)	SumCmp/ Comps	50
Dispersión	SumCmp2L/ Comps	3460
Desviación Estándar	$\sqrt{3460 - 50^2}$	31
» La Latencia UDP está dentro del rango de 60 $\pm$ 34 mseg.		

Tabla 5.4: Cálculo de Latencia de paquetes UCP.

5.1.3.4 Prueba 4: Throughput en la red

PROCEDIMIENTO: Se enviará la mayor cantidad de **datagramas UDP de longitud variable** desde el host cliente JPerf al host servidor JPerf de forma continua, durante **60 segundos,** usando el programa Iperf. La velocidad de transmisión de los datagramas irá aumentando hasta que encontremos la máxima cantidad de tramas recibidas en el destino sin que se produzcan perdidas. Las longitudes de tramas serán las mismas que las usadas en el capítulo 4: 64, 128, 256, 512, 1024, 1280 y 1518 octetos.

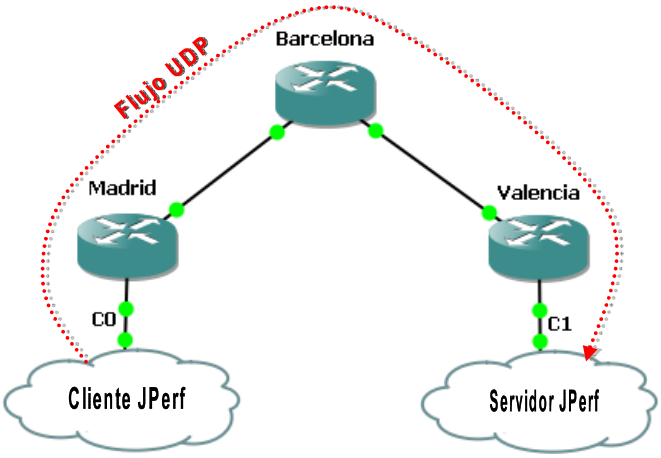


Figura 5.15: Descripción del procedimiento de la prueba.

Los datos obtenidos se muestran en el siguiente cuadro.

Longitud de trama (octetos)	64	128	256	512	1024	1280	1512
Velocidad de Tx (Mbps)	0,015	0,05	0,11	0,2	0,28	0,29	0,31
Velocidad de Rx (Mbps)	0,015	0,05	0,11	0,2	0,28	0,29	0,31
Tramas Transmitidas	5092	4357	3857	3193	2140	1748	1584
Tramas Recibidas	5092	4357	3857	3193	2140	1748	1584
Tramas Perdidas	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
Tramas Recibidas (pps)	**84,8**	**72,6**	**64,2**	**53,2**	**35,6**	**29,3**	**26,4**

Tabla 5.5: Datos de Throughput en una red puramente virtual.

5.1.3.5 Prueba 5: Jitter en la red

PROCEDIMIENTO: El host cliente JPerf enviará **flujos de paquetes UDP de diferentes longitudes** al host servidor JPerf de forma constante durante **60 segundos** a las velocidades de Throughput encontradas en la prueba anterior. Las longitudes de tramas serán de 64, 128, 256, 512, 1024, 1280 y 1518 octetos.

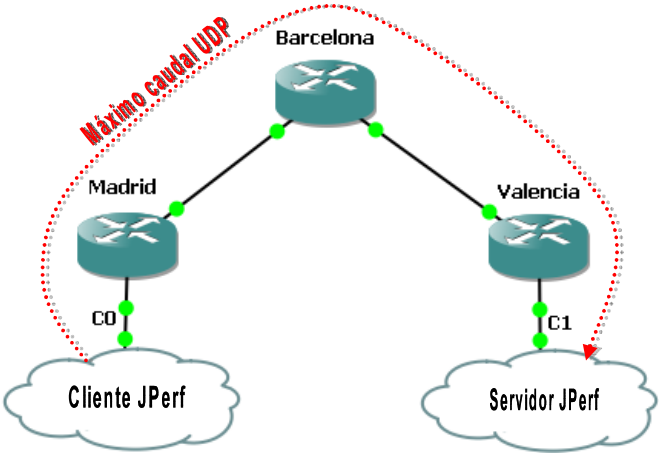


Figura 5.16: Descripción del procedimiento de la prueba.

La tabla 5.6 muestra los valores promedio de Jitter obtenido en la red tras el envío de datos a la velocidad máxima de Throughput.

Longitud de trama (octetos)	64	128	256	512	1024	1280	1512
Velocidad de Tx (Mbps)	0,015	0,05	0,11	0,2	0,28	0,29	0,31
Tramas Transmitidas	5092	4357	3857	3193	2140	1748	1584
Tramas Perdidas	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
Jitter (ms)	**15,6**	**17,062**	**19,416**	**22,907**	**35,416**	**40,188**	**43,414**

Tabla 5.6: Datos de Jitter en una red puramente virtual.

5.2 Pruebas con Comunicaciones Externas

5.2.1 Objetivo

Este apartado tiene el objetivo de mostrar las comunicaciones que se producen en escenarios que no son completamente virtuales, es decir, que contienen equipos reales. Del mismo modo, mediremos el funcionamiento de la red simulada para contrastar los resultados obtenidos con el escenario anteriormente mostrado.

5.2.2 Escenario de prueba

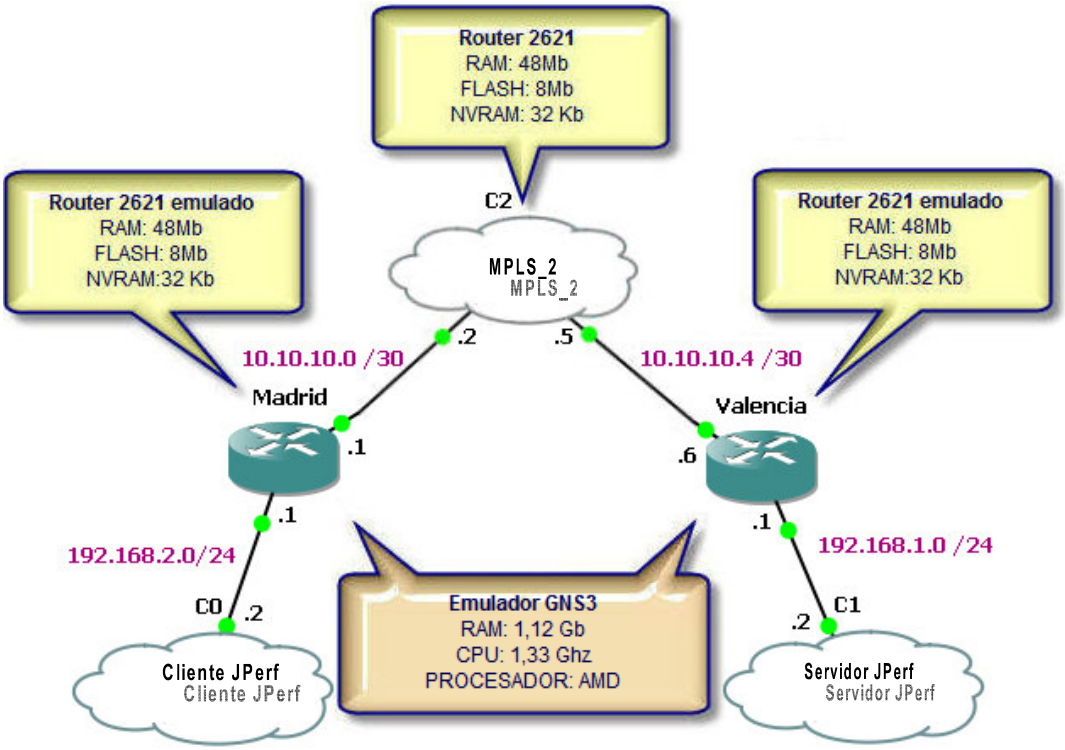


Figura 5.17: Escenario con un router real externo.

Hemos reutilizado el esquema del escenario analizado anteriormente para los routers emulados Madrid y Valencia y para los hosts cliente y Servidor JPerf, del mismo modo, el direccionamiento de red y el IOS usado es el mismo. Los cambios radican principalmente en la incorporación de nuevos enlaces reales asignados al router MPLS_2 que reemplaza al router Barcelona.

A continuación se menciona, con más detalle, todos los elementos involucrados en este escenario y sus características:

	Nombre	Tipo	Memoria	CPU	Observaciones
Router	Madrid	Virtual	RAM: 48Mb FLASH: 8Mb NVRAM:120 Kb	-	Router 2621 emulado
	MPLS_2	Real	RAM: 48Mb FLASH: 8Mb NVRAM:120 Kb	-	Router 2621
	Valencia	Virtual	RAM: 48Mb FLASH: 8Mb NVRAM:120 Kb	-	Router 2621 emulado
PC	Simulador GNS3	Real	RAM: 1,12Gb	1,30 Ghz	Con procesador AMD
	Cliente JPerf	Real	RAM: 1 Gb	1,66 Ghz	Con procesador Intel(R)
	Servidor JPerf	Real	RAM: 1 Gb	2,80 Ghz	Con procesador Intel
Interfaces	Fastethernet 0/0	Virtual	IP: 10.10.10.1		Perteneciente a Madrid
	Fastethernet 0/1	Virtual	IP: 192.168.2.1		Perteneciente a Madrid
	Fastethernet 0/0	Real	IP: 10.10.10.2		Perteneciente a MPLS_2
	Fastethernet 0/1	Real	IP: 10.10.10.5		Perteneciente a MPLS_2

	Fastethernet 0/0	Virtual	IP: 10.10.10.6	Perteneciente a Valencia
	Fastethernet 0/1	Real	IP: 192.168.1.1	Perteneciente a Valencia
	Adaptador de red 1	Real	IP: 192.168.1.2	Perteneciente a Servidor JPerf
	Adaptador de red 2	Real	IP: 192.168.2.2	Perteneciente a Cliente JPerf

Tabla 5.7: Elementos que aparecen en el escenario.

5.2.2.1 Configuración del escenario

La incorporación del router MPLS_2 al escenario se hizo siguiendo los comandos que se muestran a continuación:

```
MPLS_2#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
MPLS_2(config)#interface fastethernet 0/0
MPLS_2(config-if)#ip address 10.10.10.2 255.255.255.252
MPLS_2(config-if)#no shutdown
MPLS_2(config-if)#interface fastethernet 0/1
MPLS_2(config-if)#ip address 10.10.10.5 255.255.255.252
MPLS_2(config-if)#no shutdown
MPLS_2(config-if)#exit
MPLS_2(config)#
```

Figura 5.18: Configuración de interfaces en el router MPLS_2. [12]

5.2.3 Medición y tratamiento de datos

5.2.3.1 Prueba 1: Latencia de ICMP

PROCEDIMIENTO: Se sigue el mismo procedimiento que para la realización de la prueba del apartado 5.1.3.1.

La tabla 5.8 muestra los nuevos valores de latencia ICMP obtenidos en una red simulada que contiene un router real externo, cuando se varía la longitud de las tramas enviadas.

Longitud de trama (octetos)	64	128	256	512	1024	1280	1512
Round-trip min (ms)	31	31	32	34	42	48	52
Round-trip avg (ms)	**38**	**39**	**39**	**41**	**55**	**59**	**64**
Round-trip max (ms)	70	74	76	72	94	93	97

Tabla 5.8: Datos de latencia ICMP para la red puramente virtual.

5.2.3.2 Prueba 2: Latencia de TCP

PROCEDIMIENTO: Se sigue el mismo procedimiento que para la realización de la prueba del apartado 5.1.3.2.

La configuración de esta prueba se realiza con los comandos ilustrados en las figuras de la 5.7, 5.8, 5.9 y 5.10. Después del término de la prueba se obtienen los siguientes resultados:

```
Madrid#show rtr distribution-statistics 2
        Captured Statistics
Entry     = Entry number
StartT    = Start time of entry (hundredths of seconds)
Pth       = Path index
Hop       = Hop in path index
Dst       = Time distribution index
Comps     = Operations completed
OvrTh     = Operations completed over thresholds
SumCmp    = Sum of RTT (milliseconds)
SumCmp2L  = Sum of RTT squared low 32 bits (milliseconds)
SumCmp2H  = Sum of RTT squared high 32 bits (milliseconds)
TMax      = RTT maximum (milliseconds)
TMin      = RTT minimum (milliseconds)

Entry StartT      Pth Hop Dst Comps       OvrTh       SumCmp      SumCmp2L    SumCmp2
H   TMax        TMin
2     22738       1   1   1   60          0           2136        190592      0
    288         12

Madrid#
```

Figura 5.19: Resultados obtenidos en la segunda prueba.

El cálculo de los datos estadísticos de la Latencia TCP se muestra en la tabla 5.9 que se ve a continuación. [20]

	Operación	Resultado (mseg)
Latencia TCP(promedio)	SumCmp/ Comps	36
Dispersión	SumCmp2L/ Comps	3176
Desviación Estándar	$\sqrt{3176 - 249^2}$	43
» La Latencia TCP está dentro del rango de 36 $\pm$ 43 mseg.		

Tabla 5.9: Cálculo de Latencia de paquetes TCP.

5.2.3.3 Prueba 3: Latencia de UDP

PROCEDIMIENTO: Se sigue el mismo procedimiento que para la realización de la prueba del apartado 5.1.3.3.
Del mismo modo, para la configuración de esta prueba se siguen los siguientes pasos detallados el apartado 5.1.3.3 y los comandos ilustrados en las figuras 5.7 y 5.13. Con lo cual obtenemos los siguientes resultados:

```
Madrid#show rtr distribution-statistics 3
        Captured Statistics
Entry     = Entry number
StartT    = Start time of entry (hundredths of seconds)
Pth       = Path index
Hop       = Hop in path index
Dst       = Time distribution index
Comps     = Operations completed
OvrTh     = Operations completed over thresholds
SumCmp    = Sum of RTT (milliseconds)
SumCmp2L  = Sum of RTT squared low 32 bits (milliseconds)
SumCmp2H  = Sum of RTT squared high 32 bits (milliseconds)
TMax      = RTT maximum (milliseconds)
TMin      = RTT minimum (milliseconds)

Entry StartT      Pth Hop Dst Comps       OvrTh       SumCmp      SumCmp2L    SumCmp2
H   TMax        TMin
3     8016        1   1   1   360         0           7101        184021      0
    156         0

Madrid#_
```

Figura 5.20: Resultados obtenidos en la tercera prueba.

La Latencia de la prueba se calcula usando los valores de RTT mostrados en la figura anterior. [20]

	Operación	Resultado (mseg)
Latencia UDP(promedio)	SumCmp/ Comps	20
Dispersión	Sum2Cmp/ Comps	513
Desviación Estándar	$\sqrt{513 - 20^2}$	11
» La latencia UDP está dentro del rango de 20 $\pm$ 11 mseg.		

Tabla 5.10: Cálculo de Latencia de paquetes UCP.

5.2.3.4 Prueba 4: Throughput en la red

PROCEDIMIENTO: Se sigue el mismo procedimiento que para la realización de la prueba del apartado 5.1.3.5.

Los datos de Throughput obtenidos en esta nueva red simulada que cuenta con un router externo real, se muestran en el siguiente cuadro.

Longitud de trama (octetos)	64	128	256	512	1024	1280	1512
Velocidad de Tx (Mbps)	0,024	0,08	0,18	0,3	0,45	0,49	0,52
Velocidad de Rx (Mbps)	0,024	0,08	0,18	0,3	0,45	0,49	0,52
Tramas Transmitidas	7664	6851	6248	4756	3439	2971	2655
Tramas Recibidas	7664	6851	6248	4756	3439	2971	2655
Tramas Perdidas	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
Tramas Recibidas (pps)	**127,7**	**114,2**	**104,1**	**79,2**	**57,3**	**49,5**	**44,2**

Tabla 5.11: Datos de Throughput en una red con router real externo.

5.2.3.5 Prueba 5: Jitter en la red

PROCEDIMIENTO: Se sigue el mismo procedimiento que para la realización de la prueba del apartado 5.1.3.6.

La tabla 5.12 muestra el valor de Jitter obtenido en la red de prueba tras el envío de datos a la velocidad máxima de Throughput.

Longitud de trama (octetos)	64	128	256	512	1024	1280	1512
Velocidad de Tx (Mbps)	0,015	0,05	0,11	0,2	0,28	0,29	0,31
Tramas Transmitidas	5092	4357	3857	3193	2140	1748	1584
Tramas Perdidas	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
Jitter (ms)	**15,6**	**17,062**	**19,416**	**22,907**	**35,416**	**40,188**	**43,414**

Tabla 5.12: Datos de Jitter en una red puramente virtual.

5.3 Captura de tráfico

En este apartado realizaremos la captura de los datos que pasan por la red en el momento de realizar cada una de las pruebas descritas en el apartado 5.1, en un escenario formado enteramente por routers emulados usando GNS3. La captura se realizará gracias a la capacidad de *Wireshark* y GNS3 de mostrar el tráfico que pasa por los enlaces virtuales en cada instante de tiempo.

La captura servirá para mostrar detalladamente la estructura en las tramas características de los protocolos usados en cada una de las pruebas, así como también el uso de otros protocolos adicionales durante la comunicación de los dispositivos de red emulados.

5.3.1 Prueba 1: Latencia de ICMP

En las figuras siguientes se puede apreciar a detalle la estructura de una trama del protocolo ICMP usado para la toma de medidas en la primera prueba, además se observa cómo se lleva a cabo la organización temporal de las tramas de este protocolo a la hora de viajar por la red virtual en compañía de tramas de otros protocolos.

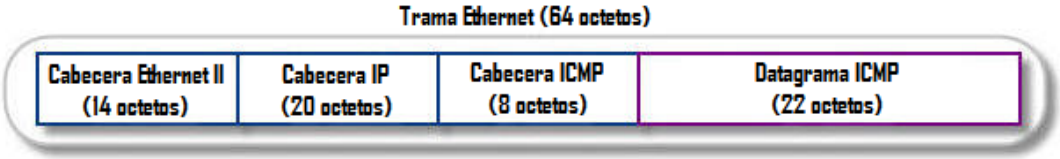


Figura 5.21: Trama ICMP enviada en la primera prueba.

No.	Time	Source	Destination	Protocol	Info
29	25.734000	192.168.2.2	192.168.1.2	ICMP	Echo (ping) request
30	25.734000	192.168.1.2	192.168.2.2	ICMP	Echo (ping) reply
31	26.844000	192.168.2.2	192.168.1.2	ICMP	Echo (ping) request
32	26.844000	192.168.1.2	192.168.2.2	ICMP	Echo (ping) reply
33	27.953000	192.168.2.2	192.168.1.2	ICMP	Echo (ping) request
34	27.953000	192.168.1.2	192.168.2.2	ICMP	Echo (ping) reply
35	29.063000	192.168.2.2	192.168.1.2	ICMP	Echo (ping) request

```
⊞ Frame 31 (64 bytes on wire, 64 bytes captured)
⊟ Ethernet II, Src: c8:03:16:bc:00:01 (c8:03:16:bc:00:01), Dst: 02:00:4c:4f:4f:50 (02:00:4c:4f:4f:50)
  ⊞ Destination: 02:00:4c:4f:4f:50 (02:00:4c:4f:4f:50)
  ⊞ Source: c8:03:16:bc:00:01 (c8:03:16:bc:00:01)
    Type: IP (0x0800)
⊟ Internet Protocol, Src: 192.168.2.2 (192.168.2.2), Dst: 192.168.1.2 (192.168.1.2)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 50
    Identification: 0xda8d (55949)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 124
    Protocol: ICMP (0x01)
  ⊞ Header checksum: 0xdfe8 [correct]
    Source: 192.168.2.2 (192.168.2.2)
    Destination: 192.168.1.2 (192.168.1.2)
⊟ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0 ()
    Checksum: 0xfe4e [correct]
    Identifier: 0x0200
    Sequence number: 23048 (0x5a08)
  ⊞ Data (22 bytes)
```

Figura 5.22: Estructura de tráfico ICMP.

La figura 5.23 muestra como se envían tramas que contienen al **protocolo ICMP** (**I**nternet **C**ontrol **M**essage **P**rotocol) cuando se aplica el comando "ping" en la red durante el tiempo de la toma de medidas, en este sentido, vemos claramente el envío de los dos tipos de mensajes característicos del protocolo ICMP, como son *"Echo request"* y *"Echo reply"* desde el host origen (192.168.2.2) y desde el host destino (192.168.1.2) respectivamente. Por otro lado, también se aprecian tramas propias del **protocolo RIP** para intercambiar información de enrutamiento entre los routers del segmento de red, las tramas tienen como dirección de destino una *dirección multicast* asignada exclusivamente a este protocolo que designa a todos los routers del segmento de red. Otro protocolo que aparece continuamente en la captura es el **protocolo LOOP** (**C**onfiguration **T**esting **P**rotocol) que nos permite determinar si una interfaz está activa o no, este protocolo forma parte del estándar de Ethernet y básicamente se puede comparar a un "ping" pero que funciona en la capa 2. [11] [29]

879	1256.500000	192.168.1.2	192.168.2.2	ICMP	Echo (ping) reply
880	1257.594000	192.168.2.2	192.168.1.2	ICMP	Echo (ping) request
881	1257.594000	192.168.1.2	192.168.2.2	ICMP	Echo (ping) reply
882	1258.703000	192.168.2.2	192.168.1.2	ICMP	Echo (ping) request
883	1258.703000	192.168.1.2	192.168.2.2	ICMP	Echo (ping) reply
884	1259.812000	192.168.2.2	192.168.1.2	ICMP	Echo (ping) request
885	1259.812000	192.168.1.2	192.168.2.2	ICMP	Echo (ping) reply
886	1260.969000	192.168.2.2	192.168.1.2	ICMP	Echo (ping) request
887	1260.969000	192.168.1.2	192.168.2.2	ICMP	Echo (ping) reply
888	1261.359000	192.168.1.1	224.0.0.9	RIPv2	Response
889	1262.047000	192.168.2.2	192.168.1.2	ICMP	Echo (ping) request
890	1262.047000	192.168.1.2	192.168.2.2	ICMP	Echo (ping) reply
891	1262.719000	c8:03:16:bc:00:01	c8:03:16:bc:00:01	LOOP	Reply
892	1263.156000	192.168.2.2	192.168.1.2	ICMP	Echo (ping) request
893	1263.156000	192.168.1.2	192.168.2.2	ICMP	Echo (ping) reply
894	1264.265000	192.168.2.2	192.168.1.2	ICMP	Echo (ping) request
895	1264.265000	192.168.1.2	192.168.2.2	ICMP	Echo (ping) reply
896	1265.375000	192.168.2.2	192.168.1.2	ICMP	Echo (ping) request
897	1265.375000	192.168.1.2	192.168.2.2	ICMP	Echo (ping) reply
898	1266.484000	192.168.2.2	192.168.1.2	ICMP	Echo (ping) request
899	1266.484000	192.168.1.2	192.168.2.2	ICMP	Echo (ping) reply
900	1267.594000	192.168.2.2	192.168.1.2	ICMP	Echo (ping) request
901	1267.594000	192.168.1.2	192.168.2.2	ICMP	Echo (ping) reply
902	1268.719000	192.168.2.2	192.168.1.2	ICMP	Echo (ping) request
903	1268.719000	192.168.1.2	192.168.2.2	ICMP	Echo (ping) reply

RIPV2

LOOP

ICMP

Figura 5.23: Captura de tráfico entre el router Madrid y el host *Cliente Jperf*.

5.3.2 Prueba 2: Latencia de TCP

Las siguientes figuras muestran la estructura de una trama del protocolo TCP usado en la segunda prueba, del mismo modo, se observa el proceso de conexión y finalización propio de este protocolo y como se comunican los dispositivos de red emulados usando otros protocolos.

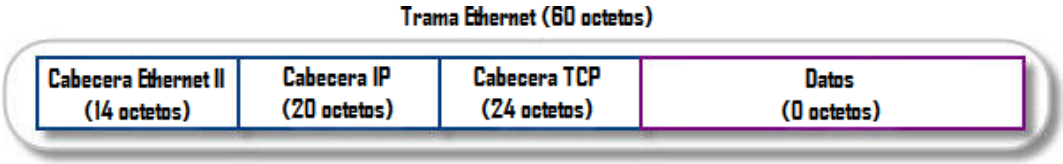


Figura 5.24: Trama de señalización TCP enviada en la segunda prueba (SYN).

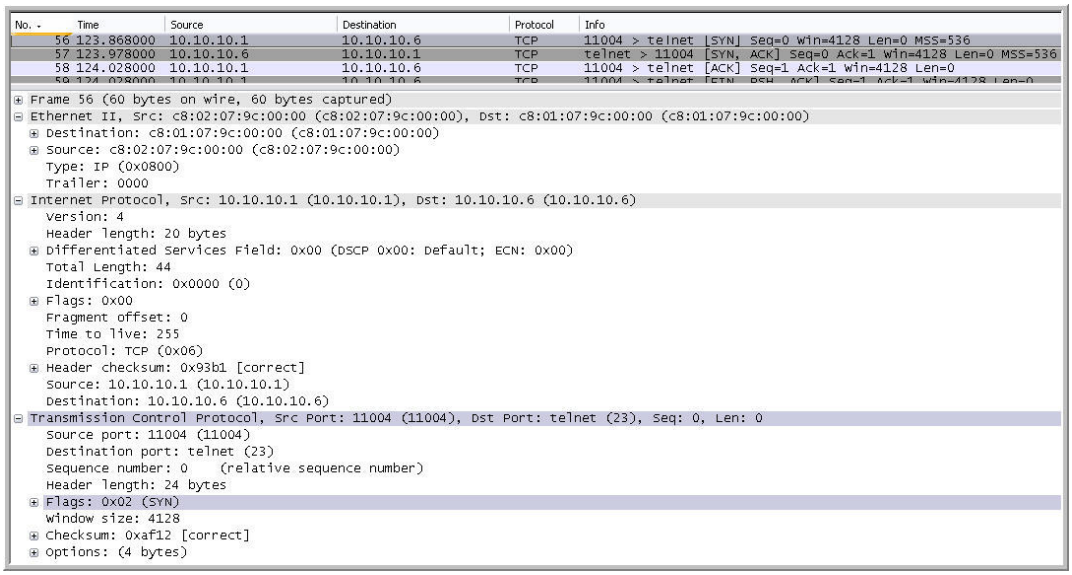


Figura 5.25: Estructura de tráfico TCP.

Como es sabido, el **protocolo TCP** (**T**ransmission **C**ontrol **P**rotocol) requiere de una conexión antes del envío de paquetes y garantiza una recepción sin errores y ordenada. En la figura 5.26 se muestran los mecanismos de establecimiento y fin de conexión llevadas a cabo por el protocolo TCP; el establecimiento de la conexión empieza con el envío de un mensaje *(SYN)* desde un puerto arbitrario del router Madrid al puerto Telnet (23) del router Valencia, luego éste le responde con un mensaje de acreditación *(SYN/ACK)* y finalmente Madrid confirma el establecimiento de la conexión con un mensaje *(ACK)*. El siguiente mensaje que envía Madrid comienza el proceso de fin de conexión *(FIN)*, dado que en TCP la conexión se termina desde cada lado de la conexión independientemente, después de recibir una confirmación (ACK) desde el router Valencia, éste también envía un mensaje (FIN) que finalmente es acreditado por el router Madrid. [11]

Adicionalmente de las tramas con información de protocolos ya citados en la captura de la figura 5.23, como son RIPv2 y LOOP, en esta nueva captura aparece el **protocolo CDP** (**C**isco **D**iscovery **P**rotocol), el cual es propio de cisco, es decir, no encapsulado en IP, que permite a los dispositivos de red conocer qué otros dispositivos CISCO tienen como vecinos. Se aprecia que este protocolo utiliza las direcciones MAC para denotar que interfaz (extremo del enlace) manda los paquetes CDP. [12]

1	0.000000	c8:02:04:98:00:00	CDP/VTP/DTP/PAgP/UDLD	CDP	Device ID: Madrid Port ID: FastEthernet0/0
2	0.043000	c8:01:04:98:00:00	CDP/VTP/DTP/PAgP/UDLD	CDP	Device ID: Barcelona Port ID: FastEthernet0/0
3	1.707000	c8:02:04:98:00:00	c8:02:04:98:00:00	LOOP	Reply
4	2.228000	c8:01:04:98:00:00	c8:01:04:98:00:00	LOOP	Reply
5	6.435000	10.10.10.1	224.0.0.9	RIPv2	Response
6	11.701000	c8:02:04:98:00:00	c8:02:04:98:00:00	LOOP	Reply
7	12.244000	c8:01:04:98:00:00	c8:01:04:98:00:00	LOOP	Reply
8	21.701000	c8:02:04:98:00:00	c8:02:04:98:00:00	LOOP	Reply
9	21.926000	10.10.10.2	224.0.0.9	RIPv2	Response
10	22.220000	c8:01:04:98:00:00	c8:01:04:98:00:00	LOOP	Reply
11	30.105000	10.10.10.1	10.10.10.6	UDP	Source port: 54722 Destination port: sns-quote
12	30.120000	10.10.10.6	10.10.10.1	UDP	Source port: sns-quote Destination port: 54722
13	30.153000	10.10.10.1	10.10.10.6	TCP	11054 > telnet [SYN] Seq=0 Win=4128 Len=0 MSS=536
14	30.168000	10.10.10.6	10.10.10.1	TCP	telnet > 11054 [SYN, ACK] Seq=0 Ack=1 Win=4128 Len=0 MSS=536
15	30.184000	10.10.10.1	10.10.10.6	TCP	11054 > telnet [ACK] Seq=1 Ack=1 Win=4128 Len=0
16	30.191000	10.10.10.1	10.10.10.6	TCP	11054 > telnet [FIN, PSH, ACK] Seq=1 Ack=1 Win=4128 Len=0
17	30.211000	10.10.10.6	10.10.10.1	TCP	telnet > 11054 [ACK] Seq=1 Ack=2 Win=4128 Len=0
18	30.216000	10.10.10.6	10.10.10.1	TCP	telnet > 11054 [FIN, PSH, ACK] Seq=1 Ack=2 Win=4128 Len=0
19	30.222000	10.10.10.1	10.10.10.6	TCP	11054 > telnet [ACK] Seq=2 Ack=2 Win=4128 Len=0
20	31.695000	c8:02:04:98:00:00	c8:02:04:98:00:00	LOOP	Reply
21	32.231000	c8:01:04:98:00:00	c8:01:04:98:00:00	LOOP	Reply
22	32.589000	10.10.10.1	224.0.0.9	RIPv2	Response
23	41.676000	c8:02:04:98:00:00	c8:02:04:98:00:00	LOOP	Reply
24	42.235000	c8:01:04:98:00:00	c8:01:04:98:00:00	LOOP	Reply
25	47.666000	10.10.10.2	224.0.0.9	RIPv2	Response

CDP
RIPV2
TCP
LOOP

Figura 5.26: Captura de tráfico en el enlace desde el router Madrid a Barcelona.

5.3.3 Prueba 3: Latencia de UDP

Este apartado muestra la estructura y la organización temporal de las tramas del protocolo UDP, así como también como viajan, por un enlace virtual, junto con otras tramas de diferentes protocolos durante la duración de la prueba.

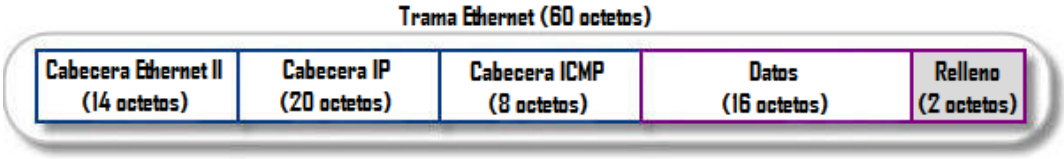


Figura 5.27: Trama UDP enviada en la tercera prueba.

No.	Time	Source	Destination	Protocol	Info
41	75.989000	10.10.10.1	10.10.10.6	UDP	Source port: 59088 Destination port: sns-quote
42	76.049000	10.10.10.6	10.10.10.1	UDP	Source port: sns-quote Destination port: 59088
43	76.069000	10.10.10.1	10.10.10.6	UDP	Source port: 50877 Destination port: accessbuilder
44	76.109000	10.10.10.6	10.10.10.1	UDP	Source port: accessbuilder Destination port: 50877
45	83.450000	c8:01:0d:08:00:00	c8:01:0d:08:00:00	LOOP	Reply
46	83.991000	c8:02:0d:08:00:00	c8:02:0d:08:00:00	LOOP	Reply
47	86.024000	10.10.10.1	10.10.10.6	UDP	Source port: 53114 Destination port: sns-quote

```
⊞ Frame 43 (60 bytes on wire, 60 bytes captured)
⊟ Ethernet II, Src: c8:02:0d:08:00:00 (c8:02:0d:08:00:00), Dst: c8:01:0d:08:00:00 (c8:01:0d:08:00:00)
  ⊞ Destination: c8:01:0d:08:00:00 (c8:01:0d:08:00:00)
  ⊞ Source: c8:02:0d:08:00:00 (c8:02:0d:08:00:00)
    Type: IP (0x0800)
    Trailer: 0000
⊟ Internet Protocol, Src: 10.10.10.1 (10.10.10.1), Dst: 10.10.10.6 (10.10.10.6)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 44
    Identification: 0x0000 (0)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 255
    Protocol: UDP (0x11)
  ⊞ Header checksum: 0x93a6 [correct]
    Source: 10.10.10.1 (10.10.10.1)
    Destination: 10.10.10.6 (10.10.10.6)
⊟ User Datagram Protocol, Src Port: 50877 (50877), Dst Port: accessbuilder (888)
    Source port: 50877 (50877)
    Destination port: accessbuilder (888)
    Length: 24
  ⊞ Checksum: 0xbc14 [correct]
⊞ Data (16 bytes)
```

Figura 5.28: Estructura de tráfico UDP.

Esta tercera prueba se basa en el intercambio de tramas que usan el **protocolo UDP** (**U**ser **D**atagrama **P**rotocol), por lo tanto, no se establece una conexión previa ni se realiza control de flujo. UDP es un protocolo ubicado en la capa de transporte, por lo que trabaja principalmente con puertos, en este sentido, como los puertos origen carecen de importancia, observamos que el puerto origen del router Madrid tiene una numeración que pertenece a los puertos temporales mientras que el puerto destino es siempre el puerto 888 que tiene el nombre de *"accessbuilder".* Respecto a las tramas

de diferentes protocolos que aparecen en la figura 5.29 son las mismas que las ya citadas en capturas anteriores. [11]

22	44.003000	c8:02:0d:08:00:00	c8:02:0d:08:00:00	LOOP	Reply
23	51.144000	c8:01:0d:08:00:00	c8:01:0d:08:00:00	LOOP	Reply
24	54.008000	c8:02:0d:08:00:00	c8:02:0d:08:00:00	LOOP	Reply
25	56.742000	10.10.10.1	10.10.10.6	UDP	Source port: 54076 Destination port: sns-quote
26	56.902000	10.10.10.6	10.10.10.1	UDP	Source port: sns-quote Destination port: 54076
27	57.062000	10.10.10.1	10.10.10.6	UDP	Source port: 50709 Destination port: accessbuilder
28	57.122000	10.10.10.6	10.10.10.1	UDP	Source port: accessbuilder Destination port: 50709
29	58.284000	10.10.10.1	224.0.0.9	RIPv2	Response
30	60.197000	c8:02:0d:08:00:00	CDP/VTP/DTP/PAgP/UDLD	CDP	Device ID: Madrid Port ID: FastEthernet0/0
31	61.148000	c8:01:0d:08:00:00	c8:01:0d:08:00:00	LOOP	Reply
32	61.148000	c8:01:0d:08:00:00	CDP/VTP/DTP/PAgP/UDLD	CDP	Device ID: Barcelona Port ID: FastEthernet0/0
33	64.012000	c8:02:0d:08:00:00	c8:02:0d:08:00:00	LOOP	Reply
34	66.005000	10.10.10.1	10.10.10.6	UDP	Source port: 57945 Destination port: sns-quote
35	66.055000	10.10.10.6	10.10.10.1	UDP	Source port: sns-quote Destination port: 57945
36	66.085000	10.10.10.1	10.10.10.6	UDP	Source port: 49997 Destination port: accessbuilder
37	66.115000	10.10.10.6	10.10.10.1	UDP	Source port: accessbuilder Destination port: 49997
38	69.390000	10.10.10.2	224.0.0.9	RIPv2	Response
39	71.222000	c8:01:0d:08:00:00	c8:01:0d:08:00:00	LOOP	Reply
40	75.288000	c8:02:0d:08:00:00	c8:02:0d:08:00:00	LOOP	Reply
41	75.989000	10.10.10.1	10.10.10.6	UDP	Source port: 59088 Destination port: sns-quote
42	76.049000	10.10.10.6	10.10.10.1	UDP	Source port: sns-quote Destination port: 59088
43	76.069000	10.10.10.1	10.10.10.6	UDP	Source port: 50877 Destination port: accessbuilder
44	76.109000	10.10.10.6	10.10.10.1	UDP	Source port: accessbuilder Destination port: 50877
45	83.450000	c8:01:0d:08:00:00	c8:01:0d:08:00:00	LOOP	Reply

LOOP
UDP
CDP
RIPV2

Figura 5.29: Captura de tráfico en el enlace desde el router Madrid a Barcelona.

5.3.4 Prueba 4: Throughput de la red

En las figuras siguientes se puede apreciar a detalle la estructura de la trama UDP usada para la medición de Throughput de la red, además se observa la interacción de los dispositivos simulados usando otros protocolos adiciones durante la duración de la prueba.

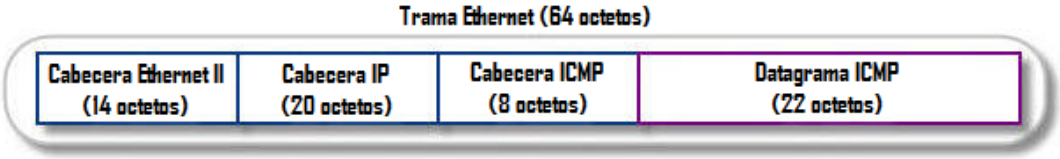


Figura 5.30: Trama más pequeña enviada para la medición del throughput.

No.	Time	Source	Destination	Protocol	Info
14	32.547000	192.168.2.2	192.168.1.2	UDP	Source port: 3546 Destination port: commplex-link
15	32.627000	192.168.2.2	192.168.1.2	UDP	Source port: 3546 Destination port: commplex-link
16	32.647000	192.168.2.2	192.168.1.2	UDP	Source port: 3546 Destination port: commplex-link
17	32.667000	192.168.2.2	192.168.1.2	UDP	Source port: 3546 Destination port: commplex-link
18	32.827000	192.168.2.2	192.168.1.2	UDP	Source port: 3546 Destination port: commplex-link
19	32.847000	192.168.2.2	192.168.1.2	UDP	Source port: 3546 Destination port: commplex-link
20	32.857000	192.168.2.2	192.168.1.2	UDP	Source port: 3546 Destination port: commplex-link

```
⊞ Frame 16 (64 bytes on wire, 64 bytes captured)
⊟ Ethernet II, Src: CompalEl_e5:70:89 (00:0f:b0:e5:70:89), Dst: c8:00:02:20:00:01 (c8:00:02:20:00:01)
  ⊞ Destination: c8:00:02:20:00:01 (c8:00:02:20:00:01)
  ⊞ Source: CompalEl_e5:70:89 (00:0f:b0:e5:70:89)
    Type: IP (0x0800)
⊟ Internet Protocol, Src: 192.168.2.2 (192.168.2.2), Dst: 192.168.1.2 (192.168.1.2)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 50
    Identification: 0x3b2e (15150)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 127
    Protocol: UDP (0x11)
  ⊞ Header checksum: 0x7c38 [correct]
    Source: 192.168.2.2 (192.168.2.2)
    Destination: 192.168.1.2 (192.168.1.2)
⊟ User Datagram Protocol, Src Port: 3546 (3546), Dst Port: commplex-link (5001)
    Source port: 3546 (3546)
    Destination port: commplex-link (5001)
    Length: 30
  ⊞ Checksum: 0x89f4 [correct]
⊞ Data (22 bytes)
```

Figura 5.31: Captura de tráfico.

La figura 5.32 muestra como, durante esta prueba, el host Cliente JPerf con IP 192.168.2.2 usa toda la capacidad posible del enlace virtual para enviar tramas UDP de forma continua desde un puerto origen aleatorio hasta un puerto conocido llamado

"commplex-link" del host Servidor JPerf con IP 192.168.1.1. Además vemos como también se intercambiaron datos pertenecientes a los protocolos CDP, RIPv2, LOOP y UDP.

2	0.050000	c8:01:0e:c4:00:00	c8:01:0e:c4:00:00	LOOP	Reply
3	4.917000	c8:02:0e:c4:00:00	CDP/VTP/DTP/PAgP/UDLD	CDP	Device ID: Madrid Port ID: FastEthernet0/0
4	5.158000	c8:01:0e:c4:00:00	CDP/VTP/DTP/PAgP/UDLD	CDP	Device ID: Barcelona Port ID: FastEthernet0/0
5	7.461000	10.10.10.1	224.0.0.9	RIPV2	Response
6	8.993000	10.10.10.2	224.0.0.9	RIPv2	Response
7	9.985000	c8:02:0e:c4:00:00	c8:02:0e:c4:00:00	LOOP	Reply
8	10.045000	c8:01:0e:c4:00:00	c8:01:0e:c4:00:00	LOOP	Reply
9	19.999000	c8:02:0e:c4:00:00	c8:02:0e:c4:00:00	LOOP	Reply
10	20.109000	c8:01:0e:c4:00:00	c8:01:0e:c4:00:00	LOOP	Reply
11	29.993000	c8:02:0e:c4:00:00	c8:02:0e:c4:00:00	LOOP	Reply
12	30.053000	c8:01:0e:c4:00:00	c8:01:0e:c4:00:00	LOOP	Reply
13	32.537000	192.168.2.2	192.168.1.2	UDP	Source port: 3546 Destination port: commplex-link
14	32.547000	192.168.2.2	192.168.1.2	UDP	Source port: 3546 Destination port: commplex-link
15	32.627000	192.168.2.2	192.168.1.2	UDP	Source port: 3546 Destination port: commplex-link
16	32.647000	192.168.2.2	192.168.1.2	UDP	Source port: 3546 Destination port: commplex-link
17	32.667000	192.168.2.2	192.168.1.2	UDP	Source port: 3546 Destination port: commplex-link
18	32.827000	192.168.2.2	192.168.1.2	UDP	Source port: 3546 Destination port: commplex-link
19	32.847000	192.168.2.2	192.168.1.2	UDP	Source port: 3546 Destination port: commplex-link
20	32.857000	192.168.2.2	192.168.1.2	UDP	Source port: 3546 Destination port: commplex-link
21	32.867000	192.168.2.2	192.168.1.2	UDP	Source port: 3546 Destination port: commplex-link
22	32.877000	192.168.2.2	192.168.1.2	UDP	Source port: 3546 Destination port: commplex-link
23	32.897000	192.168.2.2	192.168.1.2	UDP	Source port: 3546 Destination port: commplex-link
24	32.897000	192.168.2.2	192.168.1.2	UDP	Source port: 3546 Destination port: commplex-link
25	32.928000	192.168.2.2	192.168.1.2	UDP	Source port: 3546 Destination port: commplex-link
26	32.928000	192.168.2.2	192.168.1.2	UDP	Source port: 3546 Destination port: commplex-link

CDP
RIPV2
LOOP
UDP

Figura 5.32: Captura de tráfico en el enlace desde el router Madrid a Barcelona.

5.3.5 Prueba 5: Jitter en la red

Esta prueba realizada para medir el Jitter producido en la red formada por routers emulados y la anterior que nos indica el throughput capaz de alcanzase en un enlace de la misma red están ligadas entre sí ya que se realizan usando la herramienta JPerf, por lo tanto, la captura realizada en este apartado tiene las mismas características que las que se muestran en las figuras el apartado anterior.

5.4 Comparación de resultados

5.4.1 Prueba 1: Latencia de ICMP

La figura 5.33 muestra los valores de latencia obtenidos en la trasmisión de tramas de diferentes longitudes que llevan información del protocolo ICMP en los escenarios de las figuras 5.1 y 5.17.

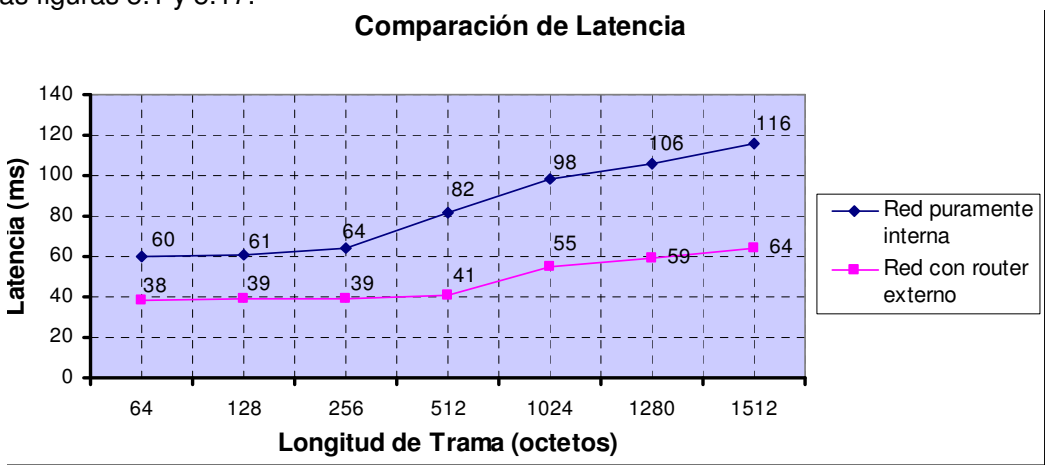


Figura 5.33: Comparación de resultados de la primera prueba.

5.4.2 Prueba 2: Latencia de TCP

La figura 5.34 muestra los valores de latencia promedios producidos en proceso de establecimiento de conexión propio del protocolo TCP en los escenarios de las figuras 5.1 y 5.17.

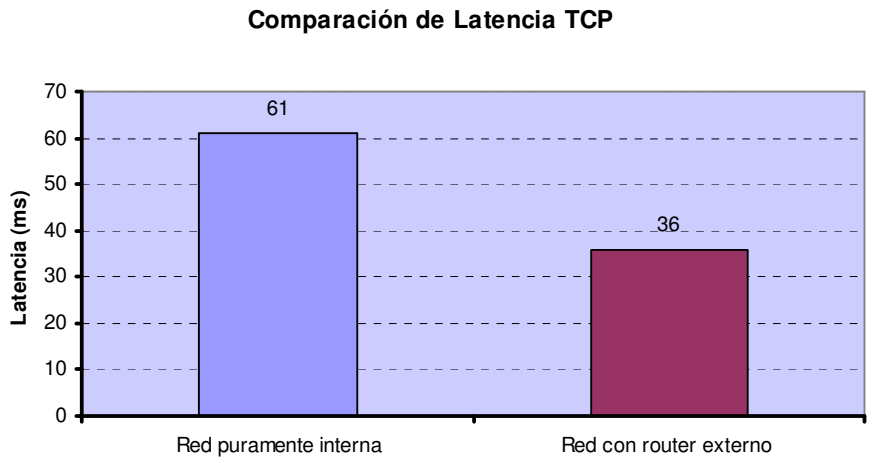


Figura 5.34: Comparación de resultados de la segunda prueba.

5.4.3 Prueba 3: Latencia de UDP

La figura 5.35 muestra los valores de latencia promedios producidos por la transmisión de tramas con el protocolo UDP por la red en los escenarios de las figuras 5.1 y 5.17.

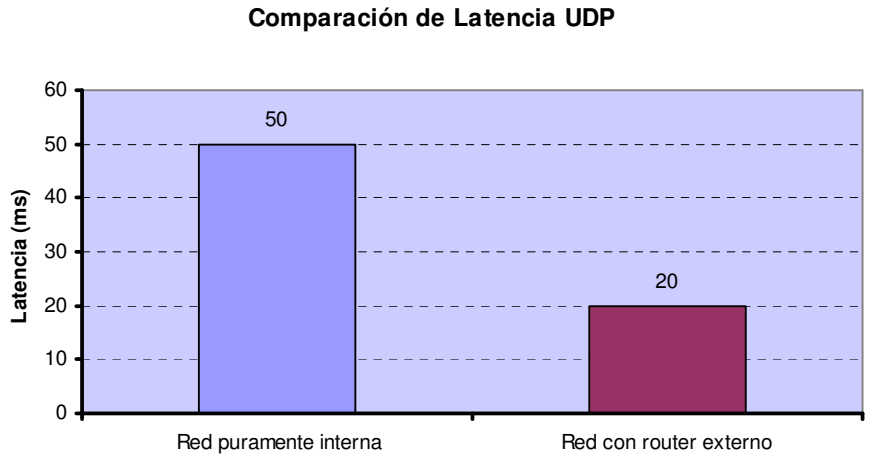


Figura 5.35: Comparación de resultados de la tercera prueba.

5.4.4 Prueba 4: Throughput en la red

La figura 5.36 muestra los valores de throughput máximo alcanzado en los escenarios de las figuras 5.1 y 5.17 para tramas de diferentes longitudes.

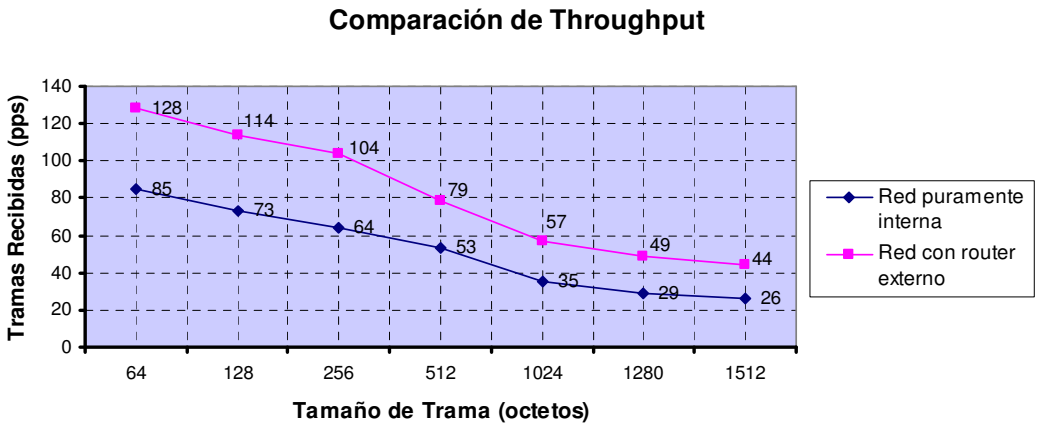


Figura 5.36: Comparación de resultados de la quinta prueba.

5.4.5 Prueba 5: Jitter en la red

La figura 5.37 muestra los valores de Jitter alcanzado en los escenarios de las figuras 5.1 y 5.17 para tramas de diferentes longitudes.

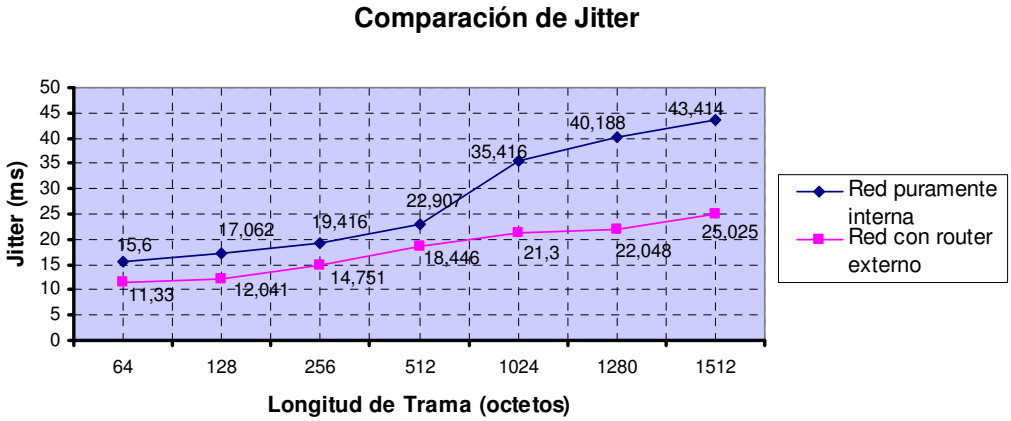


Figura 5.37: Comparación de resultados de la sexta prueba.

5.5 Conclusiones

La máxima Latencia recomendada por la **ITU G.114** en una conexión punto a punto (en un solo sentido) que se debe usar para transmisión de tráfico en tiempo real y de alta calidad, como la voz, es de ***150ms para redes LAN*** y de ***300ms para redes WAN***. Con respecto al Jitter, punto a punto, se recomienda que no se exceda de 1***00ms.*** [25]

En base a los datos en las pruebas realizadas en este capítulo y a los datos citados en el apartado anterior, hemos llegado a las siguientes conclusiones:

- La prueba de latencia ICMP en una red con 3 routers virtuales y bajo las características descritas, nos arroja valores por debajo a los recomendados (150ms), notar que la gráfica refleja los valores de RTT y no de latencia en un solo sentido, con lo cual, éste parámetro permitirá crear topologías que involucren más routers. Por otro lado, la incorporación de un router real a la red mejora los resultados.

- La prueba de Throughput en una red con 3 routers virtuales y bajo las características descritas, nos indica que se trata de una red poco eficiente que puede enviar poca cantidad de paquetes de forma correcta. La incorporación de un router real a la red mejora los resultados.
- La prueba de Jitter en una red con 3 routers virtuales y bajo las características descritas, nos dice que este parámetro no es limitante para la creación de redes con más routers virtuales ya que sus valores están por debajo de los recomendados (<100ms). La incorporación de un router real a la red mejora los resultados sobre todo para paquetes de gran longitud.
- La interconexión entre dispositivos reales y dispositivos de red emulados usando GNS3 se realizan de forma transparente para los equipos.
- El flujo de paquetes en cada una de las pruebas (Latencia ICMP, Latencia UDP, Latencia TCP) se produjo como se esperó, lo cual se pudo constatar usando la capacidad de GNS3 para la captura de datos.
- Observamos que la Latencia de 3 routers virtuales no es en ningún caso el triple de la latencia de un sólo router virtual. Esta característica también se observa en el Throughput y en el Jitter.

5.6 Observaciones

5.6.1 Asignaciones de direcciones MAC

Las direcciones MAC identifican los adaptadores de red o tarjetas de red reales dentro de un segmento de red y son de gran utilidad en redes boadcast ya que permiten que las tramas que viajan por la red sean entregadas a hots específicos.

Para la identificación en capa 2 de las interfaces virtuales creadas por el simulador, GNS3 asigna direcciones MAC a las tarjetas de red físicas con la estructura **c8XX.XXXX.XXXX**, dichas direcciones son asignadas al momento de crear las topologías virtuales y permanecen invariables hasta el momento en que se vuelvan a cargar en el emulador, entonces las mismas interfaces tendrán diferentes direcciones MAC con la misma estructura.

La numeración que usa GNS3 en las direcciones MAC no se encuentra asignada, por la IEEE, a ninguna empresa fabricante de adaptadores de red o tarjetas de red reales, por lo tanto, la asignación de esta numeración específica no tendría que crear conflictos en la comunicación de dispositivos virtuales con equipos reales de cualquier tipo de fabricante.

Las figuras siguientes muestran las direcciones MAC que fueron asignadas por el simulador a las interfaces de red de los router virtuales creados en la topología de la figura 5.1. Para conocer la tabla de direcciones MAC que cada router virtual tiene almacenado se hizo uso del comando **show arp**.

```
Madrid#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.10.10.2             123   c801.0fa4.0000  ARPA   FastEthernet0/0
Internet  10.10.10.1               -   c802.0fa4.0000  ARPA   FastEthernet0/0
```

Figura 5.38a: Tabla de direcciones MAC del router Madrid.

```
Barcelona#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.10.10.2             -    c801.0fa4.0000  ARPA   FastEthernet0/0
Internet  10.10.10.1           124    c802.0fa4.0000  ARPA   FastEthernet0/0
Internet  10.10.10.6           124    c800.0fa4.0000  ARPA   FastEthernet0/1
Internet  10.10.10.5             -    c801.0fa4.0001  ARPA   FastEthernet0/1
```

Figura 5.38b: Tabla de direcciones MAC del router Barcelona.

```
Valencia#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.10.10.6             -    c800.0fa4.0000  ARPA   FastEthernet0/0
Internet  10.10.10.5           126    c801.0fa4.0001  ARPA   FastEthernet0/0
```

Figura 5.38c: Tabla de direcciones MAC del router Valencia.

Finalmente, la información de las tablas anteriores se ve reflejada en el siguiente esquema:

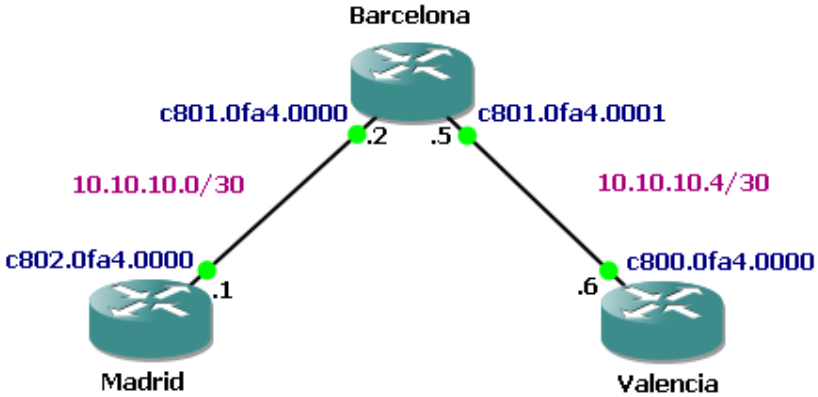


Figura 5.39: Asignación de direcciones MAC a cada interfaz.

CAPÍTULO 6: ANÁLISIS LA DE COMUNICACIÓN ENTRE VARIOS EMULADORES

Hasta el momento hemos trabajado con redes simuladas en un solo PC con el simulador GNS3 instalado, a partir de allí hemos tratado de analizar diferentes variantes, como por ejemplo, incorporar en la topología creada un router real. En este capítulo describiremos un escenario en donde los routers que forman parte de la topología son emulados por diferentes PCs que contienen a GNS3, con esta base, realizaremos pruebas que nos permitan analizar la comunicación que se produce entre ambos PCs.

6.1 Objetivo

Este apartado tiene el objetivo de mostrar las comunicaciones que se producen en escenarios que contienen routers emulados en más de un emulador. Para ello, usaremos la herramienta MRTG para la obtención de gráficas del tráfico que pasa por cada interfaz de los routers virtuales y estudiaremos como se realiza la comunicación SNMP.

6.2 Escenario de prueba

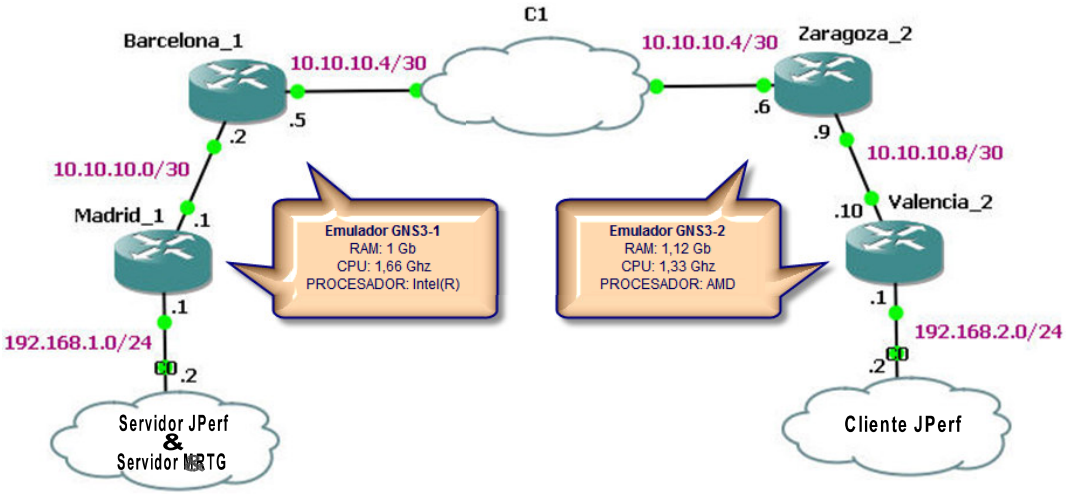


Figura 6.1: Escenario de prueba.

La figura 6.1 muestra el escenario con el que trabajaremos en este capítulo, vemos que consta de cuatro routers virtuales emulados en dos PCs, además esta topología incorpora dos enlaces reales hacia dos hosts que realizarán las funciones de cliente y servidor JPerf y uno más que será el que conecte ambos emuladores. El host *Servidor Jperf* también se comporta como un *servidor MRTG*, encargado de realizar peticiones de información SNMP a cada router virtual de la red y de recolectar dicha información para poder formar las gráficas del tráfico de cada una de las interfaces virtuales de la topología, usando la herramienta MRTG, durante momento de las pruebas.

Aquí la IOS usada también sera (***c2600-i-mz.123-3h***) y se asignarán los mismos parámetros de RAM (48MB) en los routers, por otro lado, los nombres de los routers no denotan locación sino son puramente descriptivos.

A continuación se menciona, con más detalle, todos los elementos involucrados en el escenario y sus principales características:

	Nombre	Tipo	Memoria	CPU	Observaciones
Router	Madrid_1	Virtual	RAM: 48Mb FLASH: 8Mb NVRAM:120 Kb	-	Router 2621 emulado
	Barcelona_1	Virtual	RAM: 48Mb FLASH: 8Mb NVRAM:120 Kb	-	Router 2621 emulado
	Zaragoza_2	Virtual	RAM: 48Mb FLASH: 8Mb NVRAM:120 Kb	-	Router 2621 emulado
	Valencia_2	Virtual	RAM: 48Mb FLASH: 8Mb NVRAM:120 Kb	-	Router 2621 emulado
PC	Simulador GNS3-2	Real	RAM: 1,12Gb	1,30 Ghz	Con procesador AMD
	Simulador GNS3-1	Real	RAM: 1 Gb	1,66 Ghz	Con procesador Intel(R)
	Cliente JPerf	Real	RAM: 1 Gb	2,80 Ghz	Con procesador Intel
	Servidor JPerf	Real	RAM: 1 Gb	1,66 Ghz	Con procesador Intel (R)
Interfaces	Fastethernet 0/0	Virtual	IP: 10.10.10.1		Perteneciente a Madrid_1
	Fastethernet 0/1	Real	IP: 192.168.1.1		Perteneciente a Madrid_1
	Fastethernet 0/0	Virtual	IP: 10.10.10.2		Perteneciente a Barcelona_1
	Fastethernet 0/1	Real	IP: 10.10.10.5		Perteneciente a Barcelona_1
	Fastethernet 0/0	Virtual	IP: 10.10.10.9		Perteneciente a Zaragoza_2
	Fastethernet 0/1	Real	IP: 10.10.10.6		Perteneciente a Zaragoza_2
	Fastethernet 0/0	Virtual	IP: 10.10.10.10		Perteneciente a Valencia_2
	Fastethernet 0/1	Real	IP: 192.168.2.1		Perteneciente a Valencia_2
	Adaptador de red 1	Real	IP: 192.168.1.2		Perteneciente a Servidor JPerf
	Adaptador de red 2	Real	IP: 192.168.2.2		Perteneciente a Cliente JPerf

Tabla 6.1: Elementos que aparecen en el escenario.

6.2.1 Configuración de escenario

Configuramos las interfaces de los cuatro routers emulados de nuestra topología, los comandos utilizados se muestran a continuación:

```
Madrid_1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Madrid_1(config)#interface fastethernet 0/0
Madrid_1(config-if)#ip address 10.10.10.1 255.255.255.252
Madrid_1(config-if)#no shutdown
Madrid_1(config-if)#exit
Madrid_1(config)#interface fastethernet 0/1
Madrid_1(config-if)#ip address 192.168.1.1 255.255.255.0
Madrid_1(config-if)#no shutdown
Madrid_1(config-if)#exit
Madrid_1(config)#
```

Figura 6.2: Configuración de interfaces en el router Madrid_1. [12]

```
Barcelona_1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Barcelona_1(config)#interface fastethernet 0/0
Barcelona_1(config-if)#ip address 10.10.10.2 255.255.255.252
Barcelona_1(config-if)#no shutdown
Barcelona_1(config-if)#exit
Barcelona_1(config)#interface fastethernet 0/1
Barcelona_1(config-if)#ip address 10.10.10.5 255.255.255.252
Barcelona_1(config-if)#no shutdown
Barcelona_1(config-if)#exit
Barcelona_1(config)#_
```

Figura 6.3: Configuración de interfaces en el router Barcelona_1. [12]

```
Zaragoza_2#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Zaragoza_2(config)#interface fastethernet 0/1
Zaragoza_2(config-if)#ip address 10.10.10.6 255.255.255.252
Zaragoza_2(config-if)#no shutdown
Zaragoza_2(config-if)#exit
Zaragoza_2(config)#interface fastethernet 0/0
Zaragoza_2(config-if)#ip address 10.10.10.9 255.255.255.252
Zaragoza_2(config-if)#no shutdown
Zaragoza_2(config-if)#exit
Zaragoza_2(config)#
```

Figura 6.4: Configuración de interfaces en el router Zaragoza_2. [12]

```
Valencia_2#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Valencia_2(config)#interface fastethernet 0/0
Valencia_2(config-if)#ip address 10.10.10.10 255.255.255.252
Valencia_2(config-if)#no shutdown
Valencia_2(config-if)#exit
Valencia_2(config)#interface fastethernet 0/1
Valencia_2(config-if)#ip address 192.168.2.1 255.255.255.0
Valencia_2(config-if)#no shutdown
Valencia_2(config-if)#exit
Valencia_2(config)#
```

Figura 6.5: Configuración de interfaces en el router Valencia_2. [12]

Finalmente se deberá configurar las interfaces de red de los dos PCs que actuarán como cliente y servidor Jperf como se muestra en la figura 4.15 de este trabajo.

6.3 Captura de tráfico

El análisis de la captura de tráfico en este capítulo se centrará en el intercambio de mensajes del protocolo SNMP para la creación de las gráficas del tráfico transportado por cada uno de los enlaces virtuales de la red.

La figura 6.6 muestra como se envían tramas que contienen al **protocolo SNMP** durante una de las pruebas descritas anteriormente. El servidor MRTG actúa como estación de gestión enviando paquetes SNMP (*get-request*) que solicitan los objetos ifInOctets y ifOutOctets de la MIB, dichos objetos contendrán los valores de todos los octetos que han pasado por esta interfaz durante todo el tiempo. En este caso en concreto, los routers Madrid_1 (10.10.10.1), Barcelona (10.10.10.5), Zaragoza_2 (10.10.10.6) y Valencia_2 (10.10.10.10) envían los mensajes de respuesta SNMP (*get-response*) con los datos los objetos solicitados al servidor MRTG.

Otro protocolo observado en la figura 6.6 y muy conocido que no ha sido analizado en capturas anteriores es el **protocolo ARP** (**A**ddress **R**esolution **P**rotocol), este protocolo se encarga de formar la tabla ARP de capa 2 de los hots al inicio de la

comunicación. El protocolo empieza enviando un mensaje de Broadcast a todos los hosts del segmento de red para pedir información acerca de alguna dirección MAC, en este caso, el *servidor MRTG* envía un mensaje ARP para conocer la dirección MAC de su puerta de enlace. [11]

16	108.344000	192.168.1.1	224.0.0.9	RIPv2	Response
17	109.937000	c8:01:11:bc:00:01	c8:01:11:bc:00:01	LOOP	Reply
18	113.515000	c8:01:11:bc:00:01	CDP/VTP/DTP/PAgP/UDLD	CDP	Device ID: Madrid_1 Port ID: FastEthernet0/1
19	118.625000	02:00:4c:4f:4f:50	Broadcast	ARP	Who has 192.168.1.1? Tell 192.168.1.2
20	118.640000	c8:01:11:bc:00:01	02:00:4c:4f:4f:50	ARP	192.168.1.1 is at c8:01:11:bc:00:01
21	118.640000	192.168.1.2	10.10.10.1	SNMP	get-request IF-MIB::ifInOctets.1 IF-MIB::ifOutOctets.1 SNMPv2-MIB::sysUpTime.0 SNMPv
22	118.719000	10.10.10.1	192.168.1.2	SNMP	get-response IF-MIB::ifInOctets.1 IF-MIB::ifOutOctets.1 SNMPv2-MIB::sysUpTime.0 SNMF
23	118.719000	192.168.1.2	10.10.10.1	SNMP	get-request IF-MIB::ifInOctets.2 IF-MIB::ifOutOctets.2 SNMPv2-MIB::sysUpTime.0 SNMPv
24	118.765000	10.10.10.1	192.168.1.2	SNMP	get-response IF-MIB::ifInOctets.2 IF-MIB::ifOutOctets.2 SNMPv2-MIB::sysUpTime.0 SNMF
25	119.937000	c8:01:11:bc:00:01	c8:01:11:bc:00:01	LOOP	Reply
26	122.328000	192.168.1.2	10.10.10.5	SNMP	get-request IF-MIB::ifInOctets.1 IF-MIB::ifOutOctets.1 SNMPv2-MIB::sysUpTime.0 SNMPv
27	122.375000	10.10.10.5	192.168.1.2	SNMP	get-response IF-MIB::ifInOctets.1 IF-MIB::ifOutOctets.1 SNMPv2-MIB::sysUpTime.0 SNMF
28	122.375000	192.168.1.2	10.10.10.5	SNMP	get-request IF-MIB::ifInOctets.2 IF-MIB::ifOutOctets.2 SNMPv2-MIB::sysUpTime.0 SNMPv
29	122.422000	10.10.10.5	192.168.1.2	SNMP	get-response IF-MIB::ifInOctets.2 IF-MIB::ifOutOctets.2 SNMPv2-MIB::sysUpTime.0 SNMF
30	127.156000	192.168.1.2	10.10.10.6	SNMP	get-request IF-MIB::ifInOctets.1 IF-MIB::ifOutOctets.1 SNMPv2-MIB::sysUpTime.0 SNMPv
31	127.219000	10.10.10.6	192.168.1.2	SNMP	get-response IF-MIB::ifInOctets.1 IF-MIB::ifOutOctets.1 SNMPv2-MIB::sysUpTime.0 SNMF
32	127.219000	192.168.1.2	10.10.10.6	SNMP	get-request IF-MIB::ifInOctets.2 IF-MIB::ifOutOctets.2 SNMPv2-MIB::sysUpTime.0 SNMPv
33	127.281000	10.10.10.6	192.168.1.2	SNMP	get-response IF-MIB::ifInOctets.2 IF-MIB::ifOutOctets.2 SNMPv2-MIB::sysUpTime.0 SNMF
34	129.937000	c8:01:11:bc:00:01	c8:01:11:bc:00:01	LOOP	Reply
35	130.625000	192.168.1.2	10.10.10.10	SNMP	get-request IF-MIB::ifInOctets.1 IF-MIB::ifOutOctets.1 SNMPv2-MIB::sysUpTime.0 SNMPv
36	130.750000	10.10.10.10	192.168.1.2	SNMP	get-response IF-MIB::ifInOctets.1 IF-MIB::ifOutOctets.1 SNMPv2-MIB::sysUpTime.0 SNMF
37	130.750000	192.168.1.2	10.10.10.10	SNMP	get-request IF-MIB::ifInOctets.2 IF-MIB::ifOutOctets.2 SNMPv2-MIB::sysUpTime.0 SNMPv
38	130.844000	10.10.10.10	192.168.1.2	SNMP	get-response IF-MIB::ifInOctets.2 IF-MIB::ifOutOctets.2 SNMPv2-MIB::sysUpTime.0 SNMF
39	135.437000	192.168.1.1	224.0.0.9	RIPv2	Response
40	139.937000	c8:01:11:bc:00:01	c8:01:11:bc:00:01	LOOP	Reply
41	149.937000	c8:01:11:bc:00:01	c8:01:11:bc:00:01	LOOP	Reply
42	159.953000	c8:01:11:bc:00:01	c8:01:11:bc:00:01	LOOP	Reply
43	164.859000	192.168.1.1	224.0.0.9	RIPv2	Response
44	169.953000	c8:01:11:bc:00:01	c8:01:11:bc:00:01	LOOP	Reply

CDP
ICMP
SNMP
LOOP
RIPV2

Figura 6.6: Captura de tráfico entre el router Madrid_1 y el servidor MRTG.

6.4 Gráficas MRTG de tráfico transportado

6.4.1 Prueba 1: Latencia de ICMP

PROCEDIMIENTO: Se sigue el mismo procedimiento que para la realización de la prueba del apartado 5.1.3.1 de este trabajo, con la diferencia que en esta ocasión el host origen de los paquetes ICMP será "*Cliente JPerf",* además alargaremos la duración de la prueba a **1500 segundos** con el fin de obtener gráficos MRTG más claros.

En la figura siguiente se puede apreciar claramente el flujo de octetos que han pasado por la interfaz fastethernet 0/0 del router Madrid_1 durante toda la prueba. Vemos como se envían y se reciben más octetos conforme se va aumentando la longitud de las tramas transmitidas, además se comprueba que, las tramas que salen de la interfaz, es decir, que transportan los mensajes de "*Echo Request*" al destino, son de la misma longitud que las tramas que entran a la interfaz, es decir, que transportan los mensajes *"Echo Response"* de vuelta al origen. Del gráfico podemos decir que el tráfico intercambiado entre cada flujo de datos (tramas LOOP, CDP, ARP, RIP) que se realiza continuamente entre los routers de la red es mínimo.

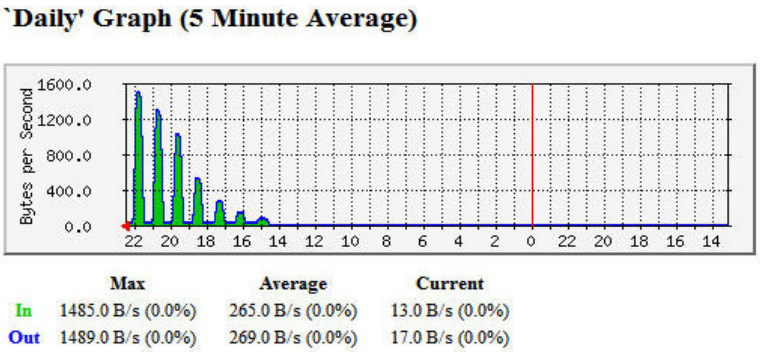
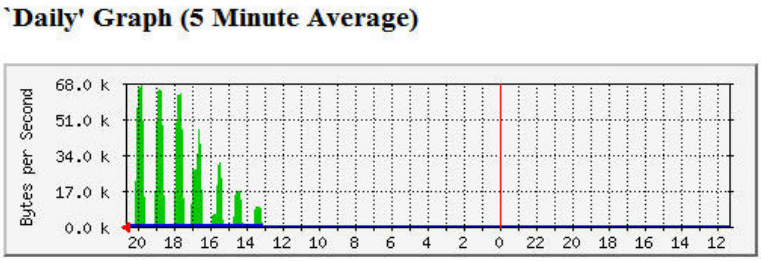


Figura 6.7: Tráfico en el enlace fastethernet 0/0 del router Madrid_1.

6.4.2 Prueba 2: Throughput en la red

PROCEDIMIENTO: Se sigue el mismo procedimiento que para la realización de la prueba del apartado 5.1.3.4 pero se alarga la duración de la prueba a **1500 segundos** y el origen del flujo de datos UDP es el host *"Cliente Jperf"*.

La figura 6.8 muestra el flujo de datos que ha pasado por la interfaz fastethernet 0/0 del router Madrid_1 durante toda la prueba. Aquí también observamos el aumento progresivo del flujo de datos cuando se va agrandando la longitud de las tramas enviadas, de la figura podemos decir que la interfaz puede soportar un flujo de 67,6 Kbyes como máximo sin pérdidas de paquetes. Por otro lado, en este caso notamos que el tráfico entrante proveniente del cliente IPerf es el que varía mientras que el que sale del router Madrid permanece en sus valores mínimos.



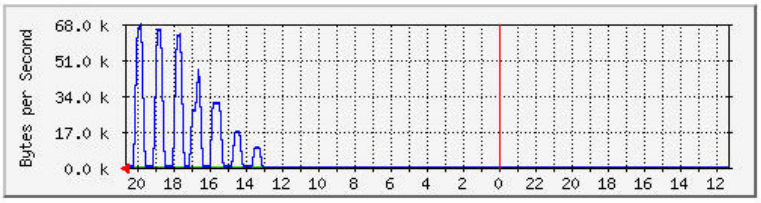
	Max	Average	Current
In	67.6 kB/s (0.5%)	16.0 kB/s (0.1%)	10.0 B/s (0.0%)
Out	110.0 B/s (0.0%)	21.0 B/s (0.0%)	19.0 B/s (0.0%)

Figura 6.8: Tráfico en el enlace fastethernet 0/0 del router Madrid_1.

6.4.3 Prueba 3: Jitter en la red

PROCEDIMIENTO: Se sigue el mismo procedimiento que para la realización de la prueba del apartado 5.1.3.5, teniendo como origen de los datos al router Madrid_1 y con la duración de la prueba de **1500 segundos.**

Como las medias de Throughput y de Jitter están ligadas, es este caso, la figura 6.9 muestra el flujo de octetos que pasan por el router Barcelona_1 en su interfaz 0/0. En este caso, vemos que el flujo de datos que sale hacia el router Madrid_1 por la interfaz 0/0 es el que cambia mientras que el flujo entrante permanece en sus mínimos.



	Max	Average	Current
In	105.0 B/s (0.0%)	15.0 B/s (0.0%)	13.0 B/s (0.0%)
Out	67.5 kB/s (0.5%)	16.9 kB/s (0.1%)	16.0 B/s (0.0%)

Figura 6.9: Tráfico en el enlace fastethernet 0/0 del router Barcelona_1.

6.5 Conclusiones

Después de la realización de las pruebas de este capítulo hemos podido llegar a las siguientes conclusiones:

- Es posible la interconexión de emuladores que trabajan conjuntamente.
- La conexión externa es transparente a los equipos.
- Los routers virtuales soportan ser configurados como agentes SNMP para enviar sus datos al servidor MRTG.

CAPÍTULO 7: CONCLUSIONES Y RECOMENDACIONES

Después del desarrollo de este proyecto se puede decir que se han cumplido los objetivos planteados inicialmente. Los siguientes puntos explican detalladamente las conclusiones obtenidas:

- GNS3 se presenta como una aplicación libre, fácil de instalar y usar, con gran utilidad tanto en el mundo empresarial, ya que su uso reduce el coste de implementación de las redes; como en el mundo académico, debido a que hace más accesible el estudio del *networking*.

- GNS3 trabaja conjuntamente con Dynamips y Dynagen, para realizar el "milagro" de la emulación; estos 3 componentes dotan a un PC tradicional de la capacidad de: conectarse vía Telnet a la consola de un router virtual, realizar capturas de los paquetes que pasan por sus enlaces virtuales, trabajar conjuntamente con varios emuladores para repartir su carga de procesamiento y lo más importante, permite la comunicación con equipos reales externos. Obviamente GNS3 posee algunas deficiencias, como la incapacidad de emular switches Ethernet o la falta de soporte a todas las plataformas CISCO disponible en el mercado, pero al ser una aplicación que está en constante actualización y con gran aceptación en el público es muy probable que estas carencias se solucionen en un futuro próximo.

- Algunas *recomendaciones* para el buen uso de GNS3 serían: Prestar especial atención en la configuración de los paths hacia Dynamips y hacia la aplicación que usaremos para leer los paquetes capturados (*Wireshark* y *tcpdump*), habilitar las herramientas de optimización de uso de recursos como son Ghostios, Sparsemem y Mmap para evitar problemas de funcionamiento del sistema. Además tener especial cuidado en usar una CISCO IOS que ya este descomprimida para agilizar el proceso de apertura de las conexiones Telnet a los routers.

- Las *limitaciones* de Dynamips más relevantes encontradas al culmine de este trabajo son principalmente: La falta de soporte para la emulación de toda la gama de plataformas CISCO disponibles en el mercado y la incapacidad de emular interfaces VICs y VWICs necesarias para la configuración de funcionalidades que permitan el tráfico de voz. Por lo tanto, podemos decir que GNS3, en su versión más actual 0.61, todavía no es capaz de realizar DSP (**D**igital **S**ignal **P**rossesing).

- Con la experiencia adquirida en el uso de GNS3 al término de este trabajo, podemos decir que un usuario que se conecta remotamente a un router emulado con esta aplicación, no puede diferenciar si el equipo al que se conecta es real o virtual, ya que desde la *CLI* no se aprecia ninguna diferencia a no ser que se ingrese el comando **show version**. Con este comando, sí notaremos reportes inusuales en lo referente a las memorias ROM y FLASH.

- Las capacidades de procesamiento óptimas de emulador GNS3 dependen de la cantidad de routers que se desean emular, es decir, se requerirán más recursos de CPU y memoria RAM si se quieren emular topologías con muchos routers. Una estimación aproximada nos dice que un emulador con procesador Intel de 2 Ghz y 1 GB de RAM puede emular 10 routers y otro con procesador

Intel Core 2 Duo de 2,6 Ghz y 2 GB de RAM puede emular 28 routers, siendo Linux el sistema operativo que nos otorga un comportamiento de GNS3 más estable.

- Para el buen funcionamiento del sistema el PC que se elija para la instalación de GNS3 debe tener un buen balance entre CPU y Memoria, pero además se debe tener en cuenta: elegir un buen valor de IDLE-PC, usar una IOS que tenga las funcionalidades que se necesite pero requiera bajas cantidades de RAM para funcionar, crear topologías que usen menos del 80% del CPU del emulador.

- Las pruebas del capítulo 4 realizadas para comparar el rendimiento de un router virtual y con un router real, nos confirman que es mucho peor el rendimiento del router virtual, sobre todo si se toman en cuenta los valores de Throughput obtenidos, muy por dejado de los teóricos expuestos por CISCO [24]. Con respecto a la Latencia y al Jitter obtenemos que ambos toman valores altos (Latencia 7ms-14ms y Jitter 3ms – 9ms) y son inestables conforme se aumenta la longitud de los paquetes transmitidos, en comparación con los valores obtenidos para los routers reales (Latencia 1ms y Jitter 0ms).

- La mejora en las características de procesamiento del PCs que contiene al emulador GNS3 (doble procesamiento) no nos ayuda a emular routers con mejores capacidades de Throughput, pero si nos ayuda a obtener equipos con bajos retardos (casi iguales a los equipos reales) y más estables ante cambios en la longitud de los paquetse enviados.

- De las pruebas realizadas en las topologías de red del capítulo 5 podemos concluir que una red que contiene 3 routers virtuales emulados con GNS3 bajo las condiciones expuestas no llegan a sobrepasar los valores límites punto a punto recomendamos en la **ITU G.114** para el envío de tráfico a gran velocidad: Jitter (100ms) y Latencia (150ms para redes LAN y 300ms para redes WAN). La creación de redes que involucren muchos más routers emulados no cumplirían los requisitos anteriores, además serán ineficientes en cuando al Throughput efectivo que pase por ellas.

- En el capítulo 5 también se demuestra que el trabajo conjunto de equipos reales y emulados con GNS3 es posible y se realiza de forma transparente al emulador. Realizamos pruebas con tráfico de diferentes protocolos (ICMP, TCP, UDP) y todas ellas transcurrieron como se esperaba según pudimos evidenciar por medio del uso de la captura de paquetes con *Wireshark*. Cabe resaltar que a la hora de asignar direcciones MAC a las interfaces virtuales de la red, GNS3 utiliza prefijos propios que cambian en casa simulación y que no producen conflictos si se produce una comunicación con routers reales.

- El capítulo 6 nos indica que es posible la interoperabilidad entre varios emuladores que trabajan conjuntamente en la misma red. Además los routers virtuales son capaces de comportarse como agentes SNMP y mostrar el tráfico que pasa por sus interfaces por medio de *MRTG*.

CAPÍTULO 8: LÍNEAS FUTURAS

Tras la realización de este trabajo podemos resaltar las siguientes líneas futuras:

- Crear topologías que incorporen diferentes dispositivos de red como switches Ethernet, switches ATM, switches Frame Relay, Firewall PIX, etc. Estudiar su comportamiento y comprobar que también es posible comunicarlas con redes externas reales.

- Profundizar en el estudio de la capacidad que incorpora GNS3 para el balanceo de carga de procesamiento entre varios emuladores externos y analizar las consecuencias de su uso en el rendimiento de redes.

- Emular plataformas diferentes a la 2621 usada en este trabajo, además probar versiones diferentes de CISCO IOS que contengan otras funcionalidades para esta plataforma, o incluso, para cualquiera de las distintas plataformas que soporta GNS3 mostradas en el apartado 3.2.1 de este trabajo.

- Realizar un estudio más detallado del uso de recursos producido por GNS3, para estimar de forma precisa la cantidad de recursos necesarios para la emulación de un determinado número de routers sin perjudicar el funcionamiento del emulador.

- Probar la interoperabilidad entre redes conformadas por routers CISCO emulados con GNS3 y redes que cuentan con equipos de otros fabricantes como, por ejemplo, JUNIPER.

CAPÍTULO 9: BIBLIOGRAFÍA

[1] Cisco Systems, Inc., *Internetworking Technologies Handbook*, Cisco Press, 4th Edition, Septiembre 2003
http://www.ciscopress.com/articles/article.asp?p=101658

[2] Cisco Systems, Inc., *Using the Command-Line Interface*,
http://www.cisco.com/en/US/docs/ios/12_0/configfun/configuration/guide/fcui.pdf

[3] Greg Anuzelli, *Dynamips / Dynagen Tutorial*, V1.11.7
http://www.dynagen.org/tutorial.htm

[4] Mike Fuszner, *Graphical Network Simulator*, Versión 1.0

[5] Cisco Systems, Inc., *Cisco IOS Reference Guide,* Julio 2007
http://www.cisco.com/warp/public/620/1.html

[6] James Boney, *Cisco IOS in a Nutshell*, O´Reilly Media, 2nd Edition, Agosto 2005

[7] Dave Hucaby; Steve McQuerry, *Cisco® Field Manual: Router Configuration*, Cisco Press, Diciembre 2001, Apéndice A.

[8] Cisco Systems, Inc*., Portable Product Sheet – IOS Packaging*, Octubre 2005

[9] Cisco Systems, Inc., *Cisco 1700 Series Modular Access Routers*
http://www.cisco.com/en/US/docs/routers/access/1700/1711/hardware/quick/guide/171Xq_sp.html

[10] Cisco Systems, Inc., *Cisco Access Router Manager User's Guide, 1.1*, 2003
http://www.cisco.com/en/US/docs/net_mgmt/access_router_manager/1.1/user/guide/concepts.html

[11] Universidad de Murcia (DITEC), *Tutorial y descripción técnica de TCP/IP*, 2003
http://ditec.um.es/laso/docs/tut-tcpip/3376c212.html

[12] Scott Empson, *CCNA Portable Command Guide*, Cisco Presss, 1st edition, Noviembre 2005

[13] J. Manuel Huidodro, SNMP. *Un protocolo simple de gestión*, Colegio oficial de ingenieros de comunicación.
http://www.coit.es/publicac/publbit/bit102/quees.htm

[14] Joaquín Garzón Alcalde, *Monitorización gráfica del tráfico de red y otros parámetros del sistema*.
http://beta.redes-linux.com/manuales/Monitorizacion_redes/mrtg.pdf

[15] Peter J. Welcher, *Configuring SNMP in CISCO Routers*, Mayo 1999
http://www.netcraftsmen.net/resources/archived-articles/370-configuring-snmp-in-cisco-routers.html

[16] Mark Gates, *Iperf Version 2.0.0*, Universidad de Illinois, Mayo 2004
http://www.onl.wustl.edu/restricted/iperf.html

[17] Simon Leinen, *Iperf Tool*, Performance Enhancement & Response Team, Géant-eduPERT, Julio 2009
http://kb.pert.geant.net/PERTKB/IperfTool

[18] Tobi Oetiker, *The Multi Router Traffic Grapher*, Noviembre 2008
http://oss.oetiker.ch/mrtg/index.en.html

[19] Cisco Systems, Inc., *Response Time Reporter Enhance*.
http://www.optimumdata.com/shop/files/cisco/3600/3600_Response_Time_Report_Enhance.pdf

[20] Kevin Dooley; Ian J. Brown, *Cisco IOS Cookbook*, O´Reilly Media, 2nd Edition, Diciembre 2006, Capítulo 17

[21] Pablo Scaniello, Gonzalo Sosa, *Medidas a realizar en dispositivos de redes*
http://iie.fing.edu.uy/ense/asign/perfredes/trabajos/trabajos_2005/dispositivos/dispRedes.pdf

[22] Scott Bradner, RFC 1242: *Benchmarking Terminology for Network Interconnection Devices*, Network Working Group, Julio 1991

[23] Scott Bradner, RFC 2544: *Benchmarking Methodology for Network Interconnection Devices*, Network Working Group, Marzo 1999

[24] Cisco Systems, Inc., *Portable Product Sheets – Routing Performance,* Noviembre 2009.
http://www.optimumdata.com/shop/files/cisco/3600/3600_Response_Time_Report_Enhance.pdf

[25] Jonathan Davidson; Tina Fox, *Deploying Cisco® Voice over IP Solutions*, Cisco Presss, 1st edition, Noviembre 2001, Página 43

[26] Salvatore Collora,Anne Smith,Ed Leonhardt, *Cisco CallManager best practices, Cisco Presss*, 2004

[27] *FORO de consultas de Dynamips, Dynagen*.
http://7200emu.hacki.at/index.php

[28] Cisco Systems, Inc., *Cisco 2600 Series Router Architecture (Document ID: 23852)*, Cisco 2600 Series Multiservice Platforms, Abril 2007
http://www9.cisco.com/application/pdf/paws/23852/2600_architecture_23852.pdf

[29] Wireshark Wiki, *Configuration Testing Protocol (Loop)*, Abril 2008
http://wiki.wireshark.org/Loop

[30] FORO de consultas de GNS3
http://www.gns3.net/phpBB/

[31] Boson NetSim for CCNA
http://www.boson.com

[32] Vyatta Open Networking - Software-based Routing & Security - Open Alternative to Cisco (http://www.vyatta.com)