



Universidad
Zaragoza

Proyecto Fin de Carrera

Ingeniería de Telecomunicación
Especialidad de Telemática

Análisis de la interoperabilidad con entornos reales de arquitecturas de red emuladas con GNS3/Dynamips
(para ámbitos profesionales y académicos)

Autor
José Javier Serrano García

Director
Dr. Julián Fernández Navajas

Mayo de 2013



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

Agradecimientos

Desde estas líneas deseo agradecer a la Escuela de Ingeniería y Arquitectura de la Universidad de Zaragoza, y en especial al área de Ingeniería Telemática del Departamento de Ingeniería Electrónica y Comunicaciones, la formación académica a lo largo de estos años y los medios materiales y humanos puestos a mi disposición para la realización del presente proyecto. Deseo particularizar un poco más mi gratitud a José Ignacio Mendieta, por su inestimable ayuda en el laboratorio, y a Julián Fernández Navajas, profesor en varias asignaturas de la carrera y director del proyecto. Muchas gracias Julián, por tus conocimientos, tu fantástica capacidad de síntesis y por tu paciencia.

Desde un punto de vista más personal, gracias infinitas a mis cinco imprescindibles, los míos, mi familia. Suena a tópico, pero gracias por todo el apoyo, y por estar ahí arrimando el hombro cuando pintaban bastos

Resumen del proyecto

El objetivo del presente proyecto es abordar el estudio de las prestaciones, posibilidades y márgenes de trabajo que en la actualidad presentan ciertas herramientas destinadas a la emulación de sistemas telemáticos, para su posterior aplicación en entornos profesionales y/o académicos. En concreto, el proyecto se centra en el análisis del entorno software de simulación/emulación constituido por *GNS3/Dynamips*, con base y apoyo en hardware de red *Cisco Systems* y sus sistemas operativos, principalmente *Cisco IOS*, y en menor medida *Cisco CatOS*. Tras los necesarios capítulos introductorios 1 y 2 de la memoria, el corazón del proyecto y la carga principal de contenidos se desarrolla en los capítulos 3 y 4 (y anexos, elaborados *ex profeso* para este proyecto y fundamentales en el desarrollo de los análisis).

En primer lugar, el desarrollo del proyecto comienza con el proceso necesario de adquisición de habilidades de trabajo en un entorno controlado de laboratorio. Este entorno está constituido por hardware de red *Cisco Systems* y sus S.O., en concreto el dispositivo de red *Cisco Catalyst 5500 Switch*, plataforma hardware modular presente en los laboratorios de telemática del DIEC. Se profundiza, igualmente, en el estudio de los S.O. *Cisco IOS* y *Cisco CatOS*, necesarios para configurar los equipos conforme a las arquitecturas de red planteadas en el proyecto, LAN Ethernet/VTP y WAN ppp, tanto en entornos reales como emulados. Estos análisis culminan en la elaboración del anexo 1, anexo que sirve para la configuración actual de los equipos *Cisco Catalyst 5500 Switch* del laboratorio de telemática del DIEC.

Seguidamente, el proyecto se centra en el análisis del entorno emulador *GNS3/Dynamips*. Dicho entorno software permite, fundamentalmente, emular arquitecturas basadas en hardware de red *Cisco Systems*, utilizando el mismo S.O. que el hardware de red real. Así, se hace necesaria la integración de los S.O. en el entorno de emulación, según detalla el anexo 3. Con ello, mediante el análisis de escenarios que emulan arquitecturas de red como LAN Ethernet /VTP y WAN ppp en *GNS3/Dynamips*, imagen de entornos de red reales, se pretende constatar: su *usabilidad*, capacidad de diseño y adquisición de habilidades propias del entorno real (para su utilidad en entornos académicos o como herramienta de análisis previo a la actuación en un entorno real de trabajo), rigor y correspondencia funcional en la emulación. Ligado a lo anterior, se busca establecer los requerimientos técnicos, capacidad y rendimiento del hardware emulador bajo sistemas operativos de usuario comunes (Windows fundamentalmente, complementado con el análisis en Linux del anexo 4), la potencial escalabilidad de las arquitecturas emuladas sujetas al consumo de recursos del sistema emulador, la influencia sobre la latencia de red emulada y márgenes fiables de trabajo

Paralelamente, se acomete el objetivo fundamental y prioritario del proyecto que es el análisis de la potencial capacidad que tiene el entorno emulador de *interoperar* con tráfico y equipos de red reales, así como la influencia que puede tener dicha *interoperación* sobre un parámetro de red crítico y decisivo, como la latencia. El comportamiento de la latencia de red puede considerarse indicativo del comportamiento en tiempo real del entorno emulador. Finalmente, en el capítulo 5 se analizan los resultados obtenidos y se extraen conclusiones sobre los objetivos prioritarios y todo el trabajo desarrollado, planteando así mismo líneas futuras de trabajo y/o investigación relacionadas con este proyecto.

Tabla de contenidos

Capítulo 1: Introducción al proyecto.	1
1.1. Perspectiva actual del ámbito de estudio.	1
1.2. Fundamentos y herramientas de trabajo. Objetivos del proyecto.	2
1.2.1. Adquisición de habilidades <i>vía</i> entorno software <i>GNS3/Dynamips</i>	2
1.2.2. Análisis del hardware de red <i>Cisco Systems</i> y sistemas operativos <i>Cisco IOS</i> y <i>Cisco CatOS</i> . Interoperabilidad de <i>GNS3/Dynamips</i> con entornos de red reales. .	2
1.2.3. Extensión y aplicación al ámbito académico.....	3
1.2.4. Evaluación de la capacidad y rendimiento del sistema hardware emulador (PC emulador). Influencia sobre la latencia de red en la <i>interoperación entorno real ↔ entorno emulado</i> y sobre la potencial escalabilidad de las arquitecturas de red WAN ppp y LAN Ethernet/VTP.	3
1.3. Contenido esquemático de la memoria del proyecto. Distribución de los contenidos por capítulos.	4
Capítulo 2: Consideraciones de diseño y entorno de trabajo	7
2.1. Consideraciones generales del proyecto.	7
2.2. Consideraciones del entorno hardware de red <i>Cisco Systems</i>	7
2.3. Consideraciones del entorno software emulador <i>GNS3/Dynamips</i>	9
2.4. Escenario de trabajo general para el análisis. Contingencias en el diseño y soluciones adoptadas.....	11
Capítulo 3:Análisis de arquitecturas de red en <i>GNS3/Dynamips</i>	15
3.1. Análisis de la arquitectura de red WAN ppp.	15
3.1.1. Consideraciones iniciales y descripción para la configuración de la arquitectura de red.	15
3.1.2. Captura y análisis de tráfico de red emulado en el enlace serie WAN ppp... <td>17</td>	17
3.2. Análisis de la arquitectura de red LAN Ethernet / VTP	18
3.2.1. Consideraciones iniciales y descripción para la configuración de la arquitectura de red.	18
3.2.2. Captura y análisis de tráfico emulado en el enlace troncal (trunk link).	20

Capítulo 4: Análisis del rendimiento y capacidad del sistema emulador. Relación con la latencia de red en la <i>interoperación entorno real → entorno emulado</i>	25
4.1. Contexto del análisis	25
4.2. Análisis bajo sistema operativo Windows.	26
4.2.1. Análisis del rendimiento del sistema emulador para la arquitectura de red WAN ppp.....	26
4.2.2. Análisis de la latencia para la arquitectura de red WAN ppp.....	32
4.2.3. Análisis del rendimiento del sistema emulador para la arquitectura de red LAN Ethernet/VTP.....	38
4.2.4. Análisis de la latencia para la arquitectura de red LAN Ethernet /VTP.....	40
Capítulo 5: Análisis final del proyecto, conclusiones y líneas futuras de trabajo y/o investigación.	45
5.1. Conclusiones finales de los análisis de las características, rigor y correspondencia funcional de <i>GNS3/Dynamips</i> . Conclusiones de la interoperabilidad entre entornos de red reales y emulados.....	45
5.2. Análisis de la capacidad, rendimiento y latencia de red de <i>GNS3/Dynamips</i> en Windows.	47
5.2.1. Conclusiones a los análisis sobre la capacidad y rendimiento del sistema emulador	47
5.2.2. Conclusiones a los análisis de la latencia de red en la <i>interoperación entorno real ↔ entorno emulado</i>	48
5.3. Inconvenientes principales asociados al desarrollo del proyecto. Líneas futuras de trabajo y/o investigación.....	53
5.3.1. Inconvenientes principales en el desarrollo del proyecto y soluciones adoptadas.	53
5.3.2. Líneas futuras de trabajo y/o investigación relacionadas con el proyecto. ..	54
Capítulo 6: Referencias bibliográficas y <i>en línea</i>	55
Capítulo 7: Anexos	57

7.1. Anexo 1: Configuración & gestión de plataformas hardware modulares de red Cisco Catalyst 5500 Switch (y similares).....	57
7.1.1. Introducción.....	57
7.1.2. Modos de operación en arquitecturas <i>Cisco Systems</i>	58
7.1.3. Conexión y configuración <i>vía</i> módulo supervisor <i>Cisco WS-X5530-E3</i>	59
7.1.4. Configuración del módulo <i>Router-Switch Module Cisco WS-X5302</i>	63
7.1.5. Configuración de Interfaces E1 en módulos <i>Cisco 8-port Multichannel T1/E1</i> . Uso como conmutador Frame-Relay	67
 7.2. Anexo 2: Configuraciones de red para arquitecturas emuladas en GNS3/Dynamips. Interoperación entorno real $\leftarrow\rightarrow$ entorno emulado.....	71
7.2.1. Configuración y verificación de la arquitectura de red WAN ppp.....	71
7.2.2. Análisis exhaustivo del tráfico de red emulado en el enlace serie WAN ppp.	85
7.2.3. Configuración de Listas de Control de acceso (ACL).....	87
7.2.4. Análisis del tráfico de red en el enlace <i>equipo real 1</i> $\leftarrow\rightarrow$ <i>nodo virtual emulado 1</i> para el análisis de la <i>interoperabilidad</i> de los entornos de red.....	92
7.2.5. Configuraciones de red adicionales para el análisis de prestaciones y rendimiento del emulador <i>GNS3/Dynamips</i> en la arquitectura WAN ppp.....	96
7.2.6. Configuración y verificación de la arquitectura LAN Ethernet/VTP.....	102
7.2.7. Configuraciones de red adicionales para el análisis de prestaciones y rendimiento del emulador <i>GNS3/Dynamips</i> en la arquitectura LAN/VTP	112
7.2.8. Complemento <i>teórico-práctico</i> a las conclusiones del proyecto.....	114
 7.3. Anexo 3: Extracción de sistemas operativos Cisco IOS de hardware de red Cisco Systems y posterior integración en GNS3/Dynamips. Configuración básica de GNS3/Dynamips: parámetros y prestaciones fundamentales.....	121
7.3.1. Introducción.....	121
7.3.2. Extracción del sistema <i>Cisco IOS</i> de dispositivos de red <i>Cisco Systems</i> ..	121
7.3.3. <i>Interoperación</i> de dispositivos de red reales y emulados en <i>GNS3/Dynamips</i>	124
7.3.4. Integración de <i>Cisco IOS</i> en el entorno emulador <i>GNS3/Dynamips</i>	127
7.3.5. Optimización del rendimiento de CPU: parámetro <i>idle PC</i>	130
7.3.6. Optimización del rendimiento de la memoria RAM.	131

7.4. Anexo 4: Análisis de prestaciones y rendimiento del sistema emulador en entornos de trabajo Linux (Distribución <i>Ubuntu 12.04</i>).....	133
7.4.1. Instalación/Configuración de <i>GNS3/Dynamips</i> en sistemas Linux.....	133
7.4.2. Análisis de la capacidad y rendimiento del sistema emulador en entornos de trabajo Linux.....	139
7.4.3. Análisis de la latencia en la <i>interoperación entorno real ↔ entorno emulado</i> en sistemas operativos Linux.....	141
 7.5. Anexo 5: Implementación de un servicio telemático cliente /servidor FTP <i>vía interoperación entorno real (Ethernet 802.3) ↔ entorno emulado</i> en arquitecturas de red emuladas.....	144
7.5.1. Implementación de un servicio cliente /servidor FTP <i>vía interoperación entorno real (Ethernet 802.3) ↔ entorno emulado</i> en el análisis de la arquitectura de red WAN ppp.....	144
7.5.2. Implementación de un servicio cliente /servidor FTP <i>vía interoperación entorno real (Ethernet 802.3) ↔ entorno emulado</i> en el análisis de la arquitectura de red LAN Ethernet/VTP.....	146

Indice de figuras

Capítulo 2

Figura 2.1. Dispositivo de red <i>Cisco Catalyst 5500 Switch</i> del laboratorio de telemática del DIEC	8
Figura 2.2. Escenario general de trabajo para el análisis del emulador <i>GNS3/Dynamips</i>	11
Figura 2.3. Esquema de contingencias y soluciones adoptadas para implementar las arquitecturas de red del proyecto en <i>GNS3/Dynamips</i> en base al dispositivo <i>Cisco Catalyst 5500 Switch</i> del laboratorio	12
Figura 2.4. Nodo <i>EtherSwitch router</i> general disponible en el entorno de emulación <i>GNS3/Dynamips</i>	13

Capítulo 3

Figura 3.1. Topología/simulación gráfica en <i>GNS3</i> de la arquitectura de red WAN ppp	15
Figura 3.2. Tráfico de red en el enlace serie WAN ppp del entorno emulado.....	17
Figura 3.3. Topología/simulación gráfica en <i>GNS3</i> de la arquitectura LAN Ethernet/VTP	18
Figura 3.4. Captura de tráfico emulado en enlace troncal (protocolo 802.1q)	20
Figura 3.5. Detalle de trama <i>Ethernet 802.1q</i> capturada en el enlace troncal	20
Figura 3.6. Estructura de trama <i>Ethernet 802.1q</i> . Imagen extraída de “ <i>Cisco Networking Academy program, CCNP: Building Multilayer Switching Networks v.5.0</i> ” [10]	21
Figura 3.7 a). Valor del subcampo <i>Ethertype</i> (“10000001 00000000 ” = 0x8100) en la cabecera “Tag” de una <i>tagged frame</i> 802.1q	22
Figura 3.7 b). Valor del subcampo <i>PRI</i> (“000 ”) en la cabecera “Tag” de una <i>tagged frame</i> 802.1q	22

Figura 3.7 c). Valor del *subcampo Token Ring Encapsulation Flag* (“ 0 ”) en la cabecera “Tag” de una *tagged frame* 802.1q 23

Figura 3.7 d). Valor de *subcampo VLAN ID* (“ 0000 00000010 ” = VLAN2) en la cabecera “Tag” de una *tagged frame* 802.1q 23

Capítulo 4

Figura 4.0. Contexto esquemático para el análisis del capítulo 4 25

Figura 4.1. Esquema físico para la conexión/integración de hardware y tráfico real en el entorno de emulación 26

Figura 4.2. Consumo de CPU en el sistema emulador (PC) en función del número de nodos emulados activos para la arquitectura WAN ppp y *configuración hardware 1* 31

Figura 4.3. Consumo de CPU en el sistema emulador (PC) en función del número de nodos emulados activos para la arquitectura WAN ppp y *configuración hardware 2* 32

Figura 4.4. Consumo de CPU en el sistema emulador (PC) en función del número de nodos emulados activos para la arquitectura LAN/VTP y *configuración hardware 1* 39

Figura 4.5. Consumo de CPU en el sistema emulador (PC) en función del número de nodos emulados activos para la arquitectura LAN/VTP y *configuración hardware 2* 40

Capítulo 5

Figura 5.1. Secuenciación del análisis de las características, rigor y correspondencia funcional de *GNS3/Dynamips* 45

Figura 5.2. Comportamiento del emulador para la latencia de red respecto al consumo de recursos y la longitud de trama para las *tablas 4.4, 4.5 4.11 y 4.12* referentes al primero de los experimentos de la latencia en ambas arquitecturas de red 49

Figura 5.3. Interpretación gráfica de las dependencias existentes en la latencia de red emulada en la prestación de un determinado servicio telemático 52

Figura 5.4. Representación esquemática de las posibles líneas de investigación futuras .. 54

Capítulo 7

Figura 7.1. Configuración de <i>HyperTerminal</i> para la conexión al módulo supervisor <i>Cisco WS-X5530-E3</i>	60
Figura 7.2. Dispositivo de red <i>Cisco Catalyst 5500 Switch</i> presente en el laboratorio de Telemática del DICEC	64
Figura 7.3. Escenario de estudio en el laboratorio planteado para el uso del <i>Cisco Catalyst 5500</i> como conmutador <i>WAN Frame Relay</i>	68
Figura 7.4. Contexto gráfico de opciones en los nodos <i>EtherSwitch router</i>	72
Figura 7.5. Contexto gráfico para incorporar /configurar los módulos de interfaces en <i>GNS3/Dynamips</i>	73
Figura 7.6. Inicio y acceso a consola en dispositivos emulados en <i>GNS3</i>	74
Figura 7.7. Creación y configuración de interfaces virtuales SVI para VLAN2 y VLAN3 en el nodo emulado <i>EtherSwitch router 1</i>	75
Figura 7.8 a). Configuración VLAN: configuración y asignación VLAN de interfaces en nodos emulados <i>EtherSwitch router</i>	76
Figura 7.8 b). Configuración VLAN: activación de interfaces SVI desde <i>VLAN Database</i>	77
Figura 7.9. Configuración de interfaz <i>Ethernet</i> en <i>equipo_real 1</i>	78
Figura 7.10. <i>Test</i> de conectividad y trayectoria del tráfico en la commutación VLAN a nivel 3	79
Figura 7.11. Configuración del enlace serie WAN ppp en nodo <i>EtherSwitch router 1</i>	81
Figura 7.12. Tabla de enrutamiento (situación inicial) en el nodo <i>EtherSwitch router 1</i>	82
Figura 7.13. Configuración del protocolo RIP v2 en el nodo <i>EtherSwitch router</i> emulado	83
Figura 7.14. Tabla de enrutamiento final después de habilitar RIP v2 en el entorno emulado del análisis	84
Figura 7.15. Trama capturada en el enlace serie WAN ppp con información RIP v2	85
Figura 7.16. Esquema de funcionamiento general de una ACL. Imagen extraída de “Programa de la Academia de Networking CISCO CCNA2 v3.1” [11]	88

Figura 7.17. Configuración de ACL en nodo emulado <i>EtherSwitch router 1</i>	88
Figura 7.18. Configuración de ACL en nodo emulado <i>EtherSwitch router 2</i>	89
Figura 7.19. Esquema de numeración para ACL numeradas. Imagen extraída de “ <i>Programa de la Academia de Networking CISCO CCNA2 v3.1</i> “ [11]	89
Figura 7.20. <i>Test</i> de conectividad VLAN sobre el entorno de red emulado, una vez configuradas las ACL (al respecto, el <i>código de colores</i> utilizado se identifica describe en la página siguiente)	90
Figura 7.21 a) Tráfico de red capturado en el enlace <i>equipo_real 1</i> → <i>nodo virtual 1</i> , para una longitud de trama <i>Ethernet</i> de 64 bytes (longitud mínima, MTU = 46 bytes) en la arquitectura WAN ppp	92
Figura 7.21 b) Tabla ARP en nodo emulado <i>EtherSwitch router 1</i>	93
Figura 7.21 c) Tabla ARP en <i>equipo real 1</i>	93
Figura 7.21 d) Trama nº 15: ping 192.168.2.2 (<i>equipo real 1</i>) → 192.168.2.1 (interfaz SVI)	94
Figura 7.21 e) Trama nº 16 : ping 192.168.2.1 (interfaz SVI) → 192.168.2.2 (<i>equipo real 1</i>)	94
Figura 7.21 f) Tráfico de red capturado en el enlace <i>equipo_real 1</i> → <i>nodo virtual 1</i> , para una longitud de trama <i>Ethernet</i> de 1514 bytes (MTU = 1500 bytes) en la arquitectura WAN ppp	95
Figura 7.22. Esquema administrativo de diseño a nivel 3, según el modelo de referencia OSI, configurado para el análisis de la capacidad y rendimiento del sistema emulador	96
Figura 7.23. Comandos <i>Cisco IOS</i> para la configuración de la subred 155.210.157.240 / 30	98
Figura 7.24. Comandos <i>Cisco IOS</i> para la configuración de la subred 155.210.157.244 / 30	98
Figura 7.25. Topología/simulación gráfica en GNS3 de la arquitectura LAN /VTP.....	102
Figura 7.26. Configuración del nodo <i>emulado EtherSwitch router 1</i> : configuración del enlace troncal, asignación de interfaces <i>FastEthernet</i> a VLAN2 y VLAN3, características VTP	104
Figura 7.27. Creación de VLAN2 y VLAN3 (y sus respectivos interfaces virtuales SVI)	105

Figura 7.28. Configuración de VLAN nativa en el interfaz <i>Fastethernet 1/0</i> asignado al enlace troncal en el nodo emulado <i>EtherSwitch router 1</i>	106
Figura 7.29. Configuración del nodo <i>emulado EtherSwitch router 2</i> : configuración del enlace troncal, asignación de interfaces <i>Fastethernet</i> a VLAN2 y VLAN3, y características VTP.....	106
Figura 7.30. Estado VTP en el nodo emulado <i>EtherSwitch router 1</i> (servidor VTP)... ..	108
Figura 7.31. Estado VTP en el nodo emulado <i>EtherSwitch router 2</i> (cliente VTP).....	108
Figura 7.32. Relación de la información transmitida y recibida por el servidor emulado VTP	109
Figura 7.33. Relación de la información transmitida y recibida por el cliente emulado VTP.....	109
Figura 7.34. Conectividad y trayectoria en la conmutación VLAN (en <i>switch</i> emulado, nodo 2)	110
Figura 7.35. Conectividad y trayectoria en la conmutación VLAN (en <i>switch multilayer 1</i> emulado)	111
Figura 7.36. Configuración de interfaces <i>serial 0/0</i> asignados a la red 192.168.4.0 / 24 en los nodos <i>EtherSwitch router 1</i> y R3 de la arquitectura de red LAN/VTP	113
Figura 7.37. Configuración de interfaces <i>serial 0/0</i> asignados a la red 192.168.5.0 / 24 en los nodos <i>EtherSwitch router 2</i> y R4 de la arquitectura de red LAN/VTP.....	113
Figura 7.38. Cable <i>rollover</i> o consola	122
Figura 7.39. Configuración de <i>HyperTerminal</i> para la conexión al puerto consola del dispositivo de red	122
Figura 7.40. Esquema general para la integración de un equipo real en el entorno de emulación <i>GNS3/Dynamips</i> , vía interfaz <i>Ethernet</i>	124
Figura 7.41. Contexto gráfico inicial para la configuración de los parámetros básicos y fundamentales de <i>GNS3/Dynamips</i>	125
Figura 7.42. Contexto gráfico para la integración de dispositivos reales en <i>GNS3/Dynamips</i>	126
Figura 7.43. Contexto gráfico de configuración en la selección del interfaz de integración	126
Figura 7.44. Contexto gráfico para la configuración del <i>path</i> o ruta de los directorios	127

Figura 7.45. Acceso al menú de configuración de <i>GNS3/Dynamips</i>	128
Figura 7.46. Contexto gráfico para la verificación del <i>path</i> o ruta en la ejecución del proceso <i>dynamips.exe</i> y su directorio de trabajo temporal	129
Figura 7.47. Contexto gráfico para la configuración de los recursos optimizadores de la memoria RAM del sistema emulador	132
Figura 7.48. Vista detallada del directorio/carpeta creada en <i>/home</i> para la instalación de <i>GNS3/Dynamips</i> en Linux	133
Figura 7.49. Vista detallada de la instalación de las librerías <i>python-qt4</i>	134
Figura 7.50. Contexto gráfico de propiedades del archivo binario de la emulación	135
Figura 7.51. Configuración de los directorios de trabajo para <i>GNS3/Dynamips</i> en Linux	136
Figura 7.52. Contexto gráfico para la verificación/configuración del <i>path</i> para la emulación	136
Figura 7.53. Contexto gráfico para la la selección /configuración del interfaz <i>Ethernet</i> (como usuario <i>root</i>) en la <i>interoperación entorno real ↔ entorno emulado</i> en sistemas Linux	137
Figura 7.54 a) Proceso para la configuración/selección del interfaz <i>Ethernet</i> (como usuario <i>root</i>) en la <i>interoperación entorno real ↔ entorno emulado</i> en sistemas Linux	138
Figura 7.54b) Vista detallada del intérprete de comandos Linux de la figura 7.49a)	138
Figura 7.55. Arquitectura de red LAN Ethernet/VTP para analizar la capacidad y rendimiento de <i>GNS3/Dynamips</i> en Linux (idéntica que en Windows)	139
Figura 7.56. Topología en GNS3 de la arquitectura WAN ppp para implementar el servicio cliente/servidor FTP con los roles de servidor y cliente FTP particularizados	144
Figura 7.57. Interfaz gráfica de <i>TCP Optimizer 3.0.8</i> para la configuración del MTU	145
Figura 7.58. Topología en GNS3 de la arquitectura LAN/VTP para analizar el comportamiento del escenario de trabajo en servicios cliente/servidor FTP, con los roles particularizados de servidor y cliente FTP.....	147
Figura 7.59. Trafico de red capturado en servicios FTP <i>cliente/servidor</i> (conexión <i>equipo real 1 → entorno emulado → equipo real 2</i> , de la arquitectura LAN/VTP)	148

Indice de Tablas

Capítulo 1

Tabla 1.1. Distribución del contenido de la memoria por capítulos.....	5
--	---

Capítulo 2

Tabla 2.1. Tabla comparativa de las prestaciones diferenciadas de <i>GNS3</i> y <i>Dynamips</i> ...	10
---	----

Capítulo 4

Tabla 4.1. Configuraciones hardware <i>base</i> para el sistema emulador	26
--	----

Tabla 4.2. Consumo de recursos del sistema emulador para la <i>configuración hardware 1</i> , bajo S.O. Windows XP 32 bits y arquitectura de red WAN ppp	28
--	----

Tabla 4.3. Consumo de recursos del sistema emulador para la <i>configuración hardware 2</i> , bajo S.O. Windows XP 32 bits y arquitectura de red WAN ppp	29
--	----

Tabla 4.4. Latencia de red para el protocolo ICMP en la conexión <i>equipo_real 1</i> → <i>interfaz SVI en nodo virtual 1</i> , para diferentes longitudes de trama y consumo de recursos del PC emulador 1 en la arquitectura WAN ppp	34
--	----

Tabla 4.5. Latencia de red para el protocolo ICMP en la conexión <i>equipo_real 1</i> → <i>interfaz SVI en nodo virtual 1</i> , para diferentes longitudes de trama y consumo de recursos del PC emulador 2 en la arquitectura WAN ppp	34
--	----

Tabla 4.6). Latencia de red imputable a los procesos de emulación con respecto a la latencia total en el enlace <i>equipo_real 1 (Ethernet 802.3)</i> → <i>interfaz SVI en nodo virtual 1</i> , para la arquitectura WAN ppp, protocolo ICMP, diferentes longitudes de trama y 1 <i>switch-router</i> activo	35
--	----

Tabla 4.7. Latencia de red para el protocolo ICMP en la conexión <i>equipo_real 1</i> → <i>red virtual</i> → <i>equipo_real 2</i> , para diferentes longitudes de trama y consumo de recursos del PC emulador 1 en la arquitectura WAN ppp	36
--	----

Tabla 4.8. Latencia de red para el protocolo ICMP en la conexión <i>equipo real 1</i> → <i>red virtual</i> → <i>equipo real 2</i> , para diferentes longitudes de trama y consumo de recursos del PC emulador 2 en la arquitectura WAN ppp	36
Tabla 4.9. Consumo de recursos del sistema emulador para la <i>configuración hardware 1</i> , bajo S.O. Windows XP 32 bits y arquitectura de red LAN/VTP	38
Tabla 4.10. Consumo de recursos del sistema emulador para la <i>configuración hardware 2</i> , bajo S.O. Windows XP 32 bits y arquitectura de red LAN/VTP	39
Tabla 4.11. Latencia de red para el protocolo ICMP en la conexión <i>equipo_real_1</i> → <i>interfaz SVI de nodo virtual 1</i> , para diferentes longitudes de trama y consumo de recursos del PC emulador 1 y arquitectura de red LAN/VTP	41
Tabla 4.12. Latencia de red para el protocolo ICMP en la conexión <i>equipo_real_1</i> → <i>interfaz SVI de nodo virtual 1</i> para diferentes longitudes de trama y consumo de recursos del PC emulador 2 y arquitectura de red LAN/VTP	41
Tabla 4.13. Latencia de red para el protocolo ICMP en la conexión <i>equipo_real_1</i> → <i>entorno emulado</i> → <i>equipo real 2</i> , para diferentes longitudes de trama, consumo de recursos del PC emulador 1 y arquitectura de red LAN/VTP	42
Tabla 4.14. Latencia de red para el protocolo ICMP en la conexión <i>equipo_real_1</i> → <i>entorno emulado</i> → <i>equipo real 2</i> , para diferentes longitudes de trama, consumo de recursos del PC emulador 2 y arquitectura de red LAN/VTP	43

Capítulo 5

Tabla 5.1. Conclusiones en el análisis de la funcionalidad del emulador a nivel 2 y 3, y de la <i>interoperabilidad</i> entre entornos de red reales y emulados	46
Tabla 5.2. Relación de dependencias de la latencia de red en <i>GNS3/Dynamips</i>	48
Tabla 5.3. Conclusiones principales a los análisis de la latencia en la <i>interoperación</i> ...	51
Tabla 5.4. Relación de dependencias de la latencia de red en <i>GNS3/Dynamips</i>	52

Capítulo 7

Tabla 7.1. Tabla resumen de las características y prestaciones principales del protocolo VTP y que se aplican en el diseño particular del escenario de trabajo	107
--	-----

Tabla 7.2. Tabla resumen de ventajas e inconvenientes del emulador <i>GNS3/Dynamips</i> extraídas del análisis de los capítulos 3 ↔ anexo 2, y del análisis de la <i>interoperabilidad</i> del capítulo 4	115
Tabla 7.3. Consumo de recursos del sistema emulador para la <i>configuración hardware 1</i> , bajo S.O. Linux 32 bits y arquitectura de red LAN/VTP	140
Tabla 7.4. Latencia de red para el protocolo ICMP en la conexión <i>equipo_real 1</i> → <i>interfaz SVI de nodo virtual 1</i> , para diferentes longitudes de trama y consumo de recursos del PC emulador 1, arquitectura de red LAN/VTP y sistemas operativos Linux	141
Tabla 7.5. Latencia de red para el protocolo ICMP en la conexión <i>equipo_real 1</i> → <i>entorno emulado</i> → <i>equipo real 2</i> , para diferentes longitudes de trama Ethernet, consumo de recursos del PC emulador 1, arquitectura de red LAN/VTP y sistemas operativos Linux	142
Tabla 7.6. Resultados obtenidos en la transferencia FTP <i>servidor</i> → <i>cliente</i> de <i>archivo.rar</i> (250KB) para distintas longitudes de trama <i>Ethernet 802.3</i> en el análisis de la arquitectura de red WAN ppp	146
Tabla 7.7. Resultados obtenidos en la transferencia FTP <i>servidor</i> → <i>cliente</i> para distintas longitudes de trama <i>Ethernet 802.1q</i> del enlace troncal en la arquitectura LAN /VTP...	147

Capítulo 1: Introducción al proyecto.

1.1. Perspectiva actual del ámbito de estudio.

Debido al grado de competitividad actual y la necesidad de optimizar determinadas variables asociadas al diseño del proceso productivo, como bien pueden ser los costes de producción, es necesario disponer de herramientas de trabajo adecuadas y ajustadas a los niveles de exigencia del mundo profesional/laboral. Desde la perspectiva de las comunicaciones en un entorno corporativo de mayor o menor jerarquía, el diseño y la infraestructura de la red de comunicaciones es un punto clave y ciertamente crítico para el correcto desempeño de cualquier proceso productivo. Siendo así, resulta evidente que cualquier actuación y/o actualización organizativa en la infraestructura de trabajo repercute directamente en la estructura y “*rediseño*” de la red de comunicaciones y su escalabilidad. Por ello, disponer de herramientas adecuadas que reproduzcan y emulen el comportamiento y funcionalidad de la red de comunicaciones del entorno de trabajo en cuestión, minimizan el impacto directo sobre los procesos de producción y costes asociados.

Debe contemplarse que el empleo de cualquier herramienta de diseño y emulación que pretenda reproducir un escenario real de trabajo presenta un margen de fiabilidad que es preciso analizar, acotar y optimizar, desde el rol ejercido por un ingeniero de telecomunicación en todo el proceso anteriormente mencionado de diseño, optimización y/o actualización de la red de comunicaciones de una organización o empresa.

Acorde con este objetivo, el abanico de posibilidades tecnológicas que a priori se presenta en la elección de una herramienta de trabajo idónea, se puede antojar denso y farragoso. Por ello, es necesario un riguroso análisis de las condiciones y exigencias de trabajo que se plantean: *qué* objetivos queremos conseguir y *cómo* alcanzar óptimamente los mismos. Consecuentemente, y añadido a la imposibilidad de acometer un estudio global por su profundidad, se particulariza el mismo sobre el hardware de red *Cisco Systems* presente en el laboratorio de Telemática del departamento IEC y las herramientas software proporcionadas por el entorno de emulación de redes *GNS3/Dynamips*.

Así, los planteamientos anteriormente descritos son los que desembocan en la elaboración del presente proyecto que trata de analizar y extraer oportunas conclusiones acerca de la idoneidad, márgenes de aplicación y normas/recomendaciones de uso para las herramientas antes mencionadas y potencialmente empleadas en el diseño, análisis y optimización de la red de comunicaciones que toda organización precisa. Herramientas, claro está, puestas a nuestra disposición en el contexto tecnológico actual.

1.2. Fundamentos y herramientas de trabajo. Objetivos del proyecto.

Expuestas las necesidades que condicionan la elección de nuestras herramientas de trabajo, se prioriza la necesidad de disponer de un entorno software de diseño y análisis con capacidad de reproducir (rigurosamente) una determinada arquitectura de red. Estas motivaciones conducen a optar por herramientas software que aúnen características de simulación y principalmente de emulación, y con una deseable capacidad para *interoperar* con tráfico de red y equipos reales.

El centro del análisis va a ser el entorno software conformado por el interfaz gráfico/simulador *GNS3* y el software, motor de la emulación, *Dynamips* (integrando además la herramienta software para la captura de tráfico *Wireshark*). Se pretende analizar y constatar empíricamente las prestaciones, rigor y correspondencia funcional de *GNS3/Dynamips* con entornos de red reales, así como dimensionar y acotar márgenes fiables de trabajo. Siendo así, se consolidan los siguientes objetivos generales en el análisis del proyecto.

1.2.1. Adquisición de habilidades *vía* entorno software *GNS3/Dynamips*.

Un objetivo fundamental de una herramienta software de simulación/emulación debe ser el de poder adquirir las habilidades necesarias sobre un escenario real de trabajo que permitan, llegado el momento, acometer óptimamente el trabajo de campo. Por esta razón, el conocimiento del entorno software *GNS3/Dynamips*, así como del sistema operativo *Cisco IOS* (y en menor medida *Cisco CatOS* [3] [16]), motor de los dispositivos de red *Cisco Systems* reales y emulados implicados en el estudio, es el primer objetivo y punto de partida en el desarrollo del proyecto.

1.2.2. Análisis del hardware de red *Cisco Systems* y sistemas operativos *Cisco IOS* y *Cisco CatOS*. *Interoperabilidad* de *GNS3/Dynamips* con entornos de red reales.

Para el correcto análisis de nuestros objetivos, y en consonancia con el apartado anterior, se hace necesario un proceso de adquisición de conocimientos y habilidades del entorno hardware *Cisco Systems* en general, y particularmente del dispositivo de red modular *Cisco Catalyst 5500 Swicth* presente en el laboratorio de telemática, así como de los sistemas operativos *Cisco IOS* y *Cisco CatOS*, a él vinculados. Consecuencia de estos análisis es el anexo 1 de la memoria, que permite aplicar estos conocimientos y habilidades sobre ambos escenarios de trabajo, real y emulado en *GNS3/Dynamips*.

El conocimiento de estos sistemas es prioritario para establecer la configuración del hardware conforme a una determinada arquitectura de red, tanto en el entorno real como en el emulado. Además, se realiza un análisis exhaustivo de la funcionalidad y rendimiento de los

módulos hardware *Cisco WS-X5302* (que implementa funcionalidades de *routing* y *switching* a nivel 2 y 3) y *Cisco WS-X5530-E3* (módulo supervisor que comporta características fundamentalmente de gestión y administración hardware), ambos módulos integrados en nuestro hardware modular *Cisco Catalyst 5500 Switch* del laboratorio [15][16].

El análisis de la potencial *interoperabilidad* de *GNS3/Dynamips* con hardware y tráfico de red real se plantea como objetivo prioritario del análisis a desarrollar. Para ello, el anexo 3 explica detalladamente la configuración y técnica precisas para configurar la *interacción/interoperación* de los entornos de red real y emulado. Seguidamente, se analiza dicha *interoperación*: funcionalidad, límites y correspondencia funcional del tráfico de red emulado respecto del tráfico real. En nuestro caso, los dispositivos de red emulados pretendemos que sean “imagen” de los dispositivos de red *Cisco Systems* disponibles (o similares) en el laboratorio de telemática. Por lo tanto, el desarrollo del proyecto analiza la emulación en cuanto a la funcionalidad y prestaciones de dichos equipos reales participantes en dos diferentes arquitecturas de red, WAN ppp y LAN Ethernet/VTP.

Para proceder a la emulación de dispositivos de red *Cisco Systems* en *GNS3/Dynamips* es necesario extraer de los dispositivos de red reales su sistema operativo *Cisco IOS* e integrarlo posteriormente en el emulador, según se explica en el anexo 3.

1.2.3. Extensión y aplicación al ámbito académico.

Hay que destacar dentro de los objetivos de estudio previamente planteados, la posible idoneidad y aplicación del entorno software *GNS3/Dynamips* al ámbito académico, añadido al ámbito profesional/laboral.

Recordemos que *GNS3/Dynamips* es un entorno software que en principio presenta las mismas características y connotaciones que el entorno real emulado, extrayendo su rendimiento del mismo sistema operativo. Así, mediante el planteamiento y estudio de escenarios que emulan diferentes arquitecturas de red, el posterior desarrollo teórico y práctico de las configuraciones y parámetros de diseño de las mismas, unido al análisis del rendimiento del sistema emulador y del tráfico de red, se pretenden extraer conclusiones sobre la *usabilidad* y limitaciones del entorno de emulación, no sólo dentro de un marco de trabajo profesional, sino también para el ámbito académico.

1.2.4. Evaluación de la capacidad y rendimiento del sistema hardware emulador (PC emulador). Influencia sobre la latencia de red en la *interoperación entorno real ↔ entorno emulado* y sobre la potencial escalabilidad de las arquitecturas de red WAN ppp y LAN Ethernet/VTP.

La línea argumental al respecto gira en torno a los equipos *hardware* (2 tipos de configuraciones PC diferentes) y sistemas operativos (Windows fundamentalmente y Linux) responsables de ejecutar los procesos de emulación y, llegado el momento, de la

interoperación entre los entornos de red real y emulado. Se pretende así evaluar la capacidad y rendimiento del sistema emulador (principalmente, la carga del procesador y la memoria RAM consumida) y las limitaciones de diseño y escalabilidad para poder ejecutar correctamente los procesos asociados a la emulación.

Además, se busca establecer dependencias y relaciones entre el rendimiento asociado a configuraciones concretas del sistema emulador (PC de características hardware determinadas, con S.O *Windows o Linux*) para las arquitecturas de red propuestas, y su posible incidencia sobre la latencia de red a la hora de *interoperar* dispositivos reales y emulados.

1.3. Contenido esquemático de la memoria del proyecto. Distribución de los contenidos por capítulos.

Establecida la carga teórica del análisis desarrollado, su reflejo en la presente memoria se concreta esquemáticamente en la *tabla 1.1* de la página siguiente.

Contenidos del proyecto	Contenido teórico/práctico del capítulo
Capítulo 1	<ul style="list-style-type: none"> • Introducción al proyecto y objetivos del mismo.
Capítulo 2	<ul style="list-style-type: none"> • Introducción teórica a los entornos controlados hardware y software del laboratorio. • Escenario de trabajo para el análisis, problemas y soluciones.
Capítulo 3	<ul style="list-style-type: none"> • Configuración y análisis en el entorno emulado de las arquitecturas de red WAN ppp y LAN Ethernet/VTP relacionadas con el escenario de trabajo del capítulo 2. Primeras aproximaciones a la <i>interoperación entorno real ↔ entorno emulado</i>. • Se apoya en los anexos 1 y 2, y en menor medida en el anexo 3.
Capítulo 4	<ul style="list-style-type: none"> • Capacidad del emulador, análisis del rendimiento y prestaciones para dos <i>configuraciones hardware</i> distintas del sistema emulador (PC con distintos procesadores y memoria RAM) y S.O Windows. Influencia del consumo de recursos del sistema sobre la potencial escalabilidad de la arquitectura y sobre la latencia de red. • Análisis intensivo de la <i>interoperabilidad</i> del entorno emulado <i>GNS3/Dynamips</i> con equipos y tráfico de red reales. • Análisis de resultados particulares del capítulo. • Se apoya fundamentalmente en los anexos 2 y 3.
Capítulo 5	<ul style="list-style-type: none"> • Análisis y conclusiones a los objetivos prioritarios del proyecto. Inconvenientes y líneas de trabajo y/o investigación futuras.
Capítulo 6	<ul style="list-style-type: none"> • Referencias bibliográficas y <i>en línea</i> de apoyo en el proyecto.
Capítulo 7: Anexos	<ul style="list-style-type: none"> • Elaborados <i>ex profeso</i> para el desarrollo del proyecto. Los anexos 1, 2 y 3 se consideran fundamentales para el correcto desarrollo y análisis del proyecto. Los anexos 4 y 5 se elaboran como complemento a los análisis prioritarios realizados.
Anexo 1	<ul style="list-style-type: none"> • Configuración & gestión de dispositivos de red hardware modulares <i>Cisco Catalyst 5500 Switch</i> (y similares). (La elaboración de este anexo 1 ha servido para configurar los equipos <i>Cisco Catalyst 5500 Switch</i> utilizados en el laboratorio de telemática 2.03).
Anexo 2	<ul style="list-style-type: none"> • Configuraciones de red en arquitecturas emuladas en <i>GNS3/Dynamips</i>. Interoperación <i>entorno real ↔ entorno emulado</i>.
Anexo 3	<ul style="list-style-type: none"> • Extracción del sistema operativo <i>Cisco IOS</i> de hardware de red <i>Cisco Systems</i>. Integración de <i>Cisco IOS</i> en el entorno software de emulación <i>GNS3/Dynamips</i>. Configuración básica de <i>GNS3/Dynamips</i>, parámetros y prestaciones fundamentales.
Anexo 4	<ul style="list-style-type: none"> • Análisis de prestaciones y rendimiento del sistema emulador en entornos de trabajo Linux (distribución Ubuntu 12.04).
Anexo 5	<ul style="list-style-type: none"> • Implementación de un servicio telemático cliente /servidor FTP vía <i>interoperación entorno real (Ethernet 802.3) ↔ entorno emulado</i> en arquitecturas de red emuladas.

Tabla 1.1. Distribución del contenido de la memoria por capítulos.

Capítulo 2: Consideraciones de diseño y entorno de trabajo

2.1. Consideraciones generales del proyecto.

El desarrollo del proyecto descansa sobre dos entornos controlados de trabajo: el entorno hardware constituido principalmente por el dispositivo de red *router-switch multilayer Cisco Catalyst 5500 Switch* del laboratorio de telemática, y el entorno software simulador/emulador *GNS3/Dynamips* sobre el que se plantean diseños de arquitecturas emuladas potencialmente implementadas sobre escenarios reales de trabajo. El dispositivo *Cisco Catalyst 5500 Switch* es sometido a un intenso análisis para evaluar sus potenciales prestaciones telemáticas. Fruto de este análisis es el anexo 1 de la memoria, que permite la configuración del dispositivo para su uso real en el laboratorio y en el entorno emulado. Este hardware de red sirve igualmente como banco de pruebas en el proceso de comprensión y adquisición de habilidades sobre entornos de trabajo integrados por hardware de red *Cisco Systems* y sus sistemas operativos asociados, principalmente *Cisco IOS*. Estos conocimientos se aplican posteriormente sobre los entornos emulados.

2.2. Consideraciones del entorno hardware de red *Cisco Systems*.

Dentro del contexto tecnológico actual de PYMES y grandes entornos corporativos, el hardware de red *Cisco Systems* comprende una amplia gama de productos y servicios abarcando todas las potenciales necesidades telemáticas existentes, desde las más básicas y elementales hasta los contextos de diseño más avanzados de nivel de especialidad máximo. Actualmente, *Cisco Systems* puede considerarse líder destacado en su sector del mercado de las telecomunicaciones. Aunque aspectos concretos de los análisis del proyecto pretenden extrapolarse a escenarios telemáticos generales, abstrayéndonos en cierta medida del carácter corporativo del diseño, el desarrollo del proyecto se apoya fundamentalmente en hardware/software *Cisco Systems*, con sus características intrínsecas y particularidades, trasladables posteriormente al entorno emulado. Siendo así, estamos en condiciones de asegurar que el entorno hardware utilizado en el presente proyecto se adscribe a un contexto hardware real de trabajo de elevada y contrastada presencia en el escenario telemático laboral y profesional actual a todos los niveles.

Las particularidades de diseño y configuración de los entornos de trabajo integrados por dispositivos de red *Cisco Systems* se abordan de una manera general en el anexo 1. El análisis del dispositivo *Cisco Catalyst 5500 Switch*, además de abordar las perspectivas particulares para el desarrollo del presente proyecto, permite adquirir conocimientos y habilidades sobre el uso general del hardware de red *Cisco Systems*, y sobre todo de su sistema operativo *Cisco IOS*, imprescindible para la administración y configuración de los escenarios telemáticos de trabajo real y emulado. Las connotaciones y particularidades de diseño en el entorno de red real se trasladan posteriormente al entorno software de *GNS3/Dynamips*.

El hardware de red *Cisco Catalyst 5500 Switch* [7] utilizado en el proyecto es una arquitectura hardware modular muy versátil a nivel de diseño y prestaciones que presenta, entre otras, características *multilayer* que permiten su uso en arquitecturas de red que implementan *InterVLAN routing* [3], esto es, commutación a nivel 2 y 3 sobre escenarios de trabajo segmentados VLAN, además de implementar características de *routing*. El anexo 1 sirve para profundizar en estas características funcionales y prestaciones.

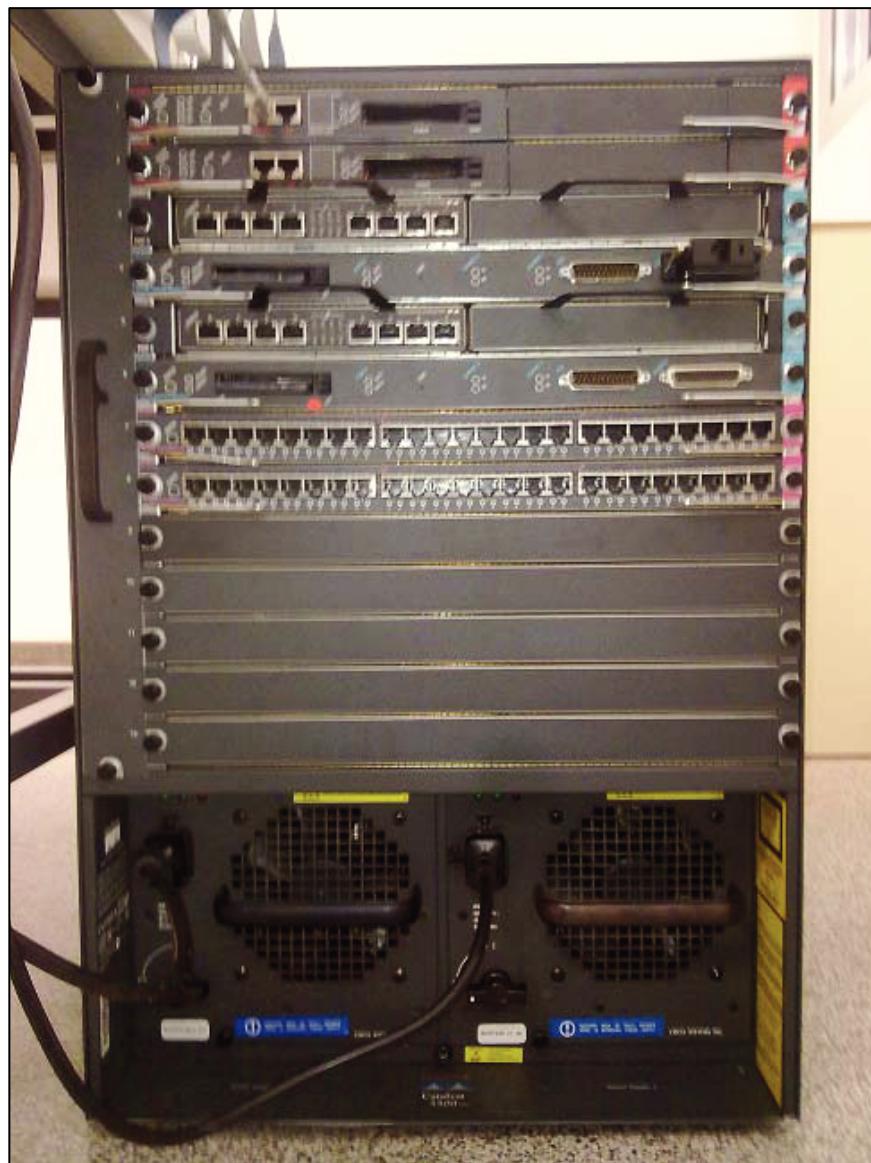


Figura 2.1. Dispositivo de red *Cisco Catalyst 5500 Switch* del laboratorio de telemática del DIEC.

2.3. Consideraciones del entorno software emulador *GNS3/Dynamips*.

Adscrito al entorno general de trabajo para el desarrollo del proyecto, el entorno software controlado de trabajo corresponde al entorno software simulador/emulador *GNS3/Dynamips* [12].

Resumidamente, el entorno software *GNS3/Dynamips* está conformado por un interfaz gráfico para la implementación de arquitecturas de red, que constituye la parte de simulación gráfica del software. Sobre dicho interfaz gráfico se implementan arquitecturas de red que abarcan desde diseños básicos y elementales, hasta diseños telemáticos más avanzados, integrando dispositivos ATM, Frame Relay, etc., así como dispositivos de seguridad *Firewall*. Pero la característica fundamental de este entorno software es que permite la emulación de dispositivos de red *Cisco Systems* (también *Juniper*) en base a los sistemas o imágenes *Cisco IOS* de los análogos dispositivos reales. Para el correcto funcionamiento de los dispositivos de red emulados hay que extraer por lo tanto el sistema operativo *Cisco IOS* de dispositivos de red reales, y posteriormente incorporarlo en los emulados, según las indicaciones del anexo 3 de la memoria. Esta es la característica más relevante y decisiva de *GNS3/Dynamips* y donde radica toda su funcionalidad y potencia de emulación, empleando en la misma idéntico sistema operativo que los dispositivos reales análogos. La emulación se implementa a través del software *Dynamips*, auténtico motor de la emulación. Además, el contexto gráfico/interfaz asociado a la consola emulada permite al usuario acceder a los dispositivos emulados como si accediese a los análogos dispositivos reales *vía* puerto consola.

Una de las características fundamentales que ofrece este entorno simulador/emulador, y que le diferencian de otras plataformas emuladoras, es que permite la *interoperación* de dispositivos de red reales y emulados (consecuentemente, tráfico de red real y emulado) sobre un mismo escenario de trabajo. Así se analiza en el desarrollo del proyecto como objetivo prioritario del mismo, con las consiguientes ventajas que esto puede reportar en ámbitos académicos y profesionales. Ineludiblemente, para el adecuado desarrollo del proyecto, y el correcto manejo de *GNS3/Dynamips*, son necesarios e imprescindibles conocimientos generales de administración, gestión y diseño de redes, así como una base consistente de conocimientos y habilidades para la gestión, administración y configuración de entornos de red hardware de *Cisco Systems*.

En esta introducción teórica se hace necesaria una breve aclaración conceptual sobre la connotación inherente a los términos simulador (básicamente, un *front end* o entorno gráfico de diseño) y emulador. Dada la variedad de acepciones asociadas con las características soportadas por un sistema emulador, sistema cuyo cometido fundamental debe ser reproducir un entorno real de trabajo, se centrará el análisis del comportamiento del emulador en el tratamiento que éste hace del tráfico de red: si es paquete a paquete de información, y si además (o sólo) realiza un tratamiento en tiempo real, más o menos óptimo. Al respecto, la *tabla 2.1* siguiente establece una comparativa general entre las características y prestaciones diferenciadas de *GNS3* y *Dynamips*.

Comparativa de prestaciones de GNS3 y Dynamips	Descripción de prestaciones
GNS3	<ul style="list-style-type: none"> • Básicamente un <u>entorno gráfico</u> (<i>front end</i> gráfico de <i>Dynagen</i>, simulador gráfico). Facilidad de implementación y uso (adecuada <i>usabilidad</i>). • Diseño e implementación de topologías de red basadas en arquitecturas y tecnologías diversas con soporte en hardware de red <i>Cisco Systems</i> y <i>Juniper</i> (actualmente). • Permite incorporar en el diseño dispositivos de red generales (no corporativos, como <i>switch Ethernet</i>, ATM y Frame Relay)
Dynamips	<ul style="list-style-type: none"> • Potente motor de <u>emulación</u> software integrado en el entorno gráfico <i>GNS3</i> para reproducir/emular sobre un PC el funcionamiento de dispositivos de red, análogos de dispositivos reales. • Sólo emula, actualmente, sistemas operativos <i>Cisco IOS</i>, unilateralmente vinculados a hardware de red <i>Cisco Systems</i>, además de sistemas <i>JunOS</i>, paralelismo extendido a hardware de red <i>Juniper</i>. • Permite la <i>interoperación</i> de entornos de red reales y emulados a través de un interfaz <i>Ethernet</i>.

Tabla 2.1. Tabla comparativa de las prestaciones diferenciadas de *GNS3* y *Dynamips*.

El entorno software simulador/emulador *GNS3/Dynamips* se integra actualmente dentro de un proyecto software *open source* en constante evolución y actualización que incorpora periódicamente actualizaciones y mejoras [12]. Existen versiones para sistemas operativos Windows, Linux y MacOS. Lejos de pretender elaborar un manual de uso sobre el mismo, para lo que se aportan las correspondientes referencias bibliográficas y *en línea*, sí se incluyen ciertos aspectos relevantes que resultan fundamentales en el desarrollo del proyecto y que permiten no perder el hilo del análisis, en consonancia y paralelamente con el contenido de los anexos. Estos anexos son elaborados *ex profeso* para el correcto desarrollo y análisis de los capítulos 3 y 4, corazón del proyecto y la memoria, y en base al análisis intensivo de los entornos controlados de laboratorio, hardware (*Router - Switch multilayer Cisco Catalyst 5500*) y software (*GNS3/Dynamips* y sistemas operativos *Cisco IOS* y *Cisco CatOS*).

2.4. Escenario de trabajo general para el análisis. Contingencias en el diseño y soluciones adoptadas.

Con el objetivo de analizar los propósitos del proyecto descritos en el capítulo 1, se plantea un escenario común de trabajo consistente en el diseño y configuración de un mismo entorno VLAN de trabajo (segmentación en VLAN2 y VLAN3) para dos ubicaciones físicas diferentes, interconectadas entre sí. Se realizará la conexión física y lógica de las dos ubicaciones según los propósitos administrativos de diseño VLAN y conforme a dos arquitecturas de red diferentes: arquitectura WAN ppp (*point-to-point*) y arquitectura commutada LAN Ethernet/VTPL. Este escenario virtual de trabajo, implementado sobre el emulador *GNS3/Dynamips*, correspondería a la “virtualización” de un entorno real de trabajo similar, en el que las ubicaciones físicas corresponderían a las *bancadas 1-2* y *3-4* del laboratorio de telemática 2.03, como refleja la figura 2.2.

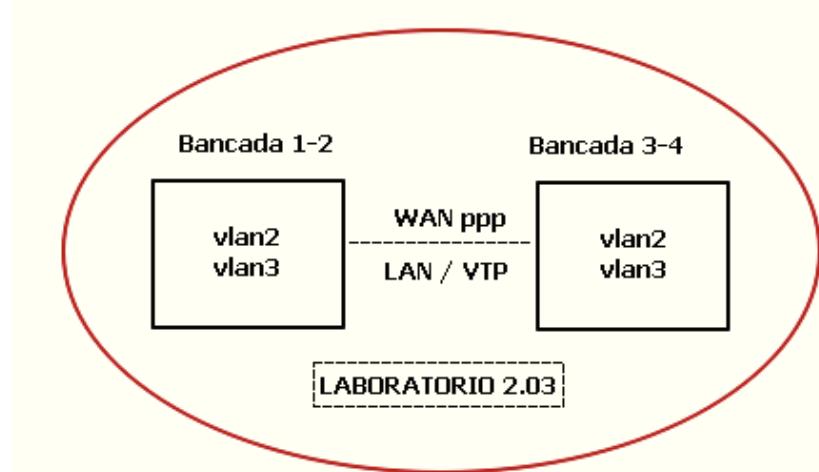


Figura 2.2. Escenario general de trabajo para el análisis del emulador *GNS3/Dynamips*.

Todo el entorno controlado de trabajo, tanto el real como el emulado, se apoya principalmente en el dispositivo de red *Catalyst 5500 Switch* de *Cisco Systems*, presente en el laboratorio anterior. Dicho dispositivo ha sido objeto de un intenso análisis para su comprensión y la adquisición de habilidades sobre él mismo, y de un modo general, sobre el hardware de red *Cisco Systems*, además de sus sistemas operativos *Cisco IOS* y *CatOS*. Así, este estudio hardware & software, que da lugar al anexo 1, nos permitirá la posterior explotación de posibilidades telemáticas, tanto en entornos de red reales del laboratorio como en su correspondiente emulación bajo *GNS3/Dynamips*, software que se articula fundamentalmente sobre hardware de red *Cisco Systems*. Igualmente, cabe mencionar que este análisis, y la consiguiente elaboración del anexo 1, ha servido para configurar los dispositivos *Cisco Catalyst 5500 Switch* actualmente utilizados en el laboratorio de telemática del Departamento de Ingeniería Electrónica y Comunicaciones (DIEC).

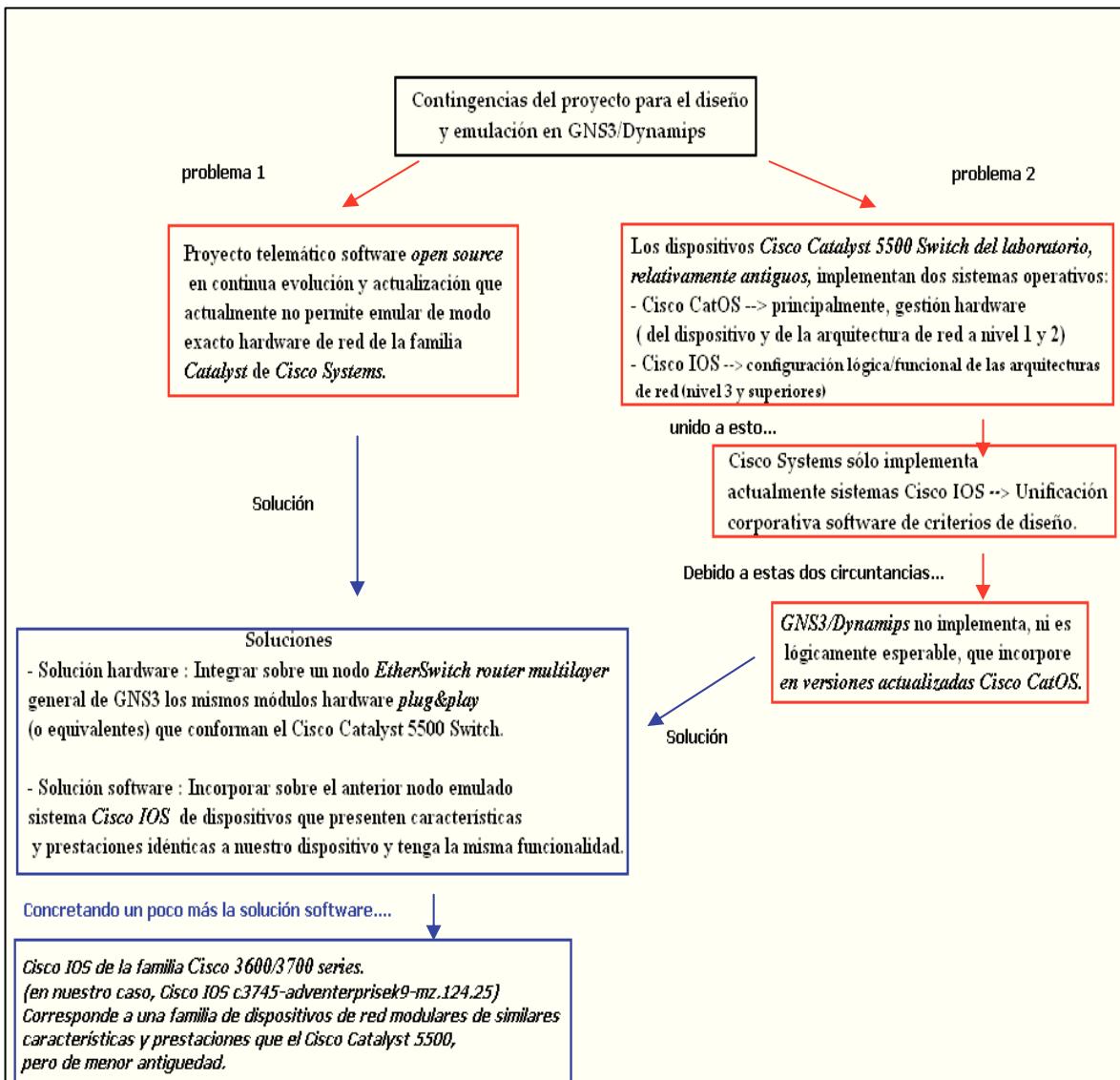


Figura 2.3. Esquema de contingencias y soluciones adoptadas para implementar las arquitecturas de red del proyecto en *GNS3/Dynamips* en base al dispositivo *Cisco Catalyst 5500 Switch* del laboratorio.

La figura 2.4 muestra el dispositivo/nodo de red general emulado *EtherSwitch router* disponible en *GNS3/Dynamips*, que será la base del diseño de las arquitecturas de red emuladas. Sobre éste se integran (conforme a las indicaciones/soluciones adoptadas a las contingencias planteadas en la figura 2.3 anterior) el sistema *Cisco IOS* de la plataforma *Cisco 3600/3700 series* [9] y los módulos de interfaces *virtuales-hardware* necesarios para conformar un dispositivo de red emulado idéntico o similar, tanto hardware como software, al dispositivo de red modular *Cisco Catalyst 5500 Switch* del laboratorio.

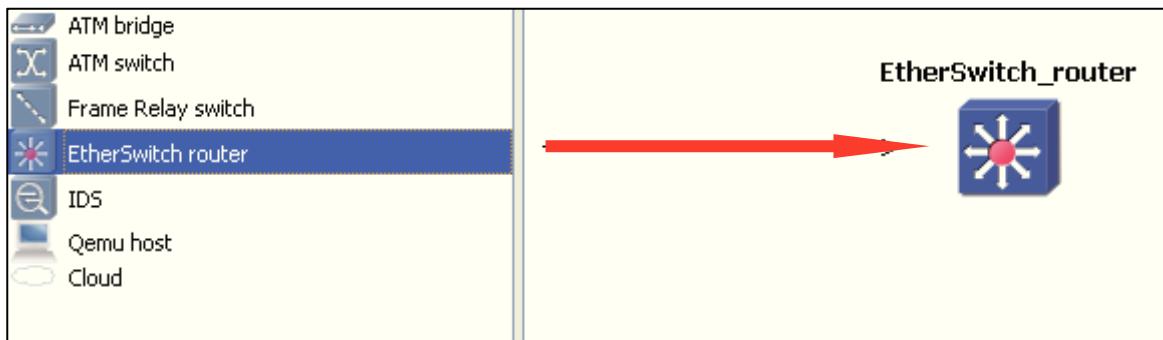


Figura 2.4. Nodo *EtherSwitch router* general disponible en el entorno de emulación *GNS3/Dynamips*.

Capítulo 3: Análisis de arquitecturas de red en GNS3/Dynamips.

3.1. Análisis de la arquitectura de red WAN ppp.

3.1.1. Consideraciones iniciales y descripción para la configuración de la arquitectura de red.

Conforme al escenario común de trabajo descrito en el *apartado 2.4*, se procede al planteamiento y diseño de la arquitectura de red en el entorno gráfico de simulación *GNS3*, arquitectura que realiza la interconexión (a nivel de red) de las VLAN configuradas en las *bancadas 1-2* y *3-4* del laboratorio de telemática 2.03 (localmente, en cada *router-switch*) según la *figura 3.1*.

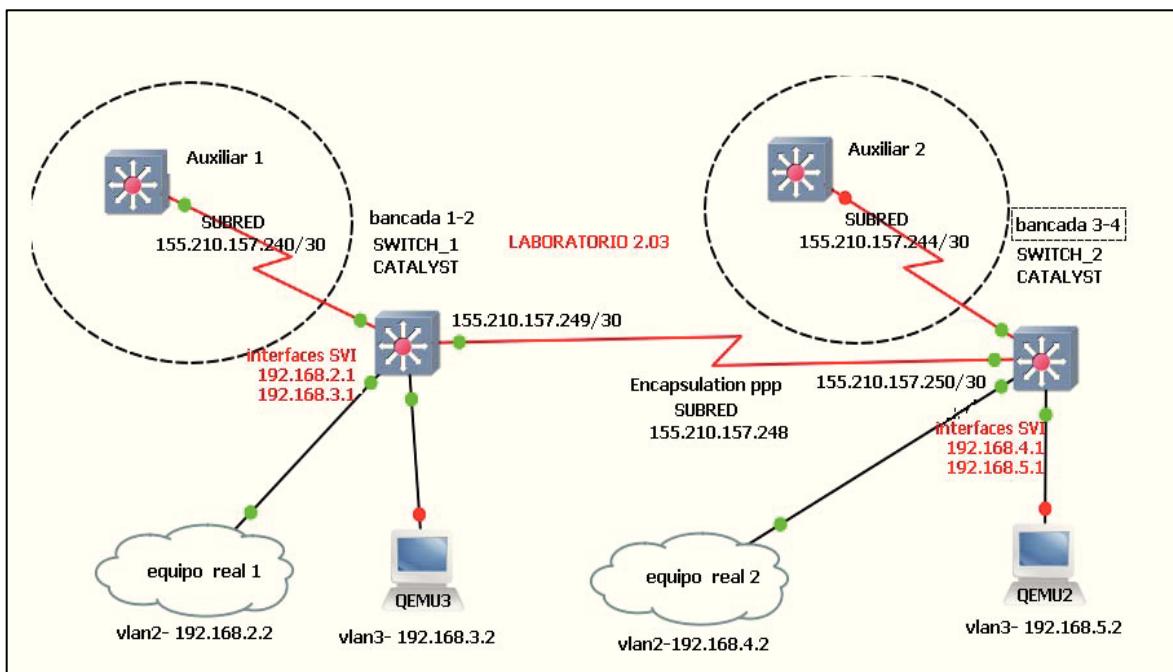


Figura 3.1. Topología/simulación gráfica en *GNS3* de la arquitectura de red WAN PPP.

Inicialmente, para configurar gráficamente sobre *GNS3* la topología de la *figura 3.1* anterior, el anexo 2 (*apartado 7.2.1*) muestra en profundidad la manera de proceder.

Seguidamente, para los análisis de este capítulo 3, cuyo cometido principal es verificar la *usabilidad*, funcionalidad y rigor del entorno emulador sólo consideraremos activos y configuraremos los nodos adscritos a la subred **155.210.157.248 / 30** (enlace WAN PPP de 1,544 MB, esto es, el enlace serie WAN PPP que conectaría las *bancadas 1-2* y *3-4* del laboratorio 2.03). Esta es considerada nuestra arquitectura de red principal.

Se consideran los siguientes aspectos de diseño sobre el escenario de la *figura 3.1*:

- Para que se produzca intercambio de la información de *routing* entre los dispositivos de red *EtherSwitch router*, con el objetivo de informarse mutuamente de sus respectivas “informaciones de red”, y con esto configurar cada uno sus tablas de enrutamiento, se activará posteriormente el protocolo RIP v2 en los interfaces pertinentes [8] (en sistemas basados en *Cisco IOS*, los protocolos de enrutamiento están desactivados por defecto; se configuran y activan a nivel de interfaz, sobre los interfaces adecuados según los propósitos del diseño).
- Los equipos reales, integrados en el escenario mediante enlaces *FastEthernet* se adscriben administrativamente y localmente a VLAN2, con direcciones IP y máscaras asignadas según el diseño particular de la *figura 3.1*. Los equipos QEMU son terminales/*host* virtuales Linux y se consideran a efectos de diseño adscritos localmente a VLAN3, en el espacio de direcciones de red indicado.
- Según esta arquitectura de red, la interconexión de los equipos *EtherSwitch router* que emulan los dispositivos *Cisco Catalyst 5500 Switch* y que en el entorno real conectarían las *bancadas 1-2* y *3-4* del laboratorio 2.03, se realiza mediante un enlace WAN *serie point-to-point* de 1,544 MB, adscrito administrativamente a la subred **155.210.157.248 / 30**, subred de la red mayor **155.210.0.0 / 16**. Los dispositivos de red emulados *EtherSwitch router* (descritos en el *apartado 2.4*) son hardware de red con capacidad para operar en los niveles 2 y 3 del modelo de referencia OSI (*router-switch multilayer*): añadidas a las características de *routing*, además de implementar conmutación a nivel 2 entre equipos adscritos a una misma VLAN en un nodo local *EtherSwitch router*, nos permiten realizar *InterVLAN routing* (conmutación a nivel 3) entre VLAN diferentes (en un mismo nodo local) mediante la configuración de interfaces virtuales SVI (sin necesidad de utilizar para ello un *router* externo y un enlace troncal configurado con el protocolo 802.1q ó ISL). Para habilitar las funcionalidades a nivel 3 en los *router-switch multilayer* (sino, el dispositivo sólo funcionaría a nivel 2) no debemos olvidar durante el proceso de configuración, y desde el *modo de configuración global* (los *modos de configuración* del hardware de red *Cisco Systems* se detallan en el anexo 1), ejecutar el comando *Cisco IOS*:

```
switch_router1(config)# ip routing
```

A tal efecto, para establecer las configuraciones de red *vía* consola de los dispositivos emulados mediante comandos *Cisco IOS*, y tomando como base los análisis reflejados en el anexo 1, el anexo 2 detalla más profundamente cómo proceder (*apartado 7.2.1*).

3.1.2. Captura y análisis de tráfico de red emulado en el enlace serie WAN ppp.

Una vez implementada la configuración de red para el escenario de trabajo actual, configuración detallada en el anexo 2 (*apartado 7.2.1*), y para ahondar en el análisis de la funcionalidad y la correspondencia funcional de *GNS3/Dynamips* con entornos de red reales, se realiza una captura del tráfico existente para su análisis mediante la herramienta software *Wireshark* [14]. Así, un análisis del tráfico en la conexión *serie WAN ppp* principal configurada entre los nodos *EtherSwitch router 1* y *2* (subred **155.210.157.248/30** de la *figura 3.1*) constata la existencia de tráfico de red relacionado con los protocolos en funcionamiento y activos en el escenario de trabajo emulado (*figura 3.2*). Se verifica, por lo tanto, que la actualización de las tablas de enrutamiento en los nodos emulados se ha producido a través de información (tráfico de red emulado) que circula por el entramado de la red virtual.

No.	Time	Source	Destination	Protocol	Length	Info
14	20.515000	N/A	N/A	PPP LCP	16	Echo Reply
15	30.718000	N/A	N/A	PPP LCP	16	Echo Request
16	30.750000	N/A	N/A	PPP LCP	16	Echo Reply
17	30.750000	N/A	N/A	PPP LCP	16	Echo Request
18	30.781000	N/A	N/A	PPP LCP	16	Echo Reply
19	37.328000	155.210.157.250	224.0.0.9	RIPv2	76	Response
20	40.953000	N/A	N/A	PPP LCP	16	Echo Request
21	40.968000	N/A	N/A	PPP LCP	16	Echo Reply
22	41.000000	N/A	N/A	PPP LCP	16	Echo Request
23	41.015000	N/A	N/A	PPP LCP	16	Echo Reply
24	47.312000	155.210.157.249	224.0.0.9	RIPv2	76	Response
25	51.187000	N/A	N/A	PPP LCP	16	Echo Request
26	51.203000	N/A	N/A	PPP LCP	16	Echo Reply
27	51.250000	N/A	N/A	PPP LCP	16	Echo Request
28	51.265000	N/A	N/A	PPP LCP	16	Echo Reply
29	52.156000	N/A	N/A	CDP	333	Device ID: switch_router2 Port ID: serial2/0
30	53.140000	N/A	N/A	CDP	333	Device ID: switch_router1 Port ID: serial2/0
31	61.437000	N/A	N/A	PPP LCP	16	Echo Request
32	61.453000	N/A	N/A	PPP LCP	16	Echo Reply
33	61.468000	N/A	N/A	PPP LCP	16	Echo Request
34	61.500000	N/A	N/A	PPP LCP	16	Echo Reply
35	63.843000	155.210.157.250	224.0.0.9	RIPv2	76	Response
36	71.672000	N/A	N/A	PPP LCP	16	Echo Request

Figura 3.2. Tráfico de red en el enlace *serie WAN ppp* del entorno emulado.

El tráfico capturado en el enlace *serie WAN ppp* pone de manifiesto la comunicación existente entre los nodos *EtherSwitch router* en el entorno emulado. En primer lugar, se constata la existencia de tráfico de red relacionado con RIP v2, por ejemplo, en las tramas capturadas 19, 24 y 35: los nodos *EtherSwitch router* envían información de sus tablas de enrutamiento con dirección de destino *multicast 224.0.0.9*. En segundo lugar, se aprecia tráfico relacionado con el protocolo point-to-point configurado en el enlace *serie WAN ppp*, tráfico éste que permite el mantenimiento de la conexión serie. También se aprecia tráfico relacionado con el protocolo CDP. CDP (*Cisco Discovery Protocol*) es un protocolo de nivel 2 propietario de *Cisco Systems*, configurado por defecto en sus dispositivos, que permite el intercambio de información entre los mismos. Dicho protocolo puede ser desactivado a voluntad del administrador del sistema. Al respecto, un análisis más exhaustivo del tráfico de red capturado (de la información contenida en las tramas anteriores) se acomete en el anexo 2 (*apartado 7.2.2*).

3.2. Análisis de la arquitectura de red LAN Ethernet / VTP

3.2.1. Consideraciones iniciales y descripción para la configuración de la arquitectura de red.

Se procede a diseñar y configurar el escenario común de trabajo del *apartado 2.4* según la arquitectura de red actual, consistente en una infraestructura LAN Ethernet/VTP conmutada sobre las dos ubicaciones físicas (*bancada 1-2* y *bancada 3-4*), segmentada en dos VLAN, VLAN2 y VLAN3 (además de una VLAN nativa), como recoge esquemáticamente la *figura 3.3*.

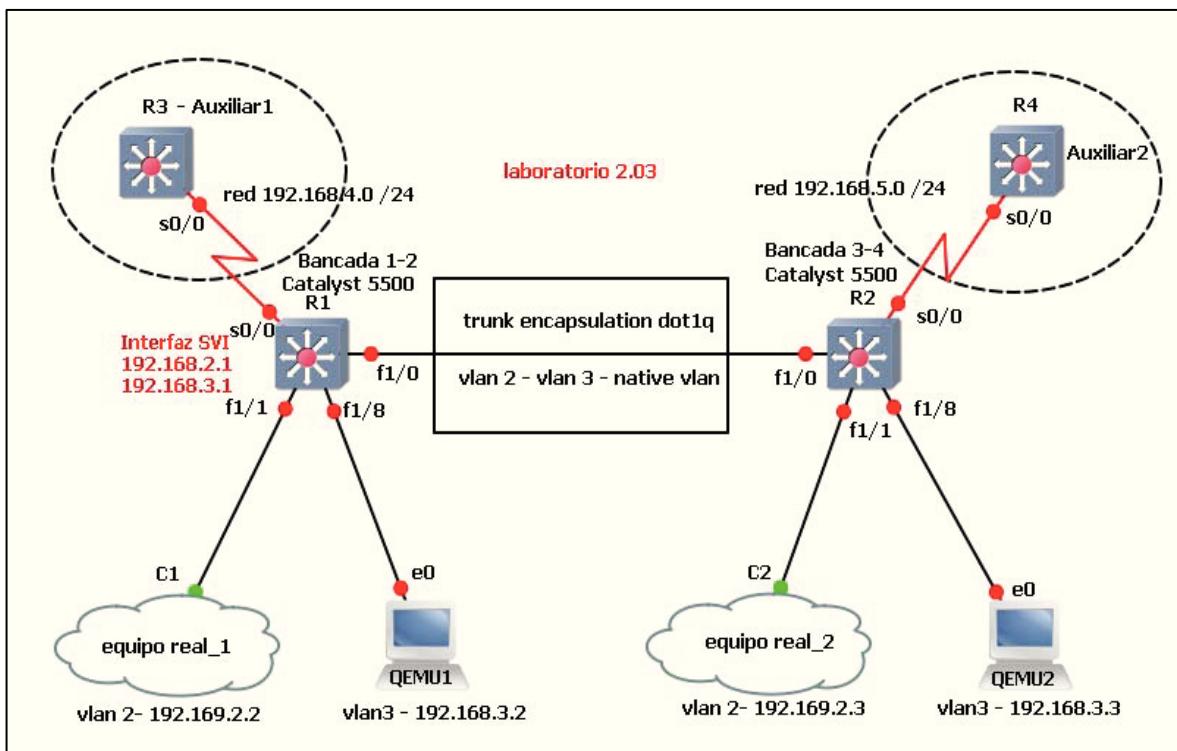


Figura 3.3. Topología/simulación gráfica en GNS3 de la arquitectura LAN Ethernet/VTP.

Se consideran los siguientes aspectos de diseño y configuración sobre el escenario general de la *figura 3.3*, escenario en el que inicialmente sólo consideramos activos para este capítulo los nodos emulados R1 y R2 (las consideraciones sobre los nodos Auxiliar 1 y 2 se hacen en el capítulo 4):

- Entorno de red LAN Ethernet, conmutado a nivel 2 y 3 (gracias a las prestaciones disponibles en dispositivos *switch multilayer* como el *Cisco Catalyst 5500 Switch* del laboratorio que permiten implementar *InterVLAN routing*) [3]. Los dispositivos de red emulados *EtherSwitch router 1* (R1) y *EtherSwitch router 2* (R2) representan y emulan los dispositivos de red *Cisco Catalyst 5500 Switch* del laboratorio.

- En cuanto al interfaz gráfico/simulación de la topología en *GNS3*: a nivel 1, interconexión de los nodos *EtherSwitch router 1* y *2* por medio de un enlace *FastEthernet*, sustituyendo el enlace *serie point-to-point* de la arquitectura WAN ppp anterior.
- Las *bancadas 1-2* y *3-4* del laboratorio 2.03 se interconectan por medio de un enlace *FastEthernet* que actúa como enlace troncal (*trunk link*) por el que circulan, mediante el protocolo 802.1q, VLAN2, VLAN3 y VLAN *nativa*. Sobre esta circunstancia se profundiza más en el anexo 2 (*apartado 7.2.6*).
- Sobre este escenario de trabajo no se configura ningún protocolo a nivel 3, como sí se hacía en la arquitectura de red WAN ppp (recordemos, RIP V2): estamos en un escenario LAN Ethernet conmutado. Solo se considera comunicación a nivel 2 y 3.
- Sobre el escenario general de la *figura 3.3*, primeramente se configuran los nodos *EtherSwitch router 1* y *2* que constituyen la infraestructura principal de esta arquitectura de red, apoyándonos en el *apartado 7.2.6* del anexo 2. Recordemos que el cometido principal en este capítulo es verificar la *usabilidad*, funcionalidad y rigor del entorno emulador. Sobre el resto de la arquitectura nos ocupamos en el capítulo siguiente (nodos Auxiliar 1 y 2).

Añadido a esto, se configura para nuestra arquitectura de red el protocolo VTP (*VLAN Trunk Protocol*), protocolo creado por *Cisco Systems* para resolver los problemas operativos en entornos conmutados/segmentados VLAN [10]. En nuestra arquitectura de red (*figura 3.3*), procedemos a configurar los dispositivos de red emulados *EtherSwitch router 1* y *2* con los siguientes “*roles*” VTP, “*roles*” que se explican brevemente en la *tabla 3.1* de la página siguiente:

- Nodo *EtherSwitch router 1* → Servidor VTP
- Nodo *EtherSwitch router 2* → Cliente VTP

Esta asignación de “*roles*” implica que tomamos el nodo 1 como “cerebro” de la configuración VLAN, sobre el que creamos las mismas y efectuamos todas las modificaciones administrativas futuras. Esta información, gracias a VTP, se transferirá al resto de *switch* participantes del escenario conmutado. En base a estas consideraciones, procedemos con la configuración y verificación del escenario de trabajo en el entorno emulado de *GNS3/Dynamips*, según el anexo 2 de la memoria.

Complementando las disposiciones anteriores, podemos contrastar un análisis más profundo de las características y prestaciones telemáticas del protocolo VTP en el *apartado 7.2.6* del anexo 2.

3.2.2. Captura y análisis de tráfico emulado en el enlace troncal (*trunk link*).

Para realizar un análisis intensivo del entorno software emulador y contrastar su funcionalidad, rigor y correspondencia funcional con entornos de red reales análogos, se realiza una captura del tráfico de red emulado en el enlace troncal de la arquitectura (*trunk link*) configurado con el protocolo 802.1q. Con ello se pretende constatar si por dicho enlace emulado circulan *tagged frames* 802.1q, identificadas según la VLAN a la que corresponden (corroborando así el rigor y la correspondencia funcional del emulador).

58	24.105000	c4:00:07:dc:f1:00	PVST+	68	STP	Conf. Root = 32768/0/c4:00:07:dc:00:02 Cost = 0 Port =
59	24.268000	0.0.0.0	255.255.255.255	331	DHCP	DHCP Discover - Transaction ID 0x532d003d
60	26.059000	c4:00:07:dc:f1:00	ning-tree-(for-bridges	60	STP	Conf. Root = 32768/0/c4:00:07:dc:00:00 Cost = 0 Port =
61	26.071000	c4:00:07:dc:f1:00	PVST+	68	STP	Conf. Root = 32768/0/c4:00:07:dc:00:00 Cost = 0 Port =
62	26.081000	c4:00:07:dc:f1:00	PVST+	68	STP	Conf. Root = 32768/0/c4:00:07:dc:00:01 Cost = 0 Port =
63	26.087000	c4:00:07:dc:f1:00	PVST+	68	STP	Conf. Root = 32768/0/c4:00:07:dc:00:02 Cost = 0 Port =
64	26.506000	CompalIn_78:16:1c	Broadcast	64	ARP	who has 192.168.2.3? Tell 192.168.2.2
65	26.520000	D-LinkIn_cf:cf:8a	CompalIn_78:16:1c	64	ARP	192.168.2.3 is at 84:c9:b2:cf:cf:8a
66	26.520000	192.168.2.2	192.168.2.3	78	ICMP	Echo (ping) request id=0x0001, seq=101/25856, ttl=128
67	26.521000	192.168.2.3	192.168.2.2	78	ICMP	Echo (ping) reply id=0x0001, seq=101/25856, ttl=128
68	27.029000	0.0.0.0	255.255.255.255	331	DHCP	DHCP Discover - Transaction ID 0xaa9c2061
69	27.204000	c4:01:07:dc:f1:00	CDP/FTP/DTP/PAgP/UDLD	341	CDP	Device ID: R2 Port ID: FastEthernet1/0
70	27.500000	192.168.2.2	192.168.2.3	78	ICMP	Echo (ping) request id=0x0001, seq=102/26112, ttl=128
71	27.501000	192.168.2.3	192.168.2.2	78	ICMP	Echo (ping) reply id=0x0001, seq=102/26112, ttl=128

Figura 3.4. Captura de tráfico emulado en enlace troncal (protocolo 802.1q)

Frame 67: 78 bytes on wire (624 bits), 78 bytes captured (624 bits)
Ethernet II, Src: D-LinkIn_cf:cf:8a (84:c9:b2:cf:cf:8a), Dst: CompalIn_78:16:1c (1c:75:08:78:16:1c)
Destination: CompalIn_78:16:1c (1c:75:08:78:16:1c)
Source: D-LinkIn_cf:cf:8a (84:c9:b2:cf:cf:8a)
Address: D-LinkIn_cf:cf:8a (84:c9:b2:cf:cf:8a)
.... .0 = IG bit: Individual address (unicast)
.... .0. = LG bit: Globally unique address (factory default)
Type: 802.1Q Virtual LAN (0x8100)
802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 2
000. = Priority: Best Effort (default) (0)
...0 = CFI: Canonical (0)
.... 0000 0000 0010 = ID: 2

Figura 3.5. Detalle de trama Ethernet 802.1q capturada en el enlace troncal.

Las *figuras 3.4* y *3.5* anteriores nos muestran ciertas características del tráfico capturado y de una trama *Ethernet 802.1q*, respectivamente. Sobre el tráfico capturado, se observan en principio 5 protocolos diferentes en operación en el entorno virtual: STP, DHCP, ARP, CDP e ICMP (comando *ping*). Sin entrar en grandes consideraciones por no ser objetivo del análisis, se apunta teóricamente que STP (*Spanning Tree Protocol*) se usa fundamentalmente en redes comutadas para crear una topología lógica sin loops a partir de una topología física con loops (topología redundante). Las topologías de red redundantes se diseñan para garantizar que las redes continúen funcionando en presencia de puntos únicos de falla, aumentando la confiabilidad del diseño [1] [11].

Los enlaces, puertos y *switch* que no forman parte de la topología activa sin *loops* no envían tramas de datos. El protocolo *Spanning Tree* es una herramienta que permite a un administrador de red disponer de una topología redundante, sin que exista el riesgo de que se produzcan problemas provocados por los *loops* de conmutación.

STP se ejecuta por defecto en dispositivos *Cisco Systems* (*switch* y *switch multilayer* en nuestro caso). Como no se corresponde con el diseño de nuestro supuesto práctico, podría ser desactivado en nuestro escenario de trabajo, mediante el comando *Cisco IOS* siguiente, ejecutado desde el *modo de configuración global* → de interfaz VLAN (ver anexo 1) :

```
switch_router1(config)#interface [vlan/id]
switch_router1(config-if)# no spanning-tree vlan [vlan/id]
```

, esto es, STP debe desactivarse a nivel de *interfaz VLAN* (para todas las VLAN del diseño), no a nivel de interfaz físico *Ethernet*, y debe hacerse en todos los *switch* participantes del diseño comutado, para que no haya errores y no se generen *loops* de conmutación.

Así mismo, un análisis inicial de la trama emulada *Ethernet 802.1q* de la *figura 3.5*, nos muestra el identificador VLAN ID (*tagged frame*, VLAN ID = 2), consecuencia de ejecutar en **192.168.2.3/24** (*equipo_real 2* en nodo *EtherSwitch router 2*, VLAN2): ping **192.168.2.2** (*equipo_real 1* en *EtherSwitch router 1*, VLAN2). Profundizando un poco más para analizar la rigurosidad funcional del emulador, y del consiguiente tráfico de red virtual, consideraremos la estructura teórica de una trama *Ethernet 802.1 q*, descrita en la *figura 3.6*.

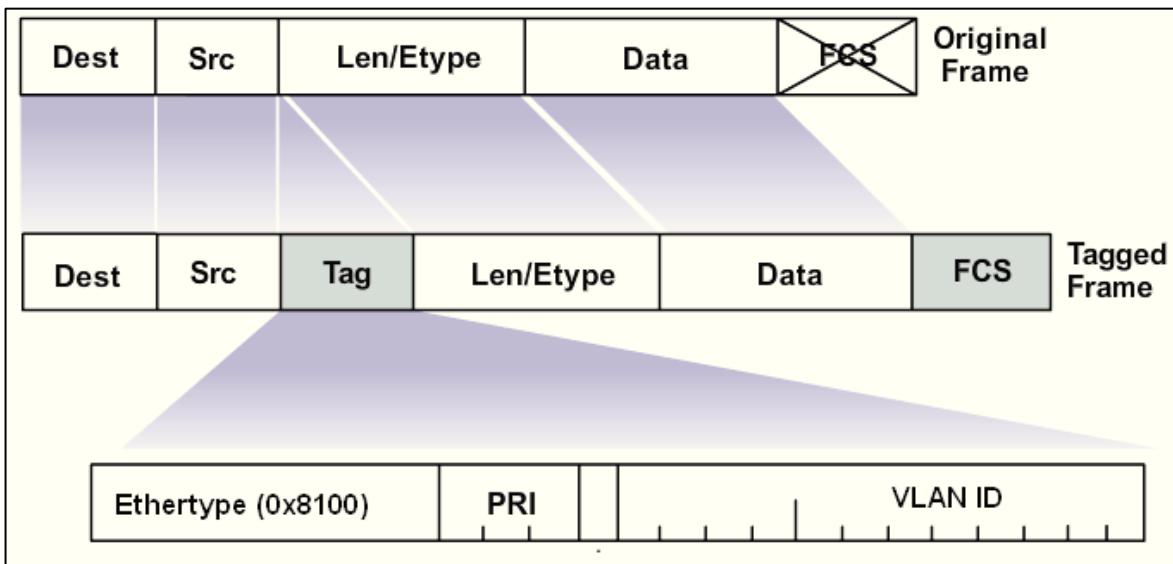


Figura 3.6. Estructura de trama *Ethernet 802.1q*. Imagen extraída de “*Cisco Networking Academy program, CCNP: Building Multilayer Switching Networks v.5.0*” [10].

Vemos como sobre la estructura de una trama *Ethernet* 802.3, el protocolo de enlace troncal añade un campo o cabecera, “*Tag*”, para agregar la siguiente información relativa al protocolo [10] (4 bytes más sobre la trama Ethernet 802.3)

- *Ethertype* (0x8100): 2 bytes; identifica el protocolo troncal 802.1q en uso.
- PRI: 3 bits; transporta información sobre prioridades de la trama.
- *Token Ring Encapsulation Flag*: 1 bit (valor “0” en switch *Ethernet*).
- VLAN ID: 12 bits, para identificar hasta 4096 VLAN.

Al añadir información nueva sobre la trama *Ethernet* 802.3, el campo FCS deberá recalcularse al conformar la trama 802.1q. Un análisis en profundidad de la *tagged frame* de tráfico emulado de figura 3.5 anterior nos muestra en detalle el campo o cabecera “*Tag*” y sus valores en la sucesión de las *subfiguras* 3.7 siguientes.

67 26.521000 192.168.2.3 192.168.2.2 78 ICMP Echo (ping) reply id=0x0001, seq=101/25856, ttl=128

Frame 67: 78 bytes on wire (624 bits), 78 bytes captured (624 bits)
 Ethernet II, Src: D-Linkin_cf:cf:8a (84:c9:b2:cf:cf:8a), Dst: Compalin_78:16:1c (1c:75:08:78:16:1c)
 Destination: Compalin_78:16:1c (1c:75:08:78:16:1c)
 Source: D-Linkin_cf:cf:8a (84:c9:b2:cf:cf:8a)
 Type: 802.1Q Virtual LAN (0x8100)
 802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 2
 000. = Priority: Best Effort (default) (0)
 ...0 = CFI: Canonical (0)
 0000 0000 0010 = ID: 2
 Type: IP (0x0800)
 Internet Protocol Version 4, Src: 192.168.2.3 (192.168.2.3), Dst: 192.168.2.2 (192.168.2.2)
 Internet Control Message Protocol

0000	00011100	01110101	000001000	01111000	00010110	00011100	10000100	11001001	.u.x....
0008	10110010	11001111	11001111	10001010	10000001	00000000	00000000	00000010*
0010	00001000	00000000	010000101	00000000	00000000	00111100	11001100	00101010	..E..<.*
0018	00000000	00000000	100000000	00000000	00000001	11101001	01000000	11000000@..
0020	00000000	00000001	110000000	10101000	00000000	00000000	00000000	00000000
0028	01010100	11110110	00000000	00000000	00000001	00000000	01100101	01100001	T....eab
0030	01100011	01100100	01100101	01100110	01100110	01100100	01101001	01101010	cdefghij
0038	01101011	01101100	01101101	01101110	01101111	01110000	01110001	01110010	klmnopqr
0040	01101011	01101100	01101101	01101110	01101111	01100001	01100010	01100011	stuvwxyz
0048	01100100	01100101	01100110	01100111	01101000	01101001	01101001	01101001	defghi

Figura 3.7 a) Valor del subcampo *Ethertype* (“10000001 00000000” = 0x8100) en la cabecera “*Tag*” de una *tagged frame* 802.1q.

67 26.521000 192.168.2.3 192.168.2.2 78 ICMP Echo (ping) reply id=0x0001, seq=101/25856, ttl=128

Frame 67: 78 bytes on wire (624 bits), 78 bytes captured (624 bits)
 Ethernet II, Src: D-Linkin_cf:cf:8a (84:c9:b2:cf:cf:8a), Dst: Compalin_78:16:1c (1c:75:08:78:16:1c)
 Destination: Compalin_78:16:1c (1c:75:08:78:16:1c)
 Source: D-Linkin_cf:cf:8a (84:c9:b2:cf:cf:8a)
 Type: 802.1Q Virtual LAN (0x8100)
 802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 2
 000. = Priority: Best Effort (default) (0)
 ...0 = CFI: Canonical (0)
 0000 0000 0010 = ID: 2
 Type: IP (0x0800)
 Internet Protocol Version 4, Src: 192.168.2.3 (192.168.2.3), Dst: 192.168.2.2 (192.168.2.2)
 Internet Control Message Protocol

0000	00011100	01110101	000001000	01111000	00010110	00011100	10000100	11001001	.u.x....
0008	10110010	11001111	11001111	10001010	10000001	00000000	00000000	00000010*
0010	00001000	00000000	010000101	00000000	00000000	00111100	11001100	00101010	..E..<.*
0018	00000000	00000000	100000000	00000000	00000001	11101001	01000000	11000000@..
0020	00000000	00000001	110000000	10101000	00000000	00000000	00000000	00000000
0028	01010100	11110110	00000000	00000000	00000001	00000000	01100101	01100001	T....eab
0030	01100011	01100100	01100101	01100110	01100111	01100100	01101001	01101010	cdefghij
0038	01101011	01101100	01101101	01101110	01101111	01110000	01110001	01110010	klmnopqr
0040	01101011	01101100	01101101	01101110	01101111	01100001	01100010	01100011	stuvwxyz
0048	01100100	01100101	01100110	01100111	01101000	01101001	01101001	01101001	defghi

Figura 3.7 b). Valor del subcampo *PRI* (“000 ”) en la cabecera “*Tag*” de una *tagged frame* 802.1q

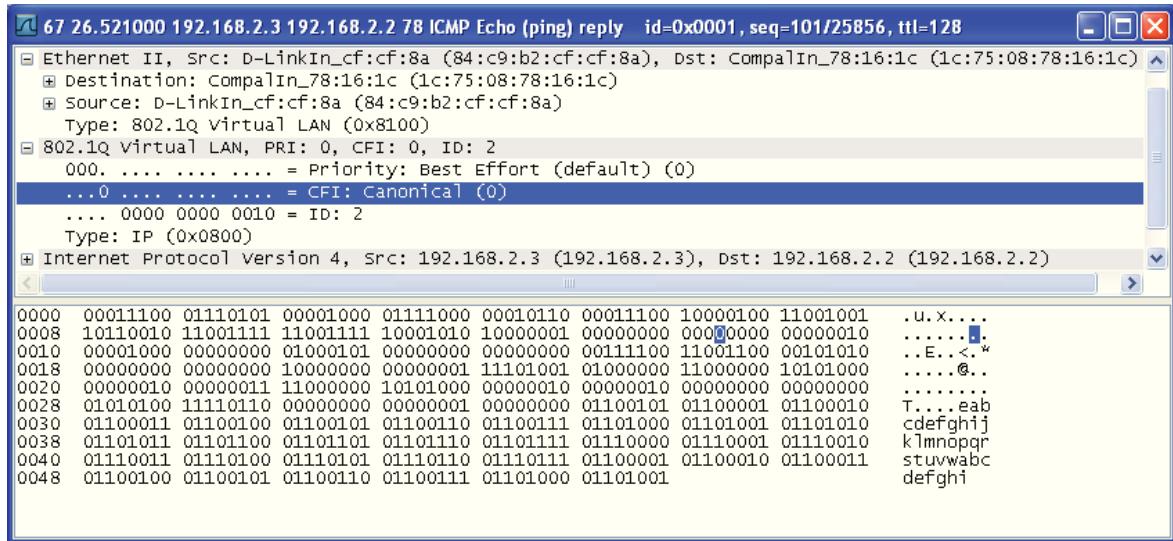


Figura 3.7 c). Valor del subcampo Token Ring Encapsulation Flag (“0”) en la cabecera “Tag” de una tagged frame 802.1q .

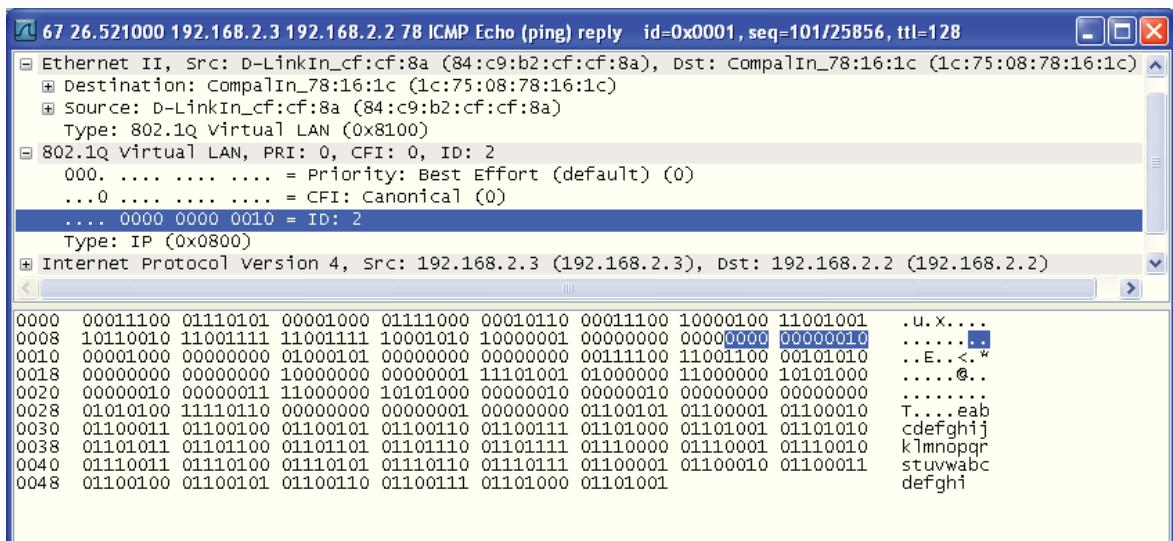


Figura 3.7 d). Valor de subcampo VLAN ID (“0000 00000010” = VLAN2) en la cabecera “Tag” de una tagged frame 802.1q.

La sucesión de subfiguras 3.7 anteriores permite comprobar cómo la estructura y valores de la cabecera “Tag” en tagged frames 802.1q de tráfico virtual se corresponde con la estructura teórica de tramas de tráfico real, corroborando el rigor y la correspondencia funcional entre los entornos de red real y emulado. Se aprecia igualmente cómo el emulador realiza un tratamiento trama a trama de la información, ajustándose con mayor o menor exactitud a un tratamiento en tiempo real de la misma.

Capítulo 4: Análisis del rendimiento y capacidad del sistema emulador. Relación con la latencia de red en la interoperación entorno real \rightarrow entorno emulado.

4.1. Contexto del análisis.

Siendo el objetivo principal de este capítulo el análisis de la capacidad y rendimiento del sistema emulador, así como su posible influencia en la latencia de red, el contexto del análisis actual y su relación con resultados anteriores se concreta esquemáticamente en la figura 4.0.

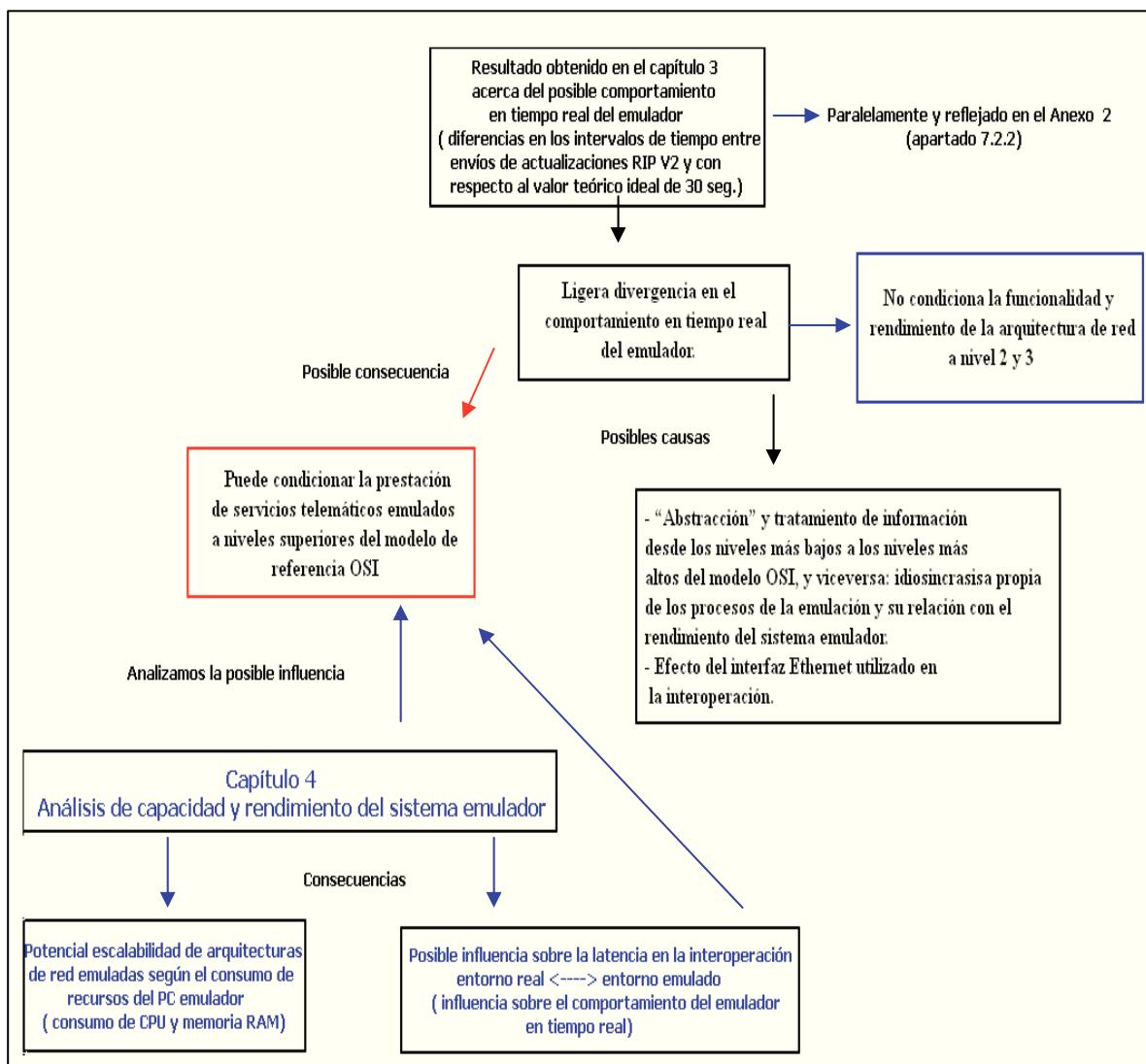


Figura 4.0. Contexto esquemático para el análisis del capítulo 4.

4.2. Análisis bajo sistema operativo Windows.

Siendo el sistema operativo motor de la emulación Windows XP de 32 bits, se analizan el rendimiento/capacidad hardware y parámetros de red (efectos sobre la latencia) para las arquitecturas de red planteadas en el capítulo 3. Como anexo 4 del proyecto, complementando los análisis en Windows, se analizan las prestaciones y rendimiento del sistema emulador en entornos de trabajo Linux (distribución Ubuntu 12.04). Con ello no se quieren obviar las prestaciones de *GNS3/Dynamips* en sistemas operativos Linux.

4.2.1. Análisis del rendimiento del sistema emulador para la arquitectura de red WAN ppp.

Se establecen las siguientes configuraciones hardware *base* para el sistema emulador (PC emulador), reflejadas en la *tabla 4.1*:

PC emulador	Procesador	Memoria RAM	Memoria Cache del Procesador
Configuración hardware 1	Intel Pentium 4 2.8 Ghz	1 GB DDR 400 Mhz	L1: 12 + 8 KB L2: 512 KB
Configuración hardware 2	Intel Pentium 4 3.2 Ghz Prescott	2 GB DDR2 667 Mhz	L1: 16 KB L2: 1 MB

Tabla 4.1. Configuraciones hardware *base* para el sistema emulador.

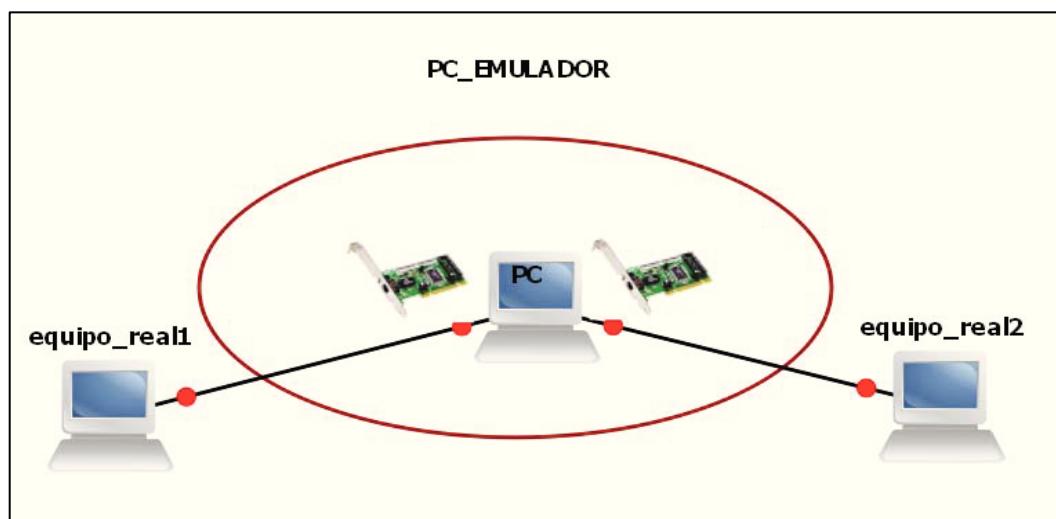


Figura 4.1. Esquema físico para la conexión/integración de hardware y tráfico real en el entorno de emulación *GNS3/Dynamips*.

La *configuración hardware 1* incorpora el hardware y tráfico de red real en el entorno emulado a través de un equipo (portátil) externo (*equipo_real 1*), y un interfaz de red USB *FastEthernet* propio en el PC emulador (*equipo_real 2*). El PC emulador dispone, por lo tanto, de 3 interfaces de red *FastEthernet*: 2 sirven para *integrar/interoperar* en el entorno emulado sendos equipos reales, y el otro para *autoejercer* el rol de *equipo_real 2* (interfaz de red USB *FastEthernet*). Esto se detalla más intensamente en el anexo 3, apartado 7.3.3.

Para conformar el análisis de la capacidad y rendimiento del PC emulador, y de cara a establecer una relación más significativa entre el número de nodos activos participantes en el escenario emulado de trabajo (*router-switch multilayer*) y el rendimiento/consumo de recursos del sistema emulador, sobre la arquitectura de red principal de la figura 3.1 del capítulo 3 se añaden 2 enlaces *serie ppp* que interconectan los nodos *EtherSwitch router 1* y *EtherSwitch router 2* con sendos nodos emulados *EtherSwitch router*, que denominamos nodo Auxiliar 1 y 2, sin ningún propósito especial de diseño telemático. El único objetivo es evaluar la repercusión directa sobre el rendimiento del sistema emulador según el número de nodos emulados activos en un escenario (para evaluar la potencial escalabilidad del diseño de la arquitectura). Estas configuraciones adicionales se detallan en el apartado 7.2.7 del anexo 2.

Siendo así, se obtienen los resultados de la *tabla 4.2*, directamente relacionados con el rendimiento y prestaciones del sistema emulador. Los mismos son estables tras un pequeño transitorio inicial una vez “*iniciamos*” los equipos participantes en el escenario emulado: es el tiempo necesario para la carga del sistema operativo *Cisco IOS* y las configuraciones asociadas a la arquitectura de red (guardadas en la memoria virtual de los nodos *EtherSwitch router*), y del sistema Linux *MicroCore 3.8.2*, sistema operativo emulado en los terminales virtuales Linux QEMU, si hacemos uso de dichos terminales en el escenario emulado. Sobre el uso y conveniencia de dichos terminales se profundiza más en el apartado 7.2.5 del anexo 2.

Configuración hardware 1	Carga de CPU (offmode) ¹	Memoria RAM en uso en MB (offmode)	Carga media de CPU (pseudoactivo) ²	Memoria RAM en uso en MB (pseudoactivo)
1 switch-router activo	1% max.	318	1-2 %	780 → 540
2 switch-router activos	1% max.	318	4 % → 2-3 %	940 → 720
2 switch-router activos + 1 QEMU	1% max.	318	16 % → 14 %	950 → 750
2 switch-router activos + 2 QEMU	1% max.	318	17 % → 15 %	960 → 760
2 switch-router activos + 1 auxiliar (sin QEMU)	1% max.	318	6 % → 4-5 %	970 → 760
2 switch-router activos + 2 auxiliar (sin QEMU)	1% max.	318	11% → 9-10 %	980 → 790

Tabla 4.2. Consumo de recursos del sistema emulador para la configuración hardware 1, bajo S.O. Windows XP 32 bits y arquitectura de red WAN ppp.

En la tabla 4.2 anterior, tabla 4.3 y siguientes, las expresiones **XX → XX** deben interpretarse como un descenso en el consumo de recursos en el PC emulador. A continuación se analiza detalladamente esta circunstancia.

Antes, se destaca que para el análisis de la configuración hardware 2, equipo_real 1 y equipo_real 2 son sendos *equipos reales externos* (equipos portátiles), integrados en el escenario emulado de trabajo vía interfaces de red Ethernet específicos en el PC emulador, y siguiendo las instrucciones y requerimientos descritos en el anexo 3. En base a esta configuración hardware, la tabla 4.3 siguiente muestra los resultados obtenidos en el análisis de la capacidad y rendimiento del sistema emulador.

¹ Estado inicial en el que ningún nodo y/o terminal QEMU está en funcionamiento.

² Estado en el que el tráfico de red presente es el relacionado con el mantenimiento y actualización de los protocolos configurados en el escenario de trabajo (información dinámica de nivel 2 y 3 → escaso tráfico)

Configuración hardware 2	Carga del procesador (<i>off mode</i>)	Memoria RAM en uso en MB (<i>off mode</i>)	Carga media del procesador (pseudoactivo)	Memoria RAM en uso en MB (pseudoactivo)
1 <i>switch-router</i> activo	1% max	330	1-2%	770 → 512
2 <i>switch-router</i> activos	1% max	330	3 % → 2 %	940 → 710
2 <i>switch-router</i> activos + 1 <i>QEMU</i>	1% max	330	15 % → 12%	970 → 770
2 <i>switch-router</i> activos + 2 <i>QEMU</i>	1% max	330	16 % → 14 %	980 → 780
2 <i>switch-router</i> activos + 1 auxiliar (sin <i>QEMU</i>)	1% max	330	5 % → 4 %	980 → 760
2 <i>switch-router</i> activos + 2 auxiliar (sin <i>QEMU</i>)	1% max	330	9% → 8 %	990 → 800

Tabla 4.3. Consumo de recursos del sistema emulador para la configuración hardware 2, bajo S.O. Windows XP 32 bits y arquitectura de red WAN ppp.

Se constata empíricamente que los valores de carga de CPU y consumo de memoria RAM son muy fluctuantes en todos los experimentos realizados, ponderándose la media de los mismos. El descenso en el consumo de recursos anterior se asocia principalmente al uso de los recursos *optimizadores* de memoria RAM disponibles en el entorno emulador, y detallados más intensamente en el *apartado 7.3.6* del anexo 3. Se contrasta empíricamente que estos recursos son manifiestamente ejecutados por el sistema transcurridos varios minutos desde la puesta en funcionamiento de la arquitectura emulada, y la optimización máxima se consigue en topologías en las que los dispositivos emulados se corresponden con un único modelo, es decir, sólo se está emulando un sistema o imagen Cisco IOS, a pesar de existir varios nodos iguales en la arquitectura de red.

La carga completa y funcionalmente estable del sistema *Cisco IOS* en el PC emulador, considerando además que se ha configurado previamente un correcto y decisivo valor del parámetro *idle PC* (parámetro de configuración inicial del emulador y recurso optimizador prioritario de CPU detallado en el *apartado 7.3.5* del anexo 3), se consigue transcurridos pocos minutos, y se refleja secundariamente en un ligero descenso de la carga de CPU (ojo, éste es un descenso de la carga de CPU pequeño y “secundario”, no tiene que ver con el asociado al anterior parámetro *idle PC*). Establecer posibles dependencias entre los descensos en el consumo de memoria RAM y CPU desembocaría en un análisis *hardware* a más bajo nivel que escapa a los propósitos del proyecto. Por ello, enunciamos de un modo general la característica de *estabilidad del sistema* alcanzada tras varios minutos de

actividad (se comprueba empíricamente que este tiempo es de 5-10 minutos máximo) y haciendo uso de los recursos optimizadores aportados por *GNS3/Dynamips* descritos en el anexo 3. Sobre los consumos de recursos asociados a los terminales QEMU se especifican y detallan ciertas connotaciones secundarias en el *apartado 7.2.5* del anexo 2.

A la hora de ejecutar los procesos asociados con la emulación, *GNS3/Dynamips* consume como mínimo, y en principio (sin el uso de los recursos *optimizadores* de memoria descritos en el anexo 3), la misma cantidad de memoria RAM real que consumirían el conjunto de todos los dispositivos reales que se emulan en el escenario de trabajo. Consecuentemente, el consumo de memoria RAM es ciertamente crítico a la hora de ejecutar la emulación. Sin el uso de los recursos *optimizadores* de memoria, la escalabilidad de la topología asociada a una determinada arquitectura sería muy limitada. Estas vicisitudes se han comprobado empíricamente en el desarrollo del proyecto, resultando inviable la emulación en caso de no optar por la optimización de memoria RAM en el supuesto de la *configuración hardware 1* (1 GB de RAM), ya que en nuestro caso los dispositivos de red emulados *EtherSwitch router* emulan/disponen de 128 Mb de RAM “virtual” cada uno).

Activada la función de optimización *ghost ios* [13] descrita en el *apartado 7.3.6* del anexo 3, la cantidad de memoria RAM ahorrada no es un proceso exacto ni matemáticamente accesible al cálculo debido a la complejidad en la administración de la memoria RAM por parte de los sistemas operativos. Simplemente, se procede a mostrar empíricamente el consumo de memoria RAM en las *tablas 4.2* y *4.3*. Con la facilidad *ghostios* activa, el consumo de memoria RAM deja de ser una circunstancia crítica a la hora de abordar la emulación para el escenario que nos ocupa, en el que todos los dispositivos emulados ejecutan el mismo sistema *Cisco IOS*. La única recomendación práctica al respecto es simple: que nuestro sistema emulador disponga de la mayor cantidad de memoria RAM posible (mínimo recomendable 2 GB). Derivado de todo lo anterior, la escalabilidad de los escenarios emulados de trabajo (en principio, y como aproximación, se interpreta este parámetro como el número de nodos que el sistema es capaz de incorporar adecuadamente al diseño y emular en una arquitectura) se ve limitada principalmente por el consumo de CPU por parte del sistema emulador.

El consumo de CPU que reflejan las *tablas 4.2* y *4.3*, mostrado en las *figuras 4.2* y *4.3*, constata una evolución según el número de nodos activos que aproximamos a una progresión geométrica con el objetivo de establecer una cota superior de escalabilidad suficientemente fiable, y así establecer un margen adecuado de trabajo para nuestro escenario telemático. Igualmente, se aprecia cómo este consumo depende clara y directamente del procesador que monta nuestro sistema emulador. Así, en el caso de la *configuración hardware 1*, en base a los resultados de la *tabla 4.2* y para una *situación estable* del escenario de trabajo caracterizada por el descenso en el consumo de recursos, podríamos emular arquitecturas con hasta 7 nodos de red activos conforme al margen de trabajo. Esta cota superior conduce a un consumo de CPU entre el 70-90 %.

La cota superior del margen *escalable* de trabajo lo establecemos considerando escenarios en los que implementamos una *interoperación* entre entornos de red real y emulado, escenarios en los que subyace un cierto interés *teórico-práctico* en analizar el comportamiento del tráfico de red. En caso de implementar escenarios emulados con objetivos telemáticos más básicos, como por ejemplo el estudio del comportamiento de protocolos de comunicaciones a nivel 2 o nivel 3, la escalabilidad del escenario de trabajo puede ser ligeramente mayor, pudiendo llegar al límite del 100 % de carga de CPU durante un cierto periodo de tiempo (esto se contrasta empíricamente). Esta situación conduce a un comportamiento del sistema emulador inadecuado y limitado, pero funcionalmente capaz de acometer los procesos de emulación para propósitos telemáticos básicos de estudio a nivel 2 y 3 (estando el número de nodos a emular, evidentemente, dentro de unos límites).

Para la *configuración hardware 2*, y según los resultados mostrados en la *tabla 4.3*, el consumo de CPU desciende ligeramente con respecto al caso anterior, evidenciando una dependencia con el *hardware* del PC emulador. Esto nos permitiría, y en caso estrictamente necesario y límite, llegar a una cota superior de escalabilidad de hasta 8 nodos emulados en nuestra arquitectura, eso sí, con un incremento en el consumo de CPU con respecto a la *configuración hardware 1* anterior. Al respecto, improvisar una fórmula matemática general que relacione el número de nodos activos con el consumo de CPU resulta inviable debido a la dependencia directa que existe con el *hardware* del sistema emulador.

En el análisis de estos resultados es preciso tomar en consideración el hecho de que los PC's que ejecutan la emulación se corresponden con equipos de gama media-baja en el contexto tecnológico actual.

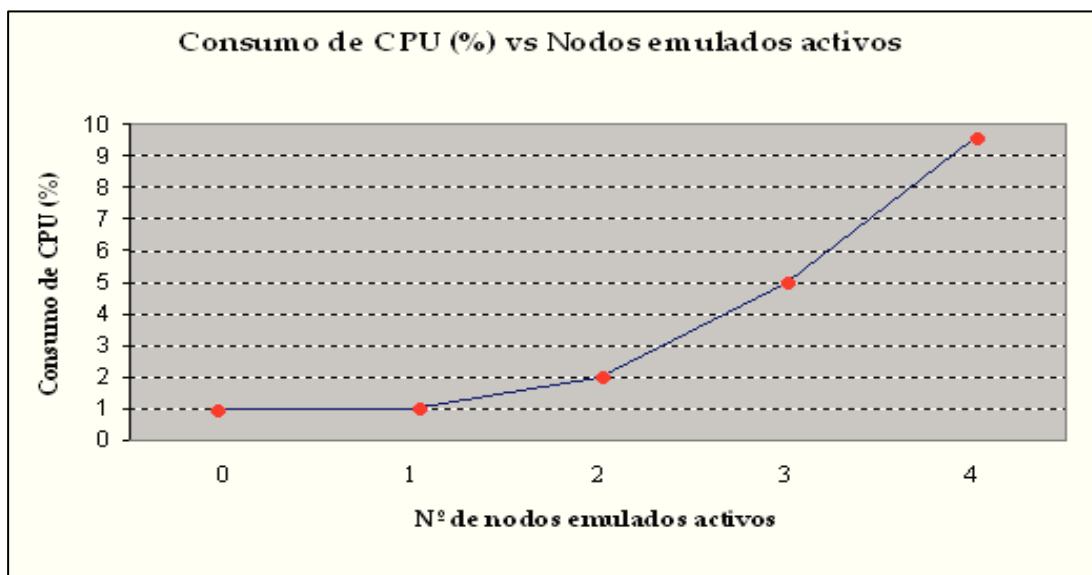


Figura 4.2. Consumo de CPU en el sistema emulador (PC) en función del número de nodos emulados activos para la arquitectura WAN ppp y *configuración hardware 1*.

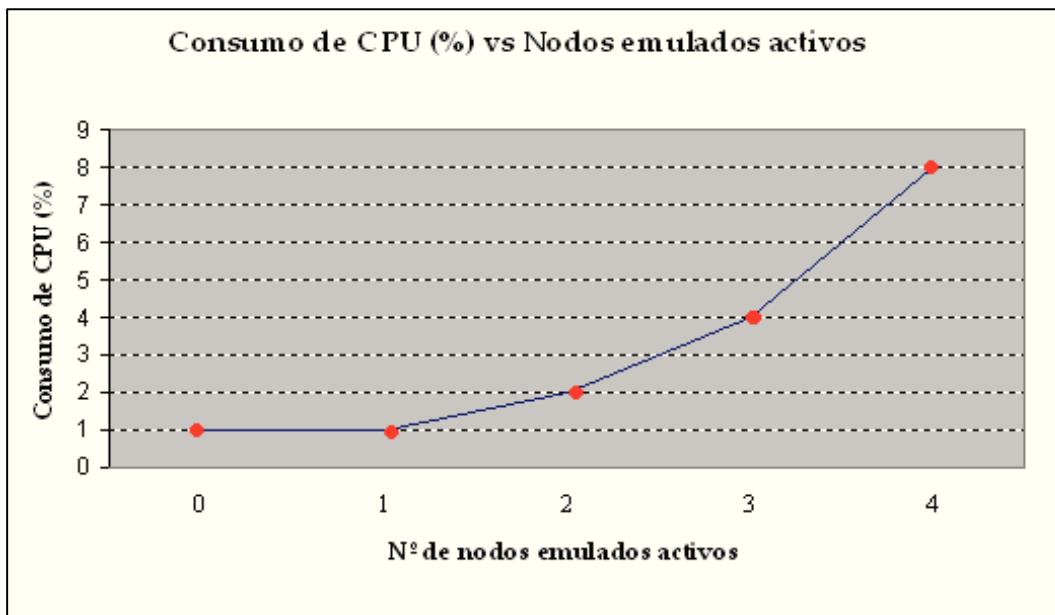


Figura 4.3. Consumo de CPU en el sistema emulador (PC) en función del número de nodos emulados activos para la arquitectura WAN ppp y *configuración hardware 2*.

4.2.2. Análisis de la latencia para la arquitectura de red WAN ppp.

Seguidamente, se analiza en la *interoperación entorno real \leftrightarrow entorno emulado* de GNS3/Dynamips el efecto sobre la latencia del tráfico de red en transmisiones sin control de errores. Los experimentos a realizar son dos:

- Latencia para diferentes longitudes de trama *Ethernet*, en transmisiones sin control de errores (protocolo ICMP) para la conexión: *equipo real 1 \leftrightarrow interfaz SVI de nodo virtual emulado 1*.
- Latencia para diferentes longitudes de trama *Ethernet*, en transmisiones sin control de errores (protocolo ICMP), para la conexión *extremo a extremo*: *equipo real 1 \leftrightarrow entorno emulado (vía SVI de nodo virtual emulado 1) \leftrightarrow equipo real 2*.

Los análisis efectuados son evaluados para las diferentes cargas de CPU y consumos de memoria RAM para las configuraciones hardware de las *tablas 4.2 y 4.3*, tablas obtenidas anteriormente y de modo empírico en el análisis de la capacidad y rendimiento del sistema emulador. Los resultados obtenidos en los análisis siguientes se relacionan con instantes de tiempo de consumo máximo de recursos del sistema emulador, según reflejan las anteriores *tablas 4.2 y 4.3*. Esto es, suponen una cota superior de los resultados.

Latencia en transmisiones sin control de errores (protocolo ICMP) para la conexión equipo real 1 (Ethernet 802.3) \leftrightarrow interfaz SVI de nodo virtual emulado 1.

Atendiendo al escenario de la figura 3.1 y haciendo uso del protocolo ICMP, protocolo que no realiza control de errores y solo informa de los mismos, desde *equipo_real_1* (**192.168.2.2**) implementamos solicitudes de eco con destino el interfaz virtual SVI de VLAN2 en el nodo virtual/emulado *EtherSwitch router 1* (**192.168.2.1**):

```
ping 192.168.2.1 -n cuenta -l longitud
```

, donde **-n cuenta** es el número de solicitudes de eco enviadas (en nuestro caso serán 200) y **-l longitud** es el tamaño en *bytes* del campo *datos* del *datagrama* ICMP a tal efecto (sin considerar el tamaño de la cabecera). Se realizan 20 repeticiones del experimento, ponderándose la media de los mismos. Las longitudes de trama *Ethernet* 802.3 consideradas para este análisis serán la longitud de trama mínima (64 *bytes*, que se corresponde con un valor de MTU mínimo de 46 *bytes*), la longitud de trama 1514 *bytes* (que corresponde a un valor de MTU máximo de 1500 *bytes*) y dos longitudes intermedias, 512 *bytes* y 1024 *bytes*. Los Δ entre las longitudes de trama consideradas son aproximadamente los mismos. Interpretando teóricamente la estructura de la trama *Ethernet* 802.3 y los campos/cabeceras que conforman el *datagrama* IP/ICMP, se calcula el valor del parámetro **-l longitud**, en la ejecución del comando *ping* anterior:

$$\text{Longitud (datagrama IP/ICMP)} = (\text{longitud de trama Ethernet} - 14 \text{ bytes de cabecera Ethernet}) - (20 \text{ bytes de cabecera IP}) - (8 \text{ bytes de cabecera ICMP})$$

La longitud de trama máxima empleada en el experimento, 1514 *bytes*, es la que nos permite utilizar el máximo valor de MTU (1500 *bytes*) sin que haya fragmentación de la información (*Fragmented IP protocol*).

En las *tablas 4.4* y *4.5* de la página siguiente, que recogen los resultados obtenidos para este experimento (valor del *Round Trip Time* promedio) se interpretan los *switch-router* activos (valor nominal de las columnas) por el consumo directo de recursos que implican en el PC emulador, consumos reflejados en las *tablas 4.2* y *4.3* anteriores.

En el desarrollo de este experimento no se ha producido ningún *TIMEOUT*, esto es, la pérdida de ningún *datagrama* ICMP.

RTT promedio <i>(configuración hardware 1)</i>	1 switch-router activo	2 switch-router activos	3 switch-router activos	4 switch-router activos
Longitud Trama 64 bytes	31 ms	40 ms	46 ms	61 ms
Longitud Trama 512 bytes	33 ms	40 ms	47 ms	62 ms
Longitud Trama 1024 bytes	35 ms	42 ms	49 ms	66 ms
Longitud Trama 1514 bytes	38 ms	43 ms	52 ms	67 ms

Tabla 4.4. Latencia de red para el protocolo ICMP en la conexión *equipo_real 1* → *interfaz SVI en nodo virtual 1*, para diferentes longitudes de trama y consumo de recursos del PC emulador 1 en la arquitectura WAN ppp.

RTT promedio <i>(configuración hardware 2)</i>	1 switch-router activo	2 switch-router activos	3 switch-router activos	4 switch-router activos
Longitud Trama 64 bytes	30 ms	40 ms	45 ms	59 ms
Longitud Trama 512 bytes	33 ms	40 ms	47ms	61 ms
Longitud Trama 1024 bytes	34 ms	41 ms	48 ms	65 ms
Longitud Trama 1514 bytes	37 ms	42 ms	50 ms	66 ms

Tabla 4.5. Latencia de red para el protocolo ICMP en la conexión *equipo_real 1* → *interfaz SVI en nodo virtual 1*, para diferentes longitudes de trama y consumo de recursos del PC emulador 2 en la arquitectura WAN ppp.

Los resultados de las tablas anteriores muestran unos valores muy elevados de latencia en la *interoperación entorno real (Ethernet 802.3) ↔ entorno emulado (interfaz SVI de nodo virtual emulado 1)* para esta arquitectura de red (comunicación a nivel 3), en comparación con la latencia de red real obtenida para este mismo experimento en el laboratorio (RTT máximo = 1ms). En base a los resultados anteriores cabe preguntarse qué valor o *ratio* de la latencia en la *interoperación* es imputable al interfaz físico *Ethernet* empleado en la *interoperación* y qué valor o *ratio* en el total de la misma es imputable al proceso de emulación en sí.

El valor imputable al proceso de emulación en la *interoperación* podemos analizarlo capturando el tráfico de red en el enlace de la interoperación en cuestión, mediante *Wireshark*, captura en la que profundizan las *figuras 7.20 a) y 7.20 f)* del anexo 2, y calculando la diferencia de tiempo entre *Request* y *Reply* de cada petición de eco,

ponderando la media en base a todas las peticiones. El trabajo para ponderar la media de las (200*20) repeticiones para todos los supuestos prácticos sería ingente y desproporcionado, por eso se ha realizado sólo en base a 100 peticiones de eco de un solo experimento (el experimento relacionado con un solo *switch-router* activo para la *configuración hardware 1*). Así, éste es un valor aproximado al buscado (reflejado en la *tabla 4.6*), pero indicativo e ilustrativo general de la incidencia/comportamiento que es objeto del análisis.

Δ Request – Reply <i>interoperación real \leftrightarrow emulado</i> <i>(Valor/ratio de latencia imputable al proceso de emulación)</i>	1 <i>switch-router</i> activo <i>(EtherSwitch router 1)</i>
Longitud Trama 64 bytes	30 ms (97 % de la latencia en la interoperación)
Longitud Trama 512 bytes	31 ms (94 % de la latencia en la interoperación)
Longitud Trama 1024 bytes	32 ms (91 % de la latencia en la interoperación)
Longitud Trama 1514 bytes	35 ms (94 % de la latencia en la interoperación)

Tabla 4.6). Latencia de red imputable a los procesos de emulación con respecto a la latencia total en el enlace *equipo_real 1 (Ethernet 802.3) \rightarrow interfaz SVI en nodo virtual 1*, para la arquitectura WAN ppp, protocolo ICMP, diferentes longitudes de trama y 1 *switch-router* activo.

La *tabla 4.6* anterior constata empíricamente en el valor de la latencia en la *interoperación entorno real \leftrightarrow entorno emulado (interfaz SVI de nodo virtual emulado 1)* un mayor peso específico del proceso de emulación que del interfaz *Ethernet* a través del que se realiza la *interoperación*.

Latencia en transmisiones sin control de errores (protocolo ICMP) en la conexión extremo a extremo *equipo real 1* ↔ *entorno emulado* ↔ *equipo real 2*.

Se realiza un experimento formalmente similar al anterior para evaluar la latencia de la arquitectura de red *extremo a extremo*, esto es, *equipo_real 1 (192.168.2.2)* ↔ *entorno virtual* ↔ *equipo_real 2 (192.168.4.2)*. Según la configuración de red establecida para el escenario de trabajo de la figura 3.1, el formato del comando ICMP a ejecutar en este caso sería:

ping 192.168.4.2 -n cuenta -1 longitud
--

RTT promedio (configuración hardware 1)	1 switch-router activo	2 switch-router activos	3 switch-router activos	4 switch-router activos
Longitud Trama 64 bytes	-	56 ms	57 ms	62 ms
Longitud Trama 512 bytes	-	57 ms	59 ms	63 ms
Longitud Trama 1024 bytes	-	90 ms	91 ms	92 ms
Longitud Trama 1514 bytes	-	119 ms	121 ms	124 ms

Tabla 4.7. Latencia de red para el protocolo ICMP en la conexión *equipo_real 1* → *red virtual* → *equipo_real 2*, para diferentes longitudes de trama y consumo de recursos del PC emulador 1 en la arquitectura WAN ppp.

RTT promedio (configuración hardware 2)	1 switch-router activo	2 switch-router activos	3 switch-router activos	4 switch-router activos
Longitud Trama 64 bytes	-	55 ms	56 ms	60 ms
Longitud Trama 512 bytes	-	56 ms	58 ms	62 ms
Longitud Trama 1024 bytes	-	89 ms	90 ms	91 ms
Longitud Trama 1514 bytes	-	118 ms	119 ms	122 ms

Tabla 4.8. Latencia de red para el protocolo ICMP en la conexión *equipo real 1* → *red virtual* → *equipo real 2*, para diferentes longitudes de trama y consumo de recursos del PC emulador 2 en la arquitectura WAN ppp.

Al igual que anteriormente, se han realizado 20 repeticiones del experimento, ponderándose la media de los mismos. Los resultados mostrados en las *tablas 4.7 y 4.8* anteriores constatan empíricamente un valor igualmente elevado para la latencia en este experimento y para la arquitectura de red actual, comparado con un valor teórico máximo esperable máximo de 2- 3 ms en este caso concreto. Los valores de la latencia en todos los experimentos realizados son bastante fluctuantes, pudiendo ser atribuibles, principalmente, a las fluctuaciones existentes en el consumo de recursos del PC emulador previamente constatadas.

En el desarrollo de este experimento no se ha producido ningún *TIMEOUT*, esto es, la pérdida de ningún *datagrama ICMP*.

4.2.3. Análisis del rendimiento del sistema emulador para la arquitectura de red LAN Ethernet/VTP.

Se realizan idénticos experimentos a los efectuados anteriormente en base a las *configuraciones hardware 1 y 2* de la *tabla 4.1*. Para ello, recordemos que se incorporan para el propósito de los análisis de este capítulo 4, sobre el diseño de la arquitectura de red principal de la *figura 3.3* dos nodos, Auxiliar 1 y 2, no participantes del escenario LAN Ethernet/VTP conmutado. El propósito es evaluar la capacidad y rendimiento del sistema emulador y la potencial escalabilidad de la arquitectura de red actual atendiendo al número de nodos de red emulados activos, es decir, al consumo de recursos del PC emulador.

Sobre la configuración de estos nodos adyacentes *EtherSwitch router* Auxiliar 1 y 2 (R3 y R4) se profundiza en el *apartado 7.2.7* del mismo anexo 2. Así, sin ningún propósito especial de diseño telemático, con el único objetivo de evaluar el rendimiento del sistema emulador según el número de nodos activos en el escenario (y evaluar así la potencial escalabilidad de la arquitectura) se configuran en dicho anexo:

Red **192.168.4.0 /24** (enlace serie): Nodo *EtherSwitch router1* → Auxiliar 1

Red **192.168.5.0/ 24** (enlace serie): Nodo *EtherSwitch router2* → Auxiliar 2

A continuación, se muestran en las *tablas 4.9 y 4.10* los resultados obtenidos en los análisis de la capacidad y del rendimiento del sistema emulador, tal como se efectuó en la arquitectura anterior.

Configuración hardware 1	Carga media de CPU (<i>offmode</i>)	Memoria RAM en uso en MB (<i>offmode</i>)	Carga media de CPU (<i>pseudoactivo</i>)	Memoria RAM en uso en MB (<i>pseudoactivo</i>)
1 <i>switch-router</i> activo	1% max.	318	2 %	840 → 570
2 <i>switch-router</i> activos	1% max.	318	5% → 4 %	960 → 740
2 <i>switch-router</i> activos + 1 nodo auxiliar	1% max.	318	7 % → 6 %	980 → 770
2 <i>switch-router</i> activos + 2 nodos auxiliar	1% max.	318	13 % → 11 %	990 → 780

Tabla 4.9. Consumo de recursos del sistema emulador para la *configuración hardware 1*, bajo S.O. Windows XP 32 bits y arquitectura de red LAN/VTP.

Configuración hardware 2	Carga media de CPU (offmode)	Memoria RAM en uso en MB (off mode)	Carga media de CPU (pseudoactivo)	Memoria RAM en uso en MB (pseudoactivo)
1 switch-router activo	1% max.	330	2 %	850 → 580
2 switch-router activos	1% max.	330	4% → 3 %	960 → 730
2 switch-router activos + 1 nodo auxiliar	1% max.	330	6% → 4%	980 → 770
2 switch-router activos + 2 nodos auxiliar	1% max.	330	10 % → 9 %	1000 → 780

Tabla 4.10. Consumo de recursos del sistema emulador para la *configuración hardware 2*, bajo S.O. Windows XP 32 bits y arquitectura de red LAN/VTP.

Válidas todas las connotaciones anteriores en el análisis de la arquitectura WAN ppp, lo primero que evidencian los resultados de las *tablas 4.9 y 4.10* es una ligera dependencia con respecto a la tecnología implementada en la arquitectura de red, aumentando en el caso que nos ocupa el consumo de CPU (muy ligeramente). Esto puede guardar relación directa con el hecho de tener que emular tráfico de red relacionado con el protocolo VTP configurado en el escenario. Independientemente, esto no afecta al número de nodos capaz de emular nuestro sistema con respecto a la arquitectura anterior, siendo equiparables a los resultados anteriores. Únicamente se hace constar la ligera diferencia empírica.

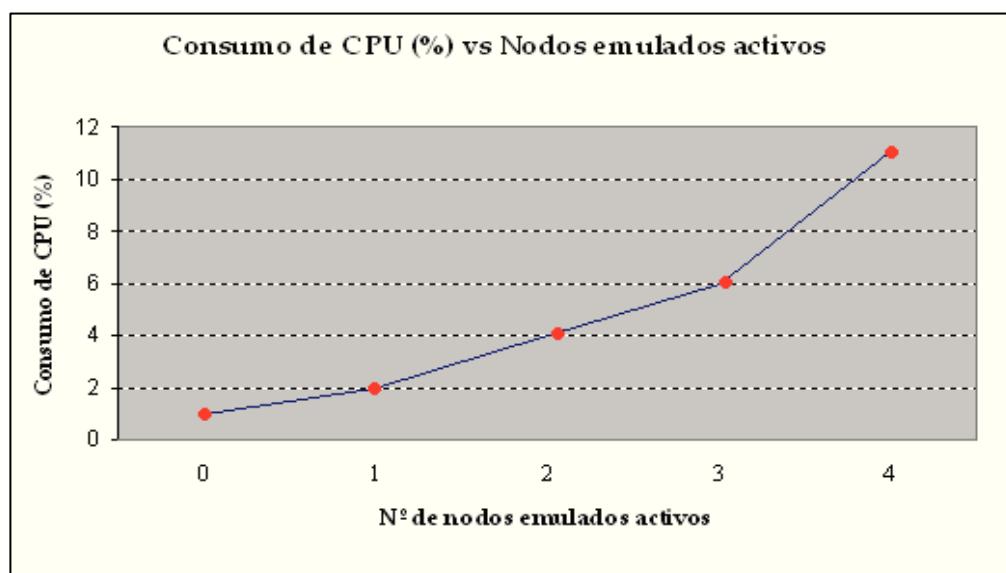


Figura 4.4. Consumo de CPU en el sistema emulador (PC) en función del número de nodos emulados activos para la arquitectura LAN/VTP y *configuración hardware 1*.

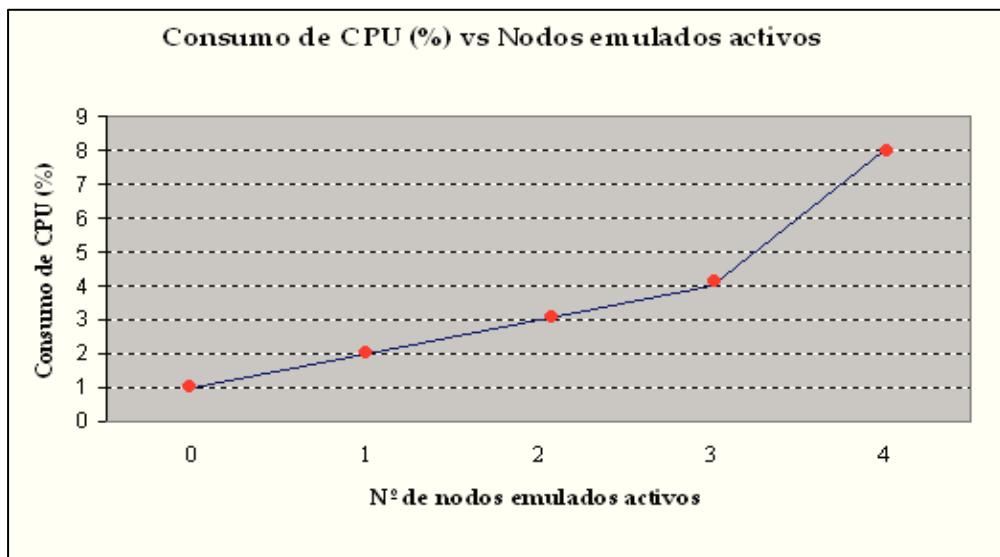


Figura 4.5. Consumo de CPU en el sistema emulador (PC) en función del número de nodos emulados activos para la arquitectura LAN/VTP y *configuración hardware 2*.

4.2.4. Análisis de la latencia para la arquitectura de red LAN Ethernet /VTP.

Se realizan idénticos experimentos que en el *apartado 4.2.2* anterior.

Los análisis efectuados son evaluados para las diferentes cargas de CPU y consumos de memoria RAM reflejados en las configuraciones de las *tablas 4.2* y *4.3*. Para conformar un análisis completo se aborda este análisis de la latencia para instantes de consumo máximo y mínimo de recursos.

Latencia en transmisiones sin control de errores (protocolo ICMP) para la conexión equipo real 1 (Ethernet 802.3) ↔ interfaz SVI de nodo virtual emulado 1.

Implementamos solicitudes de eco desde *equipo real 1 (192.168.2.2)* con destino el interfaz virtual SVI de VLAN2 configurado en el nodo virtual *EtherSwitch router 1 (192.168.2.1)* :

```
ping 192.168.2.1 -n cuenta -1 longitud
```

Las connotaciones del experimento son las mismas que en el *apartado 4.2.2*. De igual manera, las *tablas 4.11* y *4.12*, así como tablas de análisis posteriores, interpretan los *switch-router* activos (valor nominal de las columnas) por el consumo directo de recursos

que implican en el PC emulador, consumos reflejados en las *tablas 4.9 y 4.10* anteriores. Los valores recogidos con formato **XX → XX** hacen referencia a valores de latencia en instantes de tiempo de consumo máximo y mínimo de recursos respectivamente en el PC emulador (carga de CPU y consumo de memoria RAM).

RTT promedio <i>(configuración hardware 1)</i> En milisegundos	1 switch-router activo	2 switch-router activos	3 switch-router activos	4 switch-router activos
Longitud Trama 64 bytes	32 → 30	40 → 34	47 → 40	61 → 49
Longitud Trama 512 bytes	34 → 31	41 → 35	48 → 42	62 → 51
Longitud Trama 1024 bytes	37 → 32	44 → 37	51 → 45	67 → 54
Longitud Trama 1514 bytes	39 → 33	45 → 39	52 → 46	68 → 55

Tabla 4.11. Latencia de red para el protocolo ICMP en la conexión *equipo_real 1 → interfaz SVI de nodo virtual 1*, para diferentes longitudes de trama y consumo de recursos del PC emulador 1 y arquitectura de red LAN/VTP.

RTT promedio <i>(configuración hardware 2)</i> En milisegundos	1 switch-router activo	2 switch-router activos	3 switch-router activos	4 switch-router activos
Longitud Trama 64 bytes	32 → 30	40 → 34	46 → 40	60 → 48
Longitud Trama 512 bytes	34 → 31	41 → 35	47 → 41	61 → 50
Longitud Trama 1024 bytes	36 → 32	43 → 36	50 → 44	66 → 54
Longitud Trama 1514 bytes	38 → 32	44 → 38	51 → 45	67 → 54

Tabla 4.12. Latencia de red para el protocolo ICMP en la conexión *equipo_real 1 → interfaz SVI de nodo virtual 1* para diferentes longitudes de trama y consumo de recursos del PC emulador 2 y arquitectura de red LAN/VTP.

Se constata, como sucede en la arquitectura de red WAN ppp anterior, que la *interoperación entorno real (Ethernet 802.3) \leftrightarrow entorno emulado (interfaz virtual SVI)* presenta una latencia de red muy elevada. Aún así, en el desarrollo de este experimento no se ha producido ningún *TIMEOUT*, esto es, la pérdida de ningún *datagrama ICMP*.

Latencia en transmisiones sin control de errores (protocolo ICMP) en la conexión extremo a extremo: equipo real 1 → entorno virtual → equipo real 2.

Se realiza un experimento similar para evaluar la latencia de la red *extremo a extremo*, esto es, para el trayecto *equipo_real 1 (192.168.2.2) ↔ entorno de red virtual ↔ equipo_real 2 (192.168.2.3)*. Según la configuración del escenario telemático de la figura 3.3 del capítulo 3, el formato del comando ICMP a ejecutar es:

ping 192.168.2.3 -n cuenta -1 longitud
--

Al igual que en el apartado 4.2.2, se realizan 20 repeticiones del experimento, ponderándose la media de los mismos. En este caso hay que destacar que las tramas Ethernet 802.3 se “transforman” en *tagged frames* 802.1q en el enlace troncal, con 4 bytes más en la cabecera para contener/transportar la información adicional relacionada con Ethernet 802.1q, resultando las *tablas 4.13 y 4.14* de la página siguiente .

RTT promedio (configuración hardware 1) En milisegundos	1 switch-router activo	2 switch-router activos	3 switch-router activos	4 switch-router activos
Longitud Trama 802.1q 68 bytes	-	2 → 1-2	4 → 3	5 → 4
Longitud Trama 802.1q 516 bytes	-	2 → 1-2	4 → 3	6 → 5
Longitud Trama 802.1q 1028 bytes	-	2 → 1-2	4 → 3	7 → 6
Longitud Trama 802.1q 1518 bytes	-	3 → 2-3	5 → 4	8 → 7

Tabla 4.13. Latencia de red para el protocolo ICMP en la conexión *equipo_real 1 → entorno emulado → equipo real 2*, para diferentes longitudes de trama, consumo de recursos del PC emulador 1 y arquitectura de red LAN/VTP.

RTT promedio (configuración hardware 2) En milisegundos	1 switch-router activo	2 switch-router activos	3 switch-router activos	4 switch-router activos
Longitud Trama 802.1q 68 bytes	-	2 → 1	3 → 2	3 → 2
Longitud Trama 802.1q 516 bytes	-	2 → 1	3 → 2	4 → 3
Longitud Trama 802.1q 1028 bytes	-	2 → 1	3 → 2	5 → 4
Longitud Trama 802.1q 1518 bytes	-	3 → 2	4 → 3	6 → 5

Tabla 4.14. Latencia de red para el protocolo ICMP en la conexión *equipo_real 1* \rightarrow *entorno emulado* \rightarrow *equipo real 2*, para diferentes longitudes de trama, consumo de recursos del PC emulador 2 y arquitectura de red LAN/VTP.

Los valores de latencia de las *tablas 4.13 y 4.14* reflejan valores más acordes a los esperables teóricamente en este supuesto práctico (1-2 ms). Esto parece evidenciar un mejor comportamiento en la *interoperación entorno real \leftrightarrow entorno emulado* en arquitecturas de red emuladas comutadas a nivel 2, que es el escenario de trabajo al que se circunscriben las tablas anteriores. De igual modo que en la arquitectura WAN ppp, los valores de la latencia son bastante fluctuantes, pudiendo ser atribuibles, en cierta manera, a las fluctuaciones existentes en el consumo de recursos del PC emulador.

En el desarrollo de este experimento no se ha producido ningún *TIMEOUT*, esto es, la pérdida de ningún *datagrama* ICMP.

Capítulo 5: Análisis final del proyecto, conclusiones y líneas futuras de trabajo y/o investigación.

Partiendo de los análisis expuestos en los capítulos 3 y 4, apoyados en los anexos elaborados a raíz del estudio de los entornos de trabajo controlados hardware y software sobre los que se articula el proyecto, se exponen las conclusiones prioritarias del mismo. Finalmente, se proponen determinadas líneas de trabajo y/o investigación futuras para profundizar en el conocimiento de las prestaciones telemáticas potencialmente ofrecidas por el entorno *GNS3/Dynamips*, prestaciones que por profusas y diversas escapan al contenido de este proyecto. La necesidad de conformar una base teórica y práctica sólida sobre la que se sustenten posibles trabajos y análisis futuros es el condicionante principal para que este proyecto se haya concretado intensamente en los contenidos expuestos.

5.1. Conclusiones finales de los análisis de las características, rigor y correspondencia funcional de *GNS3/Dynamips*. Conclusiones de la interoperabilidad entre entornos de red reales y emulados.

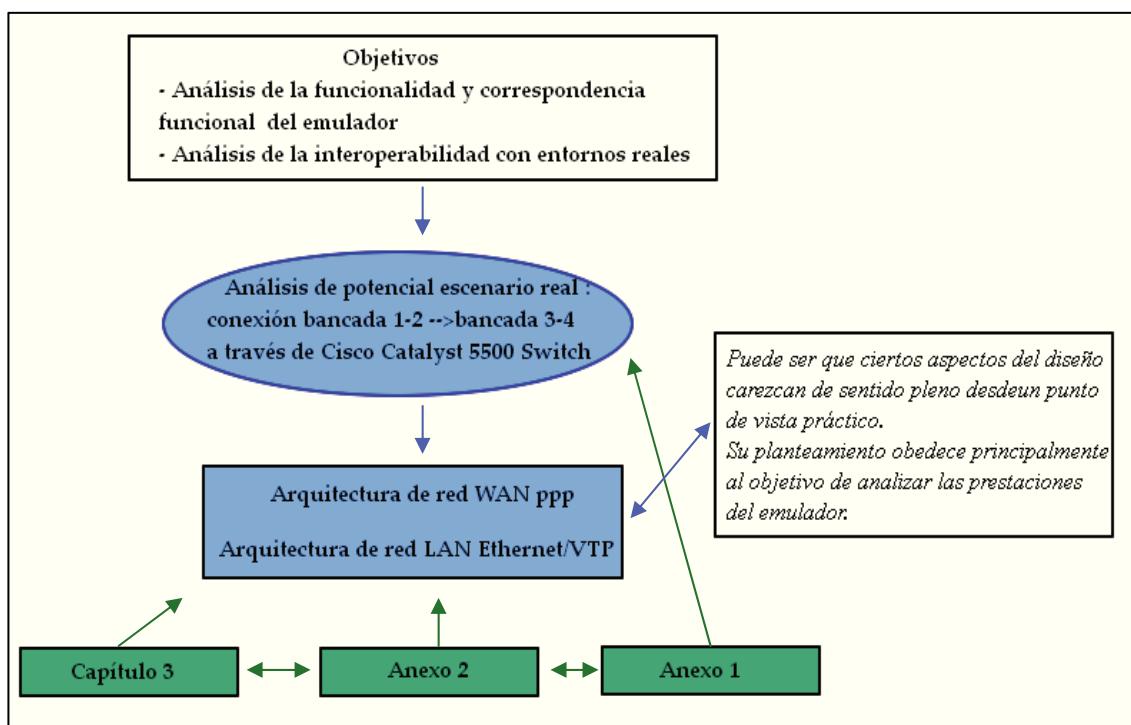


Figura 5.1. Secuenciación del análisis de las características, rigor y correspondencia funcional de *GNS3/Dynamips*.

La tabla 5.1 de la página siguiente expone resumidamente las primeras conclusiones y consecuencias extraídas en el análisis de la *interoperabilidad* y funcionalidad del entorno emulador *GNS3/Dynamips*, hasta nivel 3 según el modelo de referencia OSI.

Conclusiones de la funcionalidad e interoperabilidad de GNS3/Dynamips	Características / Prestaciones →	Consecuencias prácticas/beneficios para el diseño.
Conclusión 1	<ul style="list-style-type: none"> - <i>Interoperación</i> factible y funcional entre dispositivos de diferentes marcas y modelos. Estricta funcionalidad del entorno emulador a nivel 2 y 3. (<i>interoperamos</i> tráfico de red <i>Ethernet</i> enviado/recibido por dispositivos de red reales y emulados, a través del interfaz <i>Ethernet</i> empleado para implementar la <i>interoperación</i>). 	<ul style="list-style-type: none"> - Versatilidad de diseño a nivel 2 y 3. - Adecuado uso de GNS3/Dynamips en ámbitos profesionales y académicos a nivel 2 y 3.
Conclusión 2	<ul style="list-style-type: none"> - Permite segmentar arquitecturas/topologías de trabajo en una “parte real” y una “parte emulada”, según nuestros intereses de diseño y objetivos. 	<ul style="list-style-type: none"> - Versatilidad de diseño. - Aporta un <i>extra</i> al inconveniente de disponer recursos hardware limitados en un entorno de trabajo.
Conclusión 3	<ul style="list-style-type: none"> - Facilidad para implementar las configuraciones emuladas de red en base a comandos <i>Cisco IOS</i> (anexo 2). El interfaz gráfico de GNS3 para el diseño de las topologías resulta cómodo, intuitivo y fácilmente asimilable. 	<ul style="list-style-type: none"> - Adecuada <u>usabilidad</u> de GNS3/Dynamips.
Conclusión 4	<ul style="list-style-type: none"> - Permite fácil y cómodamente generar, y posteriormente capturar y analizar, tráfico de red (mediante la herramienta software <i>Wireshark</i>). 	<ul style="list-style-type: none"> - Adecuado uso del emulador en ámbitos profesionales y académicos para analizar el tráfico de red asociado a las arquitecturas.

Tabla 5.1. Conclusiones en el análisis de la funcionalidad del emulador a nivel 2 y 3 , y de la *interoperabilidad* entre entornos de red reales y emulados.

Para profundizar más en las conclusiones de la tabla anterior se remite al *apartado 7.2.8* y a la *tabla 7.2* del anexo 2. Igualmente, para ahondar más intensamente en las conclusiones sobre la *interoperabilidad*, objetivo prioritario del proyecto, en el *apartado 7.2.4* del mismo anexo 2 se realiza un análisis más exhaustivo mediante *Wireshark* del tráfico de red en el enlace *equipo real 1 ↔ nodo virtual 1*.

Además, el anexo 5 se elabora como complemento *teórico-práctico* intensivo a los análisis de la *interoperabilidad* de los capítulos 3 y 4.

5.2. Análisis de la capacidad, rendimiento y latencia de red de GNS3/Dynamips en Windows.

5.2.1. Conclusiones a los análisis sobre la capacidad y rendimiento del sistema emulador.

Contrastado empíricamente que la *interoperación* no afecta significativamente al consumo de recursos del sistema, se conforman en el capítulo 4 los análisis para el estudio de la dependencia inversa: cómo afecta a la latencia de red (y en cierta manera al comportamiento en tiempo real) el consumo de recursos en el sistema emulador, dependiente este consumo de la mayor o menor complejidad la topología (principalmente, el número de nodos emulados activos). Estos análisis particulares tienen su reflejo en el mismo capítulo 4. A continuación, se muestra en la *tabla 5.3* un resumen de las dependencias halladas en estos análisis.

Las consideraciones oportunas sobre el rendimiento y capacidad del sistema emulador en sistemas operativos Linux (distribución Ubuntu 12.04) se muestran, complementando los análisis del proyecto en Windows, en el anexo 4 de la memoria. A grandes rasgos, en los *apartados 7.4.2* y *7.4.3* de este anexo 4 se constata un ligero descenso en el consumo de recursos del sistema emulador en entornos de trabajo Linux, así como un ligero descenso del valor de la latencia de red (asociado indefectiblemente al descenso en el consumo de recursos).

Conclusiones finales - Tabla de dependencias sobre el consumo de recursos del entorno emulador

Se improvisa en la *tabla 5.1* siguiente una breve relación de dependencias que resume esquemáticamente las conclusiones sobre el consumo de recursos de los análisis del capítulo 4.

Dependencias en el consumo de recursos del sistema emulador	Configuración hardware del PC emulador	Tecnología de la arquitectura de red	Nº de nodos emulados activos (topología de la arquitectura)
Consumo de CPU	ALTA	BAJA	MUY ALTA
Consumo de memoria RAM	MEDIA	BAJA	MUY ALTA

Tabla 5.2. Relación de dependencias en el consumo de recursos del sistema emulador.

La diferencia existente en la dependencia respecto de la *configuración hardware* para los consumos de CPU y memoria RAM (ALTA y MEDIA respectivamente) se supedita al hecho de que al activar en *GNS3/Dynamips* el recurso optimizador de memoria *ghost ios* el consumo de memoria RAM deja de ser crítico en la emulación (para el escenario de trabajo que nos ocupa caracterizado por emular un único y común sistema *Cisco IOS* en todos los dispositivos emulados).

5.2.2. Conclusiones a los análisis de la latencia de red en la *interoperación entorno real ↔ entorno emulado*.

Indefectiblemente asociadas al objetivo principal del proyecto (análisis de la *interoperabilidad*), se establecen las conclusiones a los experimentos de los *apartados 4.2.2* y *4.2.4* en el análisis de la latencia. Los resultados para los dos experimentos considerados son concluyentes en tanto que nos proporcionan información sobre la latencia mínima a considerar, según el tamaño de trama, en la prestación de servicios telemáticos más complejos en las arquitecturas de red emuladas, por el hecho de no implementar control de errores en la transmisión (el protocolo ICMP solo informa de ellos). Esto excluye la circunstancia de que esta latencia de red emulada sea más o menos correcta.

Para el primero de los experimentos propuestos en ambas arquitecturas, las *tablas 4.4* y *4.5* obtenidas en el *apartado 4.2.2*, y las *tablas 4.11* y *4.12* del *apartado 4.2.4* revelan un valor de la latencia de red para todas las longitudes de trama *Ethernet* muy elevado con respecto al valor que se obtiene en un entorno real análogo. Este valor no sería asumible en la práctica sobre un escenario real similar (se recuerda que estamos implementando en este caso comutación a nivel 3).

Aunque estrictamente no es el caso, esta elevada latencia sugiere el hecho de que el potencial valor de la latencia media en una supuesta conexión *extremo-extremo* en topologías emuladas más complejas (mayor número de nodos intermedios o *hops* en la trayectoria), sobre todo para el caso de la arquitectura WAN ppp, incumpliese la recomendación ITU-T G.114 al respecto (valor máximo < 150 ms) [18][19] a la hora de implementar servicios telemáticos avanzados, como VoIP o vídeo multimedia. Estos valores de latencia se obtienen en situaciones de consumo máximo y mínimo de recursos. Llegado el caso, a la hora de extrapolar estos resultados de un escenario de red emulado en *GNS3/Dynamips* sobre un análogo escenario real, podríamos considerar como una cota superior amplia y fiable para la latencia de red real el valor de latencia mínimo obtenido en cada experimento emulado concreto (para cada longitud de trama y consumo de recursos). Sobre esto se profundiza más en el apartado 7.2.8 del anexo 2.

En segundo lugar, se aprecia un comportamiento del emulador con respecto a la longitud de trama acorde al comportamiento teórico en el escenario real, esto es, mayor valor de latencia de red para longitudes de trama mayores (correspondencia funcional).

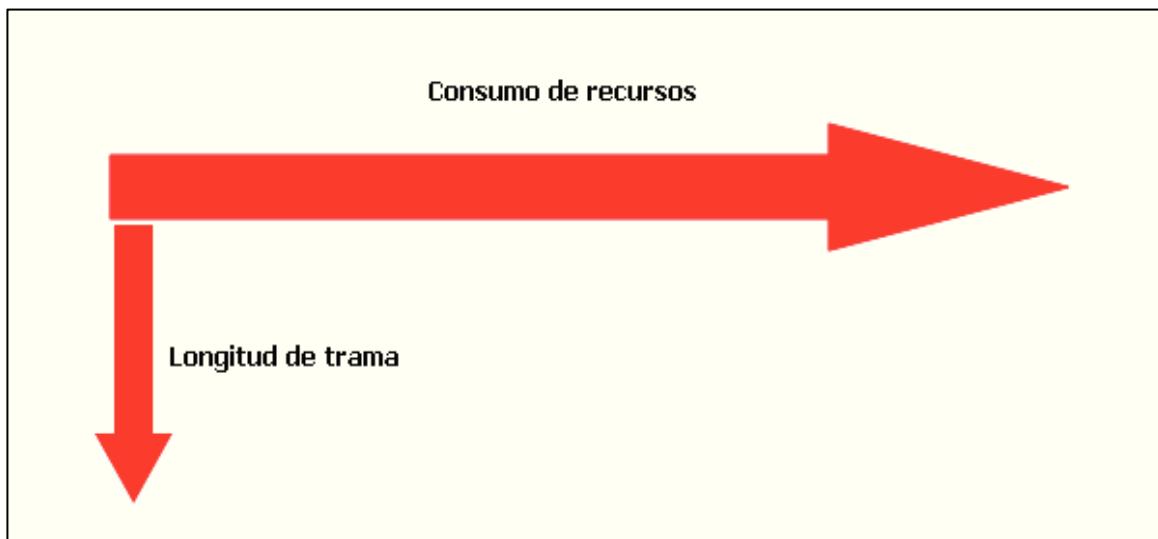


Figura 5.2. Comportamiento del emulador para la latencia de red respecto al consumo de recursos y la longitud de trama para las *tablas 4.4 , 4.5, 4.11 y 4.12* referentes al primero de los experimentos de la latencia en ambas arquitecturas de red.

Del análisis conjunto de estas últimas valoraciones de las *tablas 4.4, 4.5, 4.11 y 4.12* se observa que la dependencia del comportamiento del emulador para la latencia es más crítica y sensible en el sentido de las columnas y hacia la derecha, que en el de las filas y en sentido descendente: el comportamiento del emulador para la latencia de red en este experimento (y consecuentemente trasladable en cierta manera a su comportamiento en tiempo real) resulta más sensible al consumo de recursos del sistema emulador que al tamaño de trama *Ethernet*. De un modo gráfico, esto se muestra en la *figura 5.2* anterior.

De las *tablas 4.7 y 4.8* del *apartado 4.2.2* (arquitectura WAN ppp) podemos extraer parecidas conclusiones a las antes descritas. Pero en este caso, la comunicación entre los nodos emulados se realiza a nivel de red (nivel 3), a través de un enlace *serie WAN ppp* de 1,544 Mb. Ello conlleva que en este caso la dependencia de la latencia sea mayor en el sentido descendente de las filas que de las columnas (aumenta la dependencia con respecto a la longitud de trama, debido a la capacidad de 1,544 MB del enlace *serie ppp*). Aunque los valores de latencia media calculados en este segundo experimento cumplen estrictamente la recomendación ITU-T G.114, límite establecido por ejemplo para prestar servicios VoIP [18], hay que observar este valor con cautela y cierta perspectiva práctica relativa, pues nuestra conexión *extremo-extremo* se circunscribe a dos nodos de red únicamente (*2 hops* en la trayectoria del tráfico).

Para el segundo de los experimentos en la arquitectura LAN Ethernet/VTP, las *tablas 4.13 y 4.14* muestran un descenso contundente en la latencia con respecto al mismo experimento en la arquitectura WAN ppp. En este caso, los resultados son bastante más equiparables a los obtenidos sobre un mismo escenario real, apreciándose igualmente que estos empeoran conforme aumenta el consumo de recursos y el tamaño de la trama. Sobre el escenario emulado LAN Ethernet/VTP conmutado, y para este experimento, estamos implementando conmutación a nivel 2 entre ambos equipos real 1 y 2, equipos adscritos a la misma VLAN, y el emulador presenta un mejor comportamiento en la *interoperación*. Los valores de latencia media obtenidos, aunque superiores a los obtenidos sobre idénticos escenarios reales (aproximadamente entre 1-2 ms), se circunscriben mejor a la recomendación ITU-T G.114 [19] de una manera absoluta, en el hipotético caso de querer implementar servicios telemáticos avanzados de voz, pues su valor está ampliamente dentro de los márgenes de seguridad establecidos, y de una manera relativa, pues la conexión *extremo-extremo* implica dos nodos de red tan sólo, en este caso *2 switch* (un solo *hop* en la trayectoria *equipo real 1 ↔ equipo real 2*).

Las *tablas 5.2 y 5.3* siguientes muestran resumidamente las conclusiones más decisivas e importantes en el análisis de la latencia en la *interoperación*. Un análisis más exhaustivo de las mismas puede contrastarse en el *apartado 7.2.8* del anexo 2.

Conclusiones finales de la latencia – Tabla de dependencias de la latencia

Conclusiones a los análisis de la latencia.	Descripción →→	Consecuencias
Conclusión 1	<ul style="list-style-type: none"> - Los valores calculados de latencia media de red en el entorno emulado son superiores (y en algunos casos muy superiores) a los correspondientes sobre análogos escenarios de red reales. 	<ul style="list-style-type: none"> - Circunscribir nuestros análisis a aproximaciones asociadas con márgenes/cotas superiores de latencia fiables de trabajo.
Conclusión 2	<ul style="list-style-type: none"> - Comportamiento deficiente del emulador en la <i>interoperación</i> de algunas arquitecturas, como las que <i>interoperan</i> tráfico Ethernet con interfaces virtuales SVI (comutación a nivel 3). 	<ul style="list-style-type: none"> - Prudencia a la hora de trasladar el funcionamiento y los resultados en tiempo real de la arquitectura de red emulada y el tráfico con ella relacionado a entornos reales y profesionales de trabajo. Así, el uso de <i>GNS3/Dynamips</i> en este escenario es más adecuado para el ámbito académico que para entornos profesionales.
Conclusión 3	<ul style="list-style-type: none"> - En la arquitectura LAN Ethernet/VTP (arquitectura emulada comutada) el comportamiento de la latencia es relativamente correcto y equiparable, con un cierto margen, a la latencia de red sobre un análogo escenario de red real. 	<ul style="list-style-type: none"> - Comportamiento de la latencia en la <i>interoperación</i> trasladable a escenarios reales (adscrito al margen de trabajo de una cota superior fiable)
Conclusión 4	<ul style="list-style-type: none"> - A pesar de lo anterior, la correspondencia funcional del emulador es estricta y acorde con escenarios de red reales análogos a nivel 2 y 3. 	<ul style="list-style-type: none"> - En cuanto a la funcionalidad de una arquitectura emulada se constata el uso correcto y adecuado del emulador a nivel 2 y 3.
Conclusión 5	<ul style="list-style-type: none"> - Los dos experimentos de la latencia nos permiten <u>fijar una latencia de red mínima</u> emulada en función de la longitud de trama. 	<ul style="list-style-type: none"> - Permite fijar una <u>latencia mínima</u> a considerar sobre <u>escenarios emulados</u> de trabajo .

Tabla 5.3. Conclusiones principales a los análisis de la latencia en la *interoperación*.

	Longitud de trama	Configuración hardware	Arquitectura de red →→	Nº de nodos emulados activos
Dependencias de la latencia de red en GNS3/Dynamips (interoperación entorno real ↔ entorno emulado)	MEDIA (similar al entorno real)	MEDIA-ALTA	↓ ALTA (correcta/incorrecta elección teórico/práctica de la tecnología a implementar)	MUY ALTA (relacionado con el consumo de recursos)

Tabla 5.4. Relación de dependencias de la latencia de red en *GNS3/Dynamips*.

Las relaciones anteriores se interpretan gráficamente en la *figura 5.3*. Sobre esta figura se interpreta el grosor de los trazos que marcan las dependencias como indicativo de una mayor o menor dependencia. En la misma es conveniente señalar que la dependencia del valor de la latencia media de la red indirectamente con respecto a la arquitectura se circumscribe al hecho de que un mal diseño de la arquitectura y su topología asociada (dependencia indirecta *vía* consumo de recursos); una incorrecta elección de la tecnología para prestar un determinado servicio telemático (dependencia directa) puede implicar unos valores de latencia considerablemente elevados para la prestación del servicio, pudiendo quedar fuera de los márgenes recomendables establecidos por las recomendaciones ITU-T para la latencia *extremo a extremo* [19].

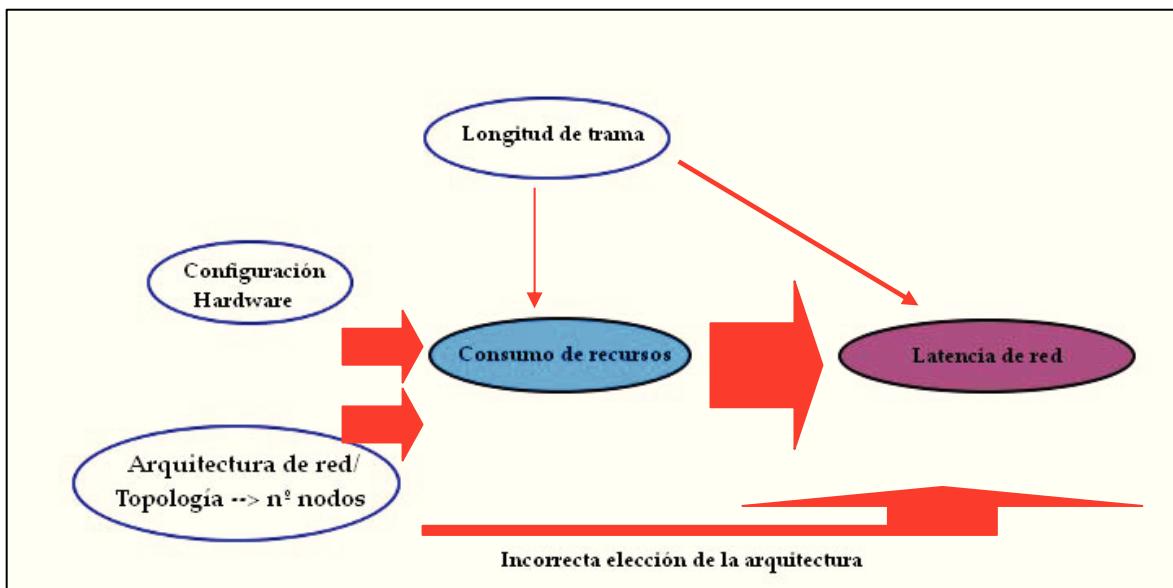


Figura 5.3. Interpretación gráfica de las dependencias existentes en la latencia de red emulada en la prestación de un determinado servicio telemático.

5.3. Inconvenientes principales asociados al desarrollo del proyecto. Líneas futuras de trabajo y/o investigación.

5.3.1. Inconvenientes principales en el desarrollo del proyecto y soluciones adoptadas.

El principal inconveniente, tal como se describe en el *apartado 2.4*, hace referencia a la imposibilidad actual de emular estrictamente dispositivos de la plataforma hardware *Catalyst* de *Cisco Systems* en *GNS3/Dynamips*. Relacionado con esta imposibilidad, la de implementar/emular sistemas *Cisco CatOS* en el entorno emulado. Es decir, en lo que a dispositivos *Cisco Systems* se refiere, *GNS3/Dynamips* solo emula sistemas *Cisco IOS*, por lo que aunando los dos inconvenientes deben analizarse y seleccionarse un adecuado dispositivo de red/sistema *Cisco IOS* para emular las características y prestaciones potenciales del hardware de red *Cisco Catalyst 5500 Switch* del laboratorio, en base al dispositivo *multilayer EtherSwitch router* de *GNS3/Dynamips*. Así mismo, debe conformarse en el entorno emulador sobre el dispositivo emulado *EtherSwitch router*, una arquitectura modular hardware similar a la del dispositivo real, disponiendo para ello de módulos virtuales de interfaces, reflejo de módulos hardware existentes en el mercado (para profundizar sobre esto puede consultarse el anexo 2, *apartado 7.2.1*).

Añadido a los inconvenientes anteriores, y aunque no puede catalogarse como una contingencia propiamente dicha, el desarrollo del proyecto ha exigido un profundo estudio y asimilación de contenidos *teórico-prácticos* relacionados con el dispositivo de red *Cisco Catalyst 5500 Switch*, de un modo particular, y del hardware de red *Cisco Systems* en general, además de los sistemas operativos asociados *Cisco CatOS* y *Cisco IOS*. Cabe destacar sobre el estudio de este dispositivo de red que al no disponer de información ni reseñas en el laboratorio sobre el mismo, la búsqueda de información, bibliografía adecuada, *datasheets* y manuales de uso ha sido una tarea ardua, debido a que la información existente en la *red es amplia y variada*. Acotar la conveniencia de la misma a nuestros propósitos de trabajo fue un proceso laborioso y complejo, por tratarse de hardware de cierta antigüedad sobre el que *Cisco Systems* no ofrece actualmente soporte. Es igualmente reseñable que el estudio en profundidad de los sistemas *Cisco IOS* ha sido absolutamente fundamental para configurar las arquitecturas emuladas en *GNS3/Dynamips*. En menor medida, también ha sido importante la asimilación del sistema operativo *Cisco CatOS*. El estudio de este último ha sido decisivo para poder trasladar al entorno emulado determinadas configuraciones que en el dispositivo *Cisco Catalyst 5500 Switch* se implementan indefectiblemente con comandos *Cisco CatOS*, configuraciones reflejadas en el anexo 1. Al respecto, los análisis anteriores, y la consiguiente elaboración del anexo 1, han servido para configurar los equipos *Cisco Catalyst 5500 Switch* actualmente utilizados en el laboratorio de telemática del DIEC.

Otro inconveniente a destacar en el desarrollo del proyecto es el que hace referencia a la correcta elección de la versión del entorno software *GNS3/Dynamips*, ya que se constató experimentalmente que versiones actuales del software presentaban ciertas

inestabilidades en su funcionamiento. Finalmente, tras numerosos experimentos y análisis, se optó por la versión 0.7.3 del mismo, versión que garantizó una adecuada estabilidad.

5.3.2. Líneas futuras de trabajo y/o investigación relacionadas con el proyecto.

La figura 5.4 muestra esquemáticamente posibles líneas de trabajo y/o investigación futuras relacionadas con este proyecto.

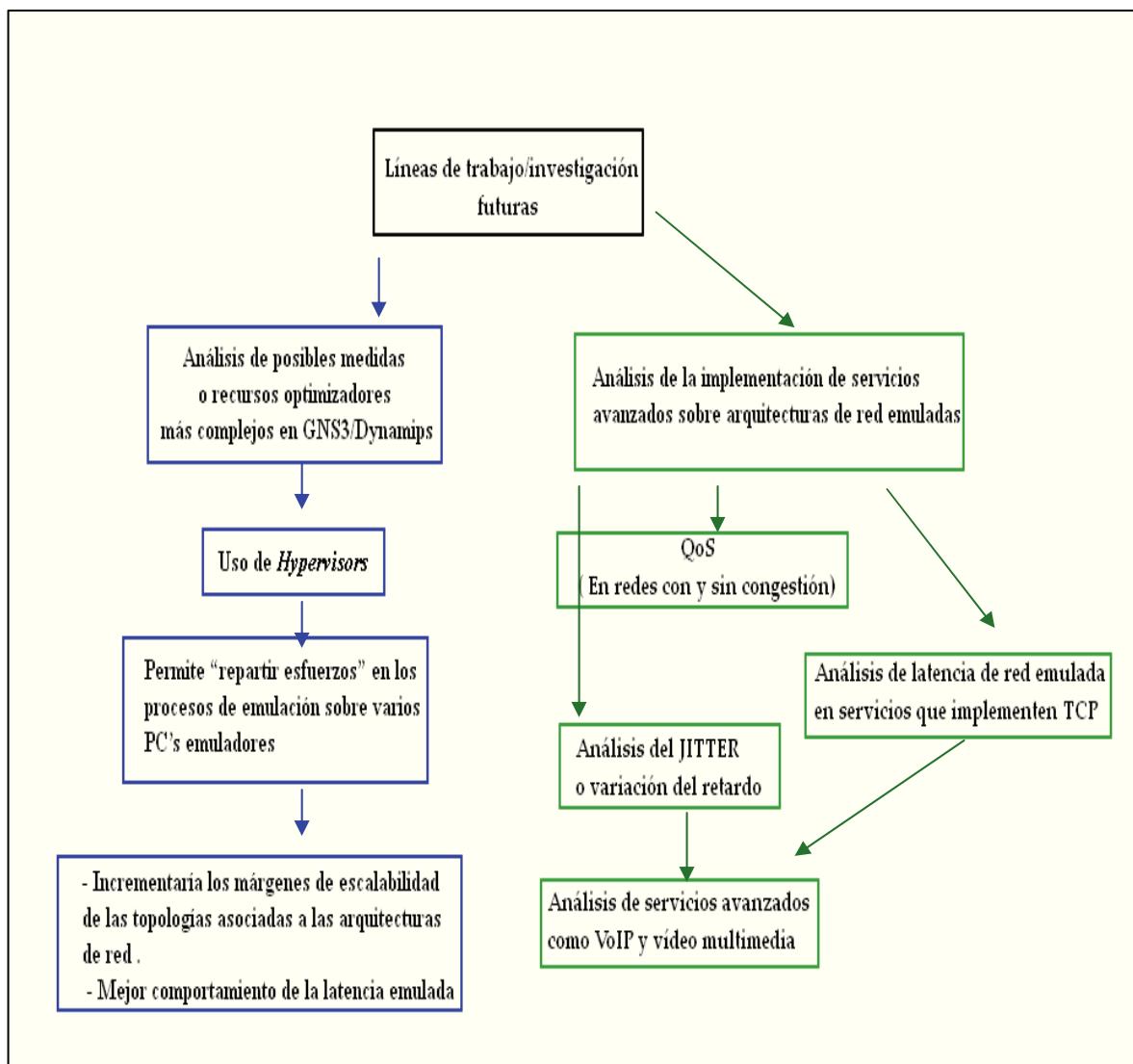


Figura 5.4. Representación esquemática de las posibles líneas de investigación futuras.

Capítulo 6: Referencias bibliográficas y en línea.

- [1] Wendell, O. (2004). *CCNA ICND certification guide*. Indianapolis, USA: Cisco Press.
- [2] Ranjbar, A. S. (2001). *CCNP certification guide*. Indianapolis, USA: Cisco Press.
- [3] Cisco Systems, Inc (n. d.). *Configuring InterVLAN Routing Using an Internal Router (Layer3 Card) on Catalyst 5500/5000 and 6500/6000 Switches that Run CatOS System Software*. Recuperado en Octubre-Noviembre de 2012 desde http://www.cisco.com/en/US/products/hw/switches/ps700/products_tech_note09186a008015bf9a.shtml
- [4] Cisco Systems, Inc (n. d.). *Catalyst 5000 Family Module Install Guide, Chapter 7, Route Switch Module WS-X5302*. Recuperado en Octubre-Noviembre de 2012 desde <http://www.cisco.com/univercd/cc/td/doc/product/lan/cat5000/hardware/modules/07rsm.pdf>
- [5] Cisco Systems, Inc (n. d.). *Supervisor Engine III and Uplink Module Installation & Configuration Note*. Recuperado en Noviembre de 2012 desde http://www.cisco.com/en/US/products/hw/switches/ps679/products_installation_and_configuration_guide09186a00800d9ee3.html
- [6] Cisco Systems, Inc (2007). *8-port Multichannel T1/E1 PRI Port Adapter Installation and configuration*. San Jose, USA: Americas Headquarters.
- [7] Cisco Systems, Inc (n. d.). *Cisco Catalyst 5500 Series Switches Datasheets & manual*. Recuperado en Octubre de 2012 desde <http://www.cisco.com/en/US/products/hw/switches/ps686/index.html>
- [8] Cisco Systems, Inc (n. d.). *Cisco IOS IP Configuration Guide, Release 12.2*. Recuperado en Noviembre de 2012 desde http://www.cisco.com/en/US/docs/ios/12_2/ip/configuration/guide/1cfrip.html
- [9] Cisco Systems, Inc (n. d.). *Cisco 3700 series Routers*. Recuperado en Noviembre de 2012 desde <http://www.cisco.com/en/US/products/hw/routers/ps282/index.html>
- [10] Cisco Systems, Inc. (2006). *Cisco Networking Academy program, CCNP: Building Multilayer Switching Networks v.5.0*, San Jose, USA: Americas Headquarters.
- [11] Cisco Systems, Inc. (2003). *Programa de la Academia de Networking de Cisco – CCNA 2*. San Jose, USA: Americas Headquarters.
- [12] Graphical Network Simulation – GNS3 – Documentation (n. d.). Recuperado en Octubre-Noviembre-Diciembre de 2012 desde <http://www.gns3.net/documentation>

- [13] Graphical Network Simulation – GNS3 – Dynamips (n. d.). Recuperado en Octubre-Noviembre-Diciembre de 2012 desde <http://www.gns3.net/dynamips>
- [14] Documentación *en línea* de Wireshark (n.d.). Recuperado en Octubre-Noviembre-Diciembre de 2012 desde <http://www.wireshark.org>
- [15] File Transfer Protocol (n.d.). En *Wikipedia*. Recuperado en Diciembre de 2012 de http://es.wikipedia.org/wiki/File_Transfer_Protocol
- [16] Documentación *en línea* de Filezilla (n.d.). Recuperado en Octubre y Diciembre de 2012 y en Enero de 2013 de <http://wiki.filezilla-project.org/Documentation>
- [17] Documentación *en línea* y foros de TCPOptimizer. Recuperado en Enero y Febrero de 2013 desde <http://www.speedguide.net/downloads.php>
- [18] Felici, S. (n.d.). *Evaluación de mecanismos de calidad de servicio en los routers para servicios multimedia*. Curso de Doctorado Sistemas y Servicios Telemáticos Departamento de Informática, Universidad de Valencia. Recuperado en Enero de 2013 desde <http://informatica.uv.es/docto-2-qos.ppt>
- [19] Unión Internacional de Telecomunicaciones (n.d.). *Normalización UIT-T*. Recuperado en Febrero de 2013 de <http://www.itu.int/ITU-T/sitemap/index.asp>
- [20] Tutorial Python-qt4 (n.d.). Recuperado en Enero de 2013 desde <http://linuxapuntes.blogspot.com.es/2010/08/python-qt-4-pyqt4.html>

Capítulo 7: Anexos

7.1. Anexo 1: Configuración & gestión de plataformas hardware modulares de red *Cisco Catalyst 5500 Switch* (y similares)

7.1.1. Introducción.

El presente anexo pretende ser un breve y preciso manual para abordar diferentes procesos de configuración del dispositivo de red *Cisco Catalyst 5550 Switch*, base del proyecto, y de un modo general sobre hardware de red de *Cisco Systems*. Por lo tanto, es el resultado del proceso de adquisición de habilidades del entorno de trabajo controlado del laboratorio de telemática, necesario para el desarrollo del proyecto. Habida cuenta de la densidad y profusión de contenidos que implicaría un análisis exhaustivo del hardware en cuestión, con la totalidad de sus potenciales prestaciones, el presente anexo pretende habilitar en las tareas comunes y fundamentales de diseño consideradas en el proyecto.

La serie *Cisco Catalyst 5500 Switch* es una plataforma hardware modular con una capacidad de hasta 13 módulos *interoperables* y gestionables, módulos enumerados desde la parte superior del equipo a la inferior del 1 al 13, y que confiere así una gran versatilidad de diseño y potencial escalabilidad a la hora de abordar diferentes escenarios tecnológicos que a día son susceptibles de considerar en escenarios telemáticos LAN/WAN. Para acceder a la gestión principal de este entramado hardware modular, que permite configurar escenarios de *routing & switching*, se precisa de un módulo supervisor que en este caso, en la disposición de los equipos manipulados en el laboratorio, es el módulo supervisor *Cisco WS-X5530-E3*. Dichos equipos *Cisco Catalyst* cuentan con dos módulos supervisores, físicamente contiguos entre sí, actuando uno de los mismos como módulo activo (*active module*), ejerciendo el otro módulo supervisor de *standby module*, redundante, confiriendo así seguridad en caso de falla mediante el respaldo de la configuración del sistema. Todo lo configurado y gestionado en el *active module WS-X5530-E3* queda respaldado en el *standby module* contiguo, siendo posible la transición en un mismo módulo supervisor de un rol a otro, a voluntad del administrador de red, utilizando para ello la instancia:

- Console>(enable) switch supervisor

Dicho comando, ejecutado en el *active module*, intercambia el rol de los módulos supervisores, pasando así el módulo supervisor activo del estado *active* a estado *standby*, y viceversa. En principio, para acceder a la gestión del *Cisco Catalyst 5500 Switch* nos conectaremos *vía* puerto consola al módulo supervisor activo. Si intentásemos cursar operaciones de administración y gestión *vía standby module*, obtendríamos el siguiente mensaje por pantalla:

*This module is now in standby mode.
Console is disabled for standby supervisor*

Visualmente, el rol ejecutado por los módulos supervisores queda patente por la iluminación del LED a tal efecto (módulo activo → verde).

7.1.2. Modos de operación en arquitecturas *Cisco Systems*.

Se establecen diferentes *modos de operación* y funcionalidades asociadas, según sea el sistema operativo que “gobierne” el hardware. Según esto, encontramos dos posibilidades:

Sistemas basados en Cisco IOS

Sus *modos de operación* o *configuración* son:

Modo usuario: es el primer modo de acceso. Como *prompt* (o interfaz del intérprete de comandos), se presenta por pantalla:

- nombre_equipo>

Modo EXEC privilegiado: se accede al mismo desde el *modo usuario* mediante el comando: `enable`, permitiendo a los usuarios ver la configuración del equipo de red (*router* o *switch*), restablecer el equipo e ingresar al *modo de configuración global*. Como *prompt* o interfaz del intérprete de comandos en pantalla tenemos:

- nombre_equipo#

El administrador puede posibilitar la petición de una contraseña para acceder al equipo en *modo usuario y/o privilegiado*.

Modo de configuración global: permite modificar la configuración del sistema. El acceso al *modo de configuración global* se realiza desde el *modo EXEC privilegiado* mediante el comando: `configure terminal` (ó simplemente: `conf t`).

Sistemas basados en Cisco CatOS (más antiguos)

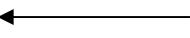
En el caso de sistemas basados en *Cisco CatOS*, el *modo de operación* para tareas de configuración y gestión es el denominado *enable-mode*, cuyo *prompt* es:

- Console> (enable)

Se accede al mismo mediante el comando: `enable` , ejecutado desde :

- `Console>`

, que es el primer *modo de operación* que nos encontramos al acceder a equipos gestionados con sistemas operativos *Cisco CatOS*.

En ambos sistemas operativos, siempre que deseemos volver/retroceder a un *modo de operación* que precede al modo en que nos encontramos (como por ejemplo, retornar de *modo usuario* → *modo EXEC privilegiado* → ...) 

, utilizaremos el comando: `exit` .

El comando: `end` , nos devuelve, en todo caso, al *modo de operación EXEC privilegiado* (en *Cisco IOS*), independientemente del *modo de operación* en que nos encontramos.

7.1.3. Conexión y configuración vía módulo supervisor *Cisco WS-X5530-E3*.

Para proceder a la gestión & administración de las arquitecturas *Cisco Catalyst 5500 Switch* y similares, nos conectaremos (fundamentalmente, ya que existen otras alternativas) al puerto consola presente en el módulo supervisor activo mediante el empleo de un cable consola *interfaz Db-9* → *RJ-45 directo*. Es preciso incidir en esto último, pues actualmente está extendido en prácticamente todas las arquitecturas y familias *Cisco Systems* el empleo para tales fines administrativos del cable consola *rollover*; pero en el caso que nos ocupa difiere manifiestamente la disposición del cableado, siendo el que antes se detalló.

Otra diferencia ciertamente relevante y decisiva a la hora de implicarnos en el manejo del módulo supervisor *Cisco WS-X5530-E3*, es el uso de *Cisco CatOS* como sistema operativo a tal efecto, a diferencia de *Cisco IOS*, que impera en la mayor parte del hardware de red actual de *Cisco Systems* (y en el entorno software *GNS3/Dynamips*).

La funcionalidad de ambos sistemas operativos, similar (o idéntica). ¿ Diferencias ? Meramente formales. La transición *Cisco CatOS* a *Cisco IOS* es fundamentalmente una decisión de matiz corporativo con objeto de dotar a todas las arquitecturas de un mismo entorno software de configuración y gestión, repercutiendo directamente en una mejor asimilación y *usabilidad* software en entornos que operan con distinto hardware (buscando el objetivo final de disponer de un único software de gestión común para todas).

Principales comandos Cisco CatOS para administración & gestión en arquitecturas hardware Cisco Catalyst 5500 Switch vía módulo supervisor Cisco WS-X5530-E3.

Una vez accedemos, *vía* consola, al módulo supervisor [5] a través del emulador de Terminal, con la configuración que se detalla en la figura 7.1, se muestra por pantalla el *prompt* para el *modo de operación usuario*. Desde aquí, mediante el ingreso del comando:

- Console> enable
, se accede al *modo de operación enable-mode*; visualmente, su *prompt* es:
- Console>(enable)

En caso de estar protegido el acceso por contraseña, se nos solicitará su ingreso. Una vez en el *modo de operación enable-mode*, con el ingreso del comando: ?, obtendremos un listado de todos las posibles instrucciones susceptibles de ser ejecutadas desde el *modo de operación* en que nos encontramos, así como todas las opciones existentes para un comando, si ingresamos : ?, al final de un determinado comando (del que desconocemos las opciones completas que presenta para su ejecución). Interesante es destacar que mediante el uso de *tabulador*, el sistema busca aquella instrucción de posible ejecución que comprenda los caracteres ingresados hasta el momento desde el teclado (al igual que ocurre en otros sistemas operativos conocidos, como Unix/Linux).

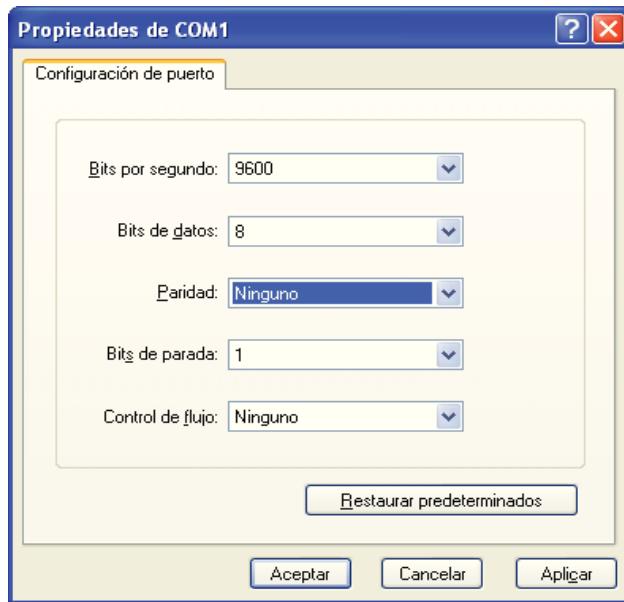


Figura 7.1. Configuración de *HyperTerminal* para la conexión al módulo supervisor Cisco WS-X5530-E3.

Desde el *modo de operación enable-mode*, podemos acceder a la disposición física y configuración de los módulos hardware presentes en la arquitectura *Cisco Catalyst*:

- `Console>(enable) show modules`
, o podemos disponer de una información completa del mismo mediante:
- `Console>(enable) show version`
- `Console>(enable) show module 5`
, nos ofrece información (en este caso concreto) del módulo hardware nº 5.
- `Console>(enable) show config`
, muestra la configuración actual del sistema.
- `Console>(enable) show port [nºmodulo / nº interfaz]`
, ofrece información del interfaz nº_{interfaz} ubicado en el módulo nº_{modulo}.
- `Console>(enable) set module name [nºmodulo mi_nombre]`
, da nombre *mi_nombre* al módulo *nºmodulo*.
- `Console>(enable) set port name [nºmódulo / nºinterfaz mi_nombre]`
, da nombre *mi_nombre* al interfaz *nºinterfaz*, del módulo *nºmódulo*.
- `Console>(enable) set port [nºmódulo / nºinterfaz] [enable , disable]`
, activa o desactiva el interfaz correspondiente; por defecto, los interfaces se encuentran activos en *Cisco CatOS*.
- `Console>(enable) set port [nºmodulo / nºinterface] duplex`
, configura el interfaz correspondiente en modo *full dúplex*.
- `Console>(enable) set port speed [velocidad]`
, establece la capacidad (bps) del interfaz indicado.

A través del módulo supervisor podemos acceder a otros módulos presentes en nuestro equipo *Cisco Catalyst 5500 Switch*, módulos que precisen de configuración y/o gestión, sin tener que conectarnos vía consola a dicho módulo específico (ni tener que desconectarnos, por lo tanto, del módulo supervisor):

- `Console>(enable) session [nº modulo]`
, nos conecta al puerto consola del módulo *nºmódulo* para su gestión y configuración.

Tras esto, una vez en el módulo hardware en cuestión, la instrucción: `exit` , ejecutada desde el *modo de operación usuario* (Router>exit), nos devuelve al módulo supervisor desde el que habíamos accedido previamente.

Para anular/eliminar una instrucción/comando de configuración previamente ejecutado, se utiliza el mismo formato de comando, pero sustituyendo `set` → `clear` .

No es necesario guardar la configuración en funcionamiento para poder disponer de ella después de cada puesta en marcha o reinicio, pues *CatOS* salva automáticamente la misma tras presionar la tecla “*Intro*” después de cada instrucción, a diferencia de *Cisco IOS*, que precisa hacer la copia explícita de la configuración (con el comando: `copy run start`)

Comandos Cisco CatOS para la configuración de escenarios VLAN desde el módulo supervisor Cisco WS-X5530-E3.

Desde el *modo de operación enable-mode* del módulo supervisor del *Catalyst 5500*, ejecutamos las tareas de creación/configuración VLAN y asignación física de interfaces a las mismas, así como tareas de configuración VTP (si procediese):

- Console>(enable) `set vlan 100`
- Console>(enable) `set vlan 200`
, creamos vlan 100 y vlan 200 .
- Console>(enable) `set vlan 200 8/13-24`
, asigna a vlan 200 los puertos / interfaces 13 al 24, ambos inclusive, pertenecientes al módulo nº 8 .
- Console>(enable) `set vlan 100 8/0-12`
, asigna a vlan 100 los puertos / interfaces 0 al 12, ambos inclusive, pertenecientes al módulo nº 8 .
- Console>(enable) `set vtp mode [client | server | transparent]`
, asigna (en el ámbito de un dominio VTP) el rol del dispositivo.
- Console>(enable) `set vtp domain [mi_nombre]`
, asigna/relaciona el *Catalyst 5500* con el dominio VTP *mi_nombre* (si el *switch* es *servidor* o *cliente*, no en *modo transparente*)[2] [10] .

Una vez acometidas las tareas de gestión VLAN anteriores, se procede con la configuración a nivel 3, para conferir a nuestro sistema de capacidades de *InterVLAN*

routing que presentan los equipos *multilayer Cisco Catalyst 5500 Switch* (y similares). Este aspecto va a ser fundamental en las configuraciones a implementar en el entorno emulado *GNS3/Dynamips*. Para ello, la disposición hardware de los equipos utilizados en el laboratorio presenta dos módulos hardware *Cisco RSM WS-X5302 (Route-Switch Module)*, independientes entre sí.

Dichos módulos en nuestro caso, adicionalmente ejercen control sobre sendos módulos *Cisco 8-port Multichannel T1/EI PRI*, debiendo estar estos físicamente contiguos a su correspondiente módulo RSM “gestor” (así es requerido desde un punto de vista de configuración hardware por la arquitectura *Cisco Catalyst 5500 Switch*)^[4]. De este modo, y bajo un prisma de diseño LAN/WAN, la versatilidad de nuestro dispositivo para ofrecer prestaciones a nivel 2 y 3 según el modelo de referencia OSI sobre escenarios de *Internetworking* queda de manifiesto.

A continuación, nos ocupamos de las prestaciones del módulo RSM para implementar *InterVLAN routing (switching a nivel 2 - 3)*, y *routing* entre dispositivos de nivel 3 .

7.1.4. Configuración del módulo *Router-Switch Module Cisco WS-X5302*.

Gestión & configuración VLAN

Tras la creación/configuración de VLAN (y VTP, si procede), y la correspondiente asignación de interfaces a las mismas desde el módulo supervisor, accedemos al módulo *Cisco RSM WS-X5302* ^[4] para acometer las tareas de *switching & routing* entre VLAN. La gestión de dicho módulo RSM se realiza por medio de comandos *Cisco IOS*, no *CatOS* .

Desde el módulo supervisor (en *modo usuario* o *enable-mode*), ejecutamos:

- Console> session 4
- Router>

, redirigiéndonos el comando anterior al módulo nº4 de nuestro dispositivo.

En nuestro caso concreto, los equipos *Cisco Catalyst 5500 Switch* del laboratorio de telemática, disponen de dos módulos RSM ubicados en respectivas *bahías/slots* nº 4 y nº 6 (pudiendo acceder a cualquiera de ellos).

¡Importante! Dichos módulos RSM son independientes, y por lo tanto así sus configuraciones. Como refleja la figura 7.2, el módulo RSM nº 4 controla y opera sobre el módulo Cisco 8-port Multichannel T1/E1 PRI situado en el módulo nº 3 contiguo del Cisco Catalyst, lo mismo que el módulo RSM nº 6 opera sobre el módulo nº5 Cisco 8-port Multichannel T1/E1 PRI . Pero ambos módulos nº 4 y nº 6 pueden operar, indistintamente, sobre los módulos hardware nº 7 y nº 8, que son módulos que disponen de 24 puertos FastEthernet cada uno.

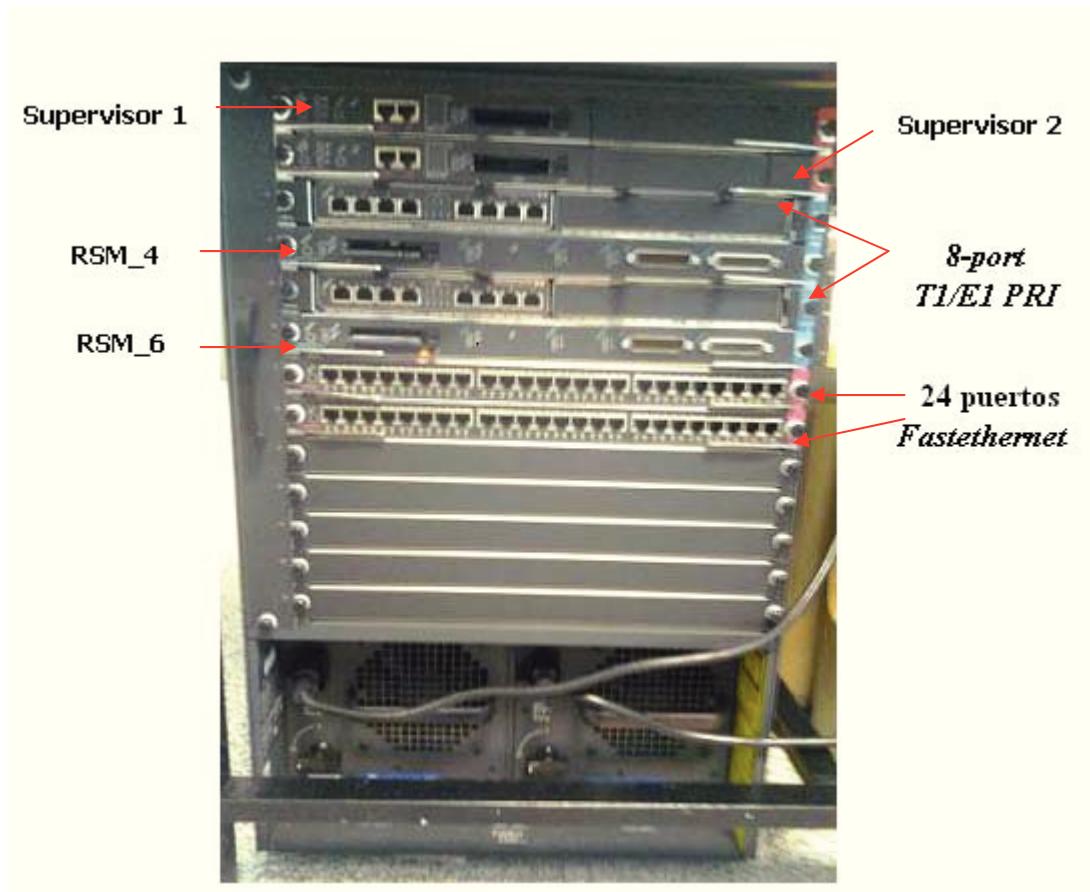


Figura 7.2. Dispositivo de red *Cisco Catalyst 5500 Switch* presente en el laboratorio de Telemática del DIEC.

Resulta interesante, y muy recomendable, dar nombre a los módulos RSM para evitar confusiones y facilitar el trabajo futuro:

- Router> enable
, accedemos al *modo de operación EXEC privilegiado*.
- Router# configure terminal
, o simplemente: conf t , accedemos al *modo de operación global*.
- Router(config)# hostname RSM4

A partir de aquí, para la configuración de los interfaces virtuales SVI asociados a cada VLAN, interfaces que nos permitirán realizar *switching* a nivel 3 entre VLAN diferentes (sin necesidad de acceder a un *router* externo mediante un enlace troncal con protocolo 802.1q /ISL), se accede al *modo de configuración global* → *de interfaz VLAN*:

- RSM4 (config) # interface vlan 100
- RSM4 (config-if) # ip address 192.168.100.1 255.255.255.0 , asignamos una dirección IP al correspondiente SVI, considerando que vlan 100 tiene la dirección de red IP 192.168.100.0 / 24
- RSM4 (config-if) # no shutdown , activamos el interface SVI.
- RSM4 (config-if) # exit , pasamos al *modo de operación global* para proceder con vlan 200
- RSM4 (config) # interface vlan 200
- RSM4 (config-if) # ip address 192.168.200.1 255.255.255.0
- RSM4 (config-if) # no shutdown , activamos el interfaz SVI .
- RSM4 (config-if) # exit , pasamos del *modo de operación global* → *de interfaz* al *modo global*.
- RSM4 (config) #

Nótese que por claridad en la gestión y administración futuras se ha asignado al tercer octeto de las direcciones de red IP de las VLAN el número identificador de la VLAN correspondiente.

Para habilitar las funciones de enrutamiento/*switching* (a nivel 3) vía RSM :

- RSM4 (config) # ip routing

Si precisamos habilitar algún protocolo dinámico de enrutamiento para establecer la comunicación con “el mundo exterior” (a nivel 3):

- RSM4 (config) # router rip
, modo de operación global → de protocolo.
- RSM4 (config-router) # version 2 .
- RSM4 (config-router) #network [dirección IP de la red mayor del interfaz y sin máscara]

, activamos en este caso RIP v2; habilitamos, para enviar y recibir actualizaciones de enrutamiento los interfaces correspondientes, con dirección IP de la *red mayor* a la que pertenece el interfaz. Por defecto, los interfaces están inactivos en este ámbito de trabajo, de ahí la necesidad de activarlos.

Para añadir rutas estáticas de enrutamiento:

- RSM4 (config-router) # ip route [red destino] [mascara]
[próximo salto] [distancia administrativa]

, por defecto, de cara a la tramitación de información de *routing*, las rutas estáticas tienen distancia administrativa = 1.

, y si añadimos una *ruta por defecto*:

- RSM4 (config-router) # ip route 0.0.0.0 0.0.0.0 [próximo salto]

Para salvar toda la configuración de nuestro módulo RSM, y desde el *modo de operación EXEC privilegiado*:

- RSM4# copy run start
, modo abreviado de: copy running-config startup-config .

Para visualizar nuestra configuración actual:

- RSM4# show running-config ,
, o simplemente, con el comando abreviado: show run .

7.1.5. Configuración de Interfaces E1 en módulos *Cisco 8-port Multichannel T1/E1*. Uso como comutador Frame-Relay.

Planteamos de un modo práctico la configuración sobre el módulo hardware nº 3 de nuestra arquitectura *Cisco Catalyst 5500 Switch* del laboratorio, que es el módulo *Cisco 8-port Multichannel T1/E1 PRI* [6] gestionado por RSM4 (recordemos, RSM4 gestiona su módulo contiguo nº 3, y RSM6 gestiona su módulo contiguo nº 5). Configuramos los interfaces E1 (*estructurados*, según estándar *G.704* [19]) que precisemos a través de los siguientes comandos *Cisco IOS*:

- Console > session 4
- RSM4> enable
- RSM4# configuration terminal
, o simplemente: conf t .
- RSM4 (config)# controller e1 0/0
, accedemos al slot 0 / interfaz 0 (de los 8 interfaces E1 controlados por RSM4)
- RSM4 (config-controller) # framing no-crc4
- RSM4 (config-controller) # channel group 0 timeslot 1 - [timeslot]
, con ello hemos creado el interfaz serial 0/0:0
- RSM4 (config-controller) #exit
- RSM4 (config) #interface serial 0/0:0
- RSM4 (config-if) #no shutdown
- RSM4 (config-if) #bandwidth [ancho de banda]
- RSM4 (config-if) #no ip address
- RSM4 (config-if) #encapsulation frame-relay ietf
- RSM4 (config-if) #no keepalive
, desactivamos LMI.
- RSM4 (config-if) #exit

En el supuesto de querer configurar interfaces E1 *desestructurados (unframed)*:

- RSM4# configuration terminal
, o simplemente: conf t.
- RSM4 (config) # controller e1 **0/0**
, accedemos al slot 0, interfaz 0 (de los 8 interfaces E1 que disponemos)
- RSM4 (config-controller) # channel group **0** unframed
, con ello hemos creado el interfaz serial **0/0 : 0 desestructurado (unframed)**

Una vez creado el interfaz *serial*, procedemos igual que antes.

- Cisco Catalyst 5500 Switch : uso como comutador WAN *Frame Relay*

Con la finalidad de profundizar en el funcionamiento del *Cisco Catalyst 5500 Switch* como comutador *Frame Relay*, se plantea el supuesto práctico básico de interconectar vía *Frame Relay* dos *router* del laboratorio 2.03, tal como muestra la figura 7.3.

Para ello, en primer lugar deberemos configurar sobre el módulo 8-port *Multichannel T1/E1 PRI* (módulo nº 3 del *switch*, por ejemplo), 2 interfaces siguiendo el estándar (G.703 ó G.704) que deseemos, según se ha mostrado en el punto anterior (configuraremos, por ejemplo, los interfaces serial 0/0:0 y serial 0/1:0).

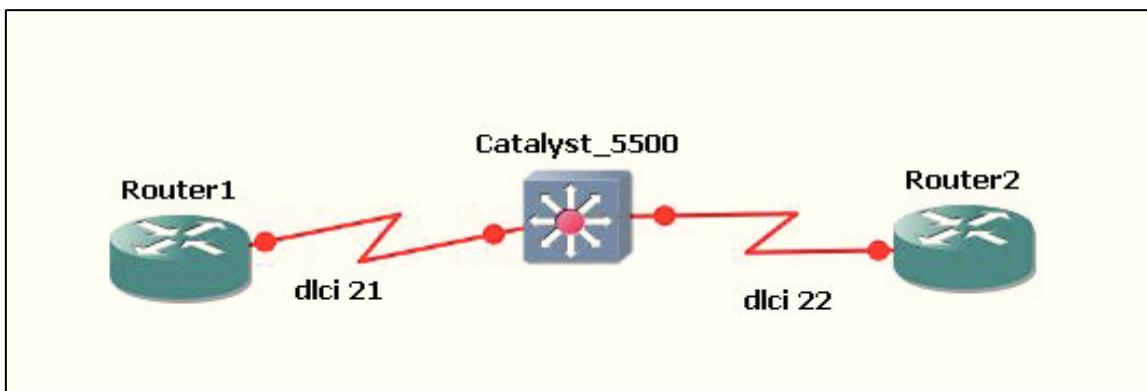


Figura 7.3. Escenario de estudio planteado en el laboratorio para el uso del *Cisco Catalyst 5500* como comutador *WAN Frame Relay*.

Según este escenario de conmutación *Frame Relay*:

Router1 :

Enlace Router 1 → interface E1 de equipo Catalyst 5500 (interfaz serial 0/0:0)
DLCI asignado: 21

Router2 :

Enlace Router 2 → interface E1 de equipo Catalyst 5500 (interfaz serial 0/1:0)
DLCI asignado: 22

¡Importante!. En este escenario, carece de sentido hablar de la configuración de los interfaces E1 del *Cisco Catalyst* como *DTE* o *DCE*. Ambos interfaces presentarán configuración *NNI*.

A continuación, se establecen los PVC (circuitos virtuales), para la conmutación *WAN Frame Relay* del tráfico, y así intercomunicar los dos *router*:

- RSM4 (config)#frame relay switching
, configuramos nuestro módulo RSM4 con capacidad de conmutación *Frame Relay*.
- RSM4 (config)# interface serial 0/0:0
- RSM4 (config-if)#frame-relay intf-type nni
- RSM4 (config-if)#frame-relay route 21 interface serial 0/1:0 22
, según esta configuración, el *switch* conmuta el tráfico que venga por el interfaz serial 0/0:0 con DLCI 21, al interfaz serial 0/1:0 con DLCI 22 .
- RSM4 (config-if)#exit

Establecemos la configuración “espejo” de la anterior para el interfaz serial 0/1:0:

- RSM4 (config)# interface serial 0/1:0
- RSM4 (config-if)#frame-relay intf-type nni
- RSM4 (config-if)# frame-relay route 22 interface serial 0/0:0 21
- RSM4 (config-if)#exit
- RSM4 (config) #

Con objeto de visualizar y comprobar el resultado de las configuraciones anteriores, disponemos de los siguientes comandos (la variedad de comandos *Cisco IOS* es amplia, pero estos a priori son los fundamentales), ejecutados desde el modo *EXEC privilegiado* :

- RSM4# show frame-relay pvc
, muestra información sobre los PVC configurados (circuitos virtuales para la commutación), y su estado .
- RSM4# show frame-relay route
, muestra información sobre rutas establecidas/asociación entre interfaces y DLCI's.

Para acceder al resto de opciones de visualización/verificación del entorno de trabajo *Frame Relay*, basta introducir el comando:

- RSM4# show frame-relay ?
, mostrándose por pantalla todas las opciones disponibles.

7.2. Anexo 2: Configuraciones de red para arquitecturas emuladas en *GNS3/Dynamips*. *Interoperación entorno real ↔ entorno emulado*

7.2.1. Configuración y verificación de la arquitectura de red WAN ppp.

Se recuerda la *figura 3.1* del capítulo 3, por claridad en la exposición, que muestra la arquitectura de red WAN ppp planteada para los análisis de los capítulos 3 y 4.

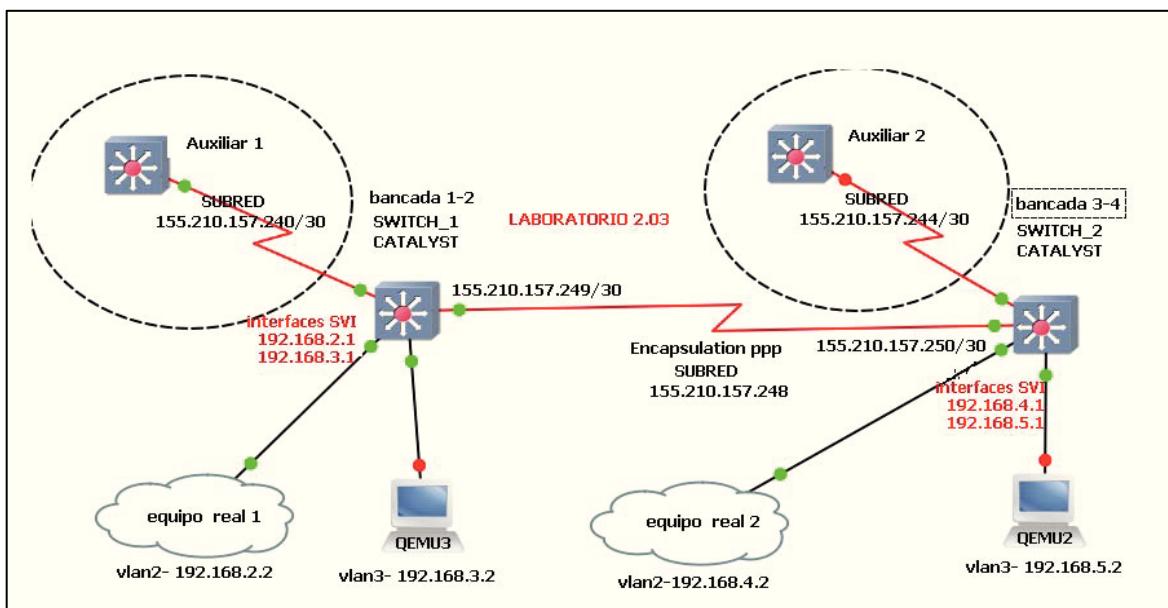


Figura 3.1. Topología/simulación gráfica en *GNS3* de la arquitectura de red WAN ppp.

Disposiciones iniciales para configurar gráficamente la topología en GNS3

- Se seleccionan del menú lateral izquierdo (“*Tipos de nodos*”) e incorporan al panel gráfico principal de diseño los nodos *EtherSwitch router*. Seguidamente, se añaden del menú lateral izquierdo del interfaz gráfico 2 “*nubes*” (reflejadas en la *figura 3.1*), interfaces de red real→virtual que servirán para integrar los equipos reales en el entorno emulado, según el anexo 3. También añadimos 2 terminales virtuales/*host* QEMU, terminales que emulan *host* con sistema operativo Linux.
- Tras añadir los módulos de interfaces adecuados, como se verá seguidamente, se realizará la conexión mediante enlaces *serie ppp* (disponible en la barra del menú principal) de los nodos *EtherSwitch router* anteriormente incorporados y que representan los equipos de red *Cisco Catalyst 5500 Switch* del entorno de red real. Se realiza también la conexión de las “*nubes*” y de los terminales QEMU a los nodos principales mediante enlaces *FastEthernet* disponibles en la barra de

herramientas principal del menú de GNS3. Los equipos reales, integrados en el escenario mediante enlaces *FastEthernet* se adscriben administrativamente y localmente a VLAN2, con direcciones IP y máscaras asignadas según el diseño particular de la *figura 3.1*. Los equipos QEMU son terminales/*host* virtuales Linux y se consideran a efectos de diseño adscritos localmente a VLAN3, en el espacio de direcciones de red indicado.

Sobre dicha arquitectura procedemos a implementar las configuraciones de red, en base a comandos *Cisco IOS*, que permitirán abordar el análisis de la misma. Posteriormente, se procede a verificar la funcionalidad de las configuraciones establecidas. Conforme a la arquitectura de red planteada, procedemos a implementar en el entorno emulado de *GNS3/Dynamips* las configuraciones sobre los dispositivos de red implicados, utilizando para ello los comandos *Cisco IOS* oportunos, y teniendo presentes las particularidades previamente descritas. Recordemos que dicha configuración de red virtual es idéntica (funcionalmente equivalente) e imagen de la que se implementaría en un entorno de red real del laboratorio.

Para configurar un adecuado marco funcional de trabajo, es preciso incorporar en los nodos *EtherSwitch router* del entorno emulado los módulos de interfaces que nos permitan así implementar dispositivos de red virtuales similares a los reales. A través de los interfaces aportados por dichos módulos, se interconectan los equipos implicados en el diseño. Finalmente, se procede a la configuración de red asociada a la arquitectura, mediante comandos *Cisco IOS*. Para ello, primeramente, se muestran las *figuras 7.4* y *7.5*.

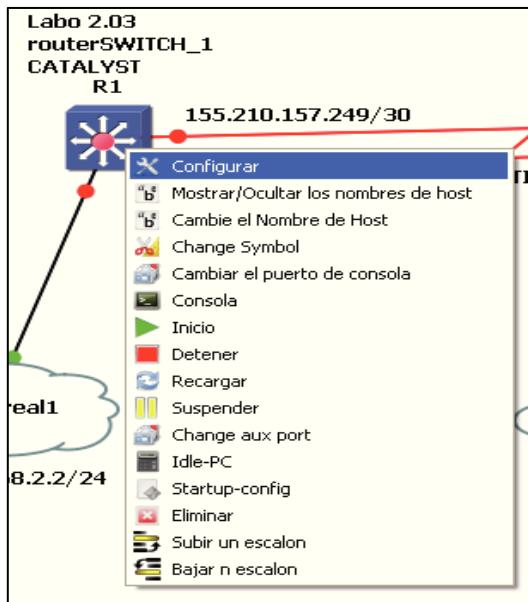


Figura 7.4. Contexto gráfico de opciones en los nodos *EtherSwitch router*.

En el *menú desplegable*, emergente al hacer “*clic*” con el botón derecho del ratón sobre el nodo *EtherSwitch router*, seleccionamos la opción *Configurar*. Esto nos dirige al contexto gráfico de configuración mostrado en la *figura 7.5* de la página siguiente.

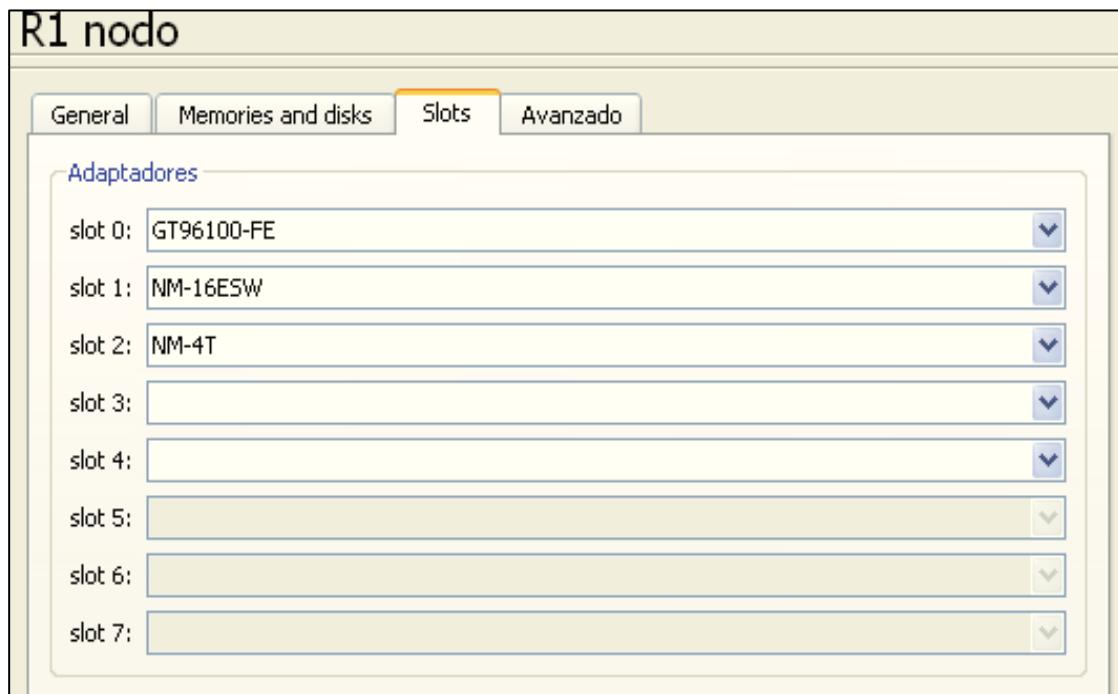


Figura 7.5. Contexto gráfico para incorporar /configurar los módulos de interfaces en *GNS3/Dynamips*.

→ A continuación: **configurar nodo** → **SLOTS** → **SLOT 1** → **NM-16ESW**
SLOT 2 → **NM-4T**

¿Cuáles son la implicación y prestaciones de los módulos de interfaces (virtuales y reales) en el desarrollo del análisis?

Ahondando en lo expuesto en el apartado 2.4, los dispositivos *Cisco Systems* implicados en el estudio son arquitecturas hardware modulares, esto es, permiten incorporar en su estructura hardware diferentes módulos *plug-in* (de interfaces), módulos disponibles en el escenario tecnológico actual, y que incrementan la funcionalidad y versatilidad del entorno de red de trabajo. En nuestro caso, *NM-16 ESW* es un módulo hardware con 16 interfaces *FastEthernet* que nos permite configurar escenarios *VLAN*, así como realizar *InterVLAN routing* (comutación a nivel 3) en *switch multilayer*. Además, *NM-4T* es un módulo hardware que incorpora 4 interfaces *serie*, permitiendo así implementar enlaces *serie* en el entorno de emulación de *GNS3/Dynamips*. Este *add-on* modular del hardware de red (virtual y real) permanecerá, y va a ser común, en el resto de arquitecturas planteadas en el proyecto. Con ello, conformamos un entorno emulado de trabajo idéntico (o muy similar) y funcionalmente equivalente al entorno hardware real conformado por dispositivos *Cisco Catalyst 5500 Switch* del laboratorio.

Para que se produzca intercambio de la información de *routing* entre ambos dispositivos *EtherSwitch router*, con el objetivo de informarse mutuamente de sus respectivas “informaciones de red”, y con esta información configurar cada uno sus tablas de enrutamiento, se activará posteriormente el protocolo RIP v2 en los interfaces pertinentes (en sistemas basados en *Cisco IOS*, los protocolos de enrutamiento están desactivados por defecto; se configuran y activan a nivel de interfaz, sobre los interfaces adecuados según el propósito del diseño)[8]. Siendo así, y después de *a)* iniciar los equipos de la arquitectura de red virtual, y *b)* “abrir” las respectivas *consolas* para acceder al entorno de configuración *Cisco IOS*, según la figura 7.6,

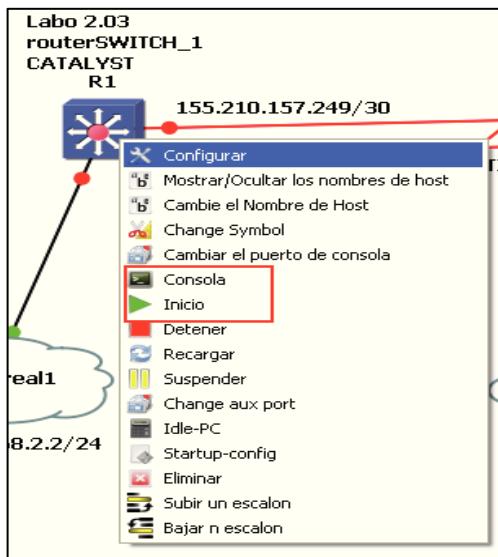


Figura 7.6. Inicio y acceso a consola en dispositivos emulados en *GNS3/Dynamips*.

, se procede seguidamente con las configuraciones según el mapa de direcciones IP y máscaras de subred asignadas para el diseño, y que muestra la figura 3.1. Según esta arquitectura de red, la interconexión de los equipos *EtherSwitch router* que en el entorno real conectarían las *bancadas 1-2* y *3-4* del laboratorio 2.03 se realiza mediante un enlace *WAN serie point-to-point*, adscrito administrativamente a la subred **155.210.157.248 / 30**, subred de **155.210.0.0 / 16**. Los dispositivos de red *EtherSwitch router* son hardware de red (*switch multilayer*) con capacidad para operar en los niveles 2 y 3: además de realizar commutación a nivel 2 entre equipos locales pertenecientes a una misma VLAN, permiten realizar *InterVLAN routing* (comunicación a nivel 3) localmente entre VLAN diferentes mediante la configuración de interfaces virtuales SVI (sin necesidad de utilizar para ello un *router* externo y un enlace troncal configurado con el protocolo 802.1q ó ISL). Para habilitar estas funcionalidades a nivel 3 en los *switch multilayer* no debemos olvidar durante el proceso de configuración ejecutar el siguiente comando *Cisco IOS*, desde el modo de configuración global :

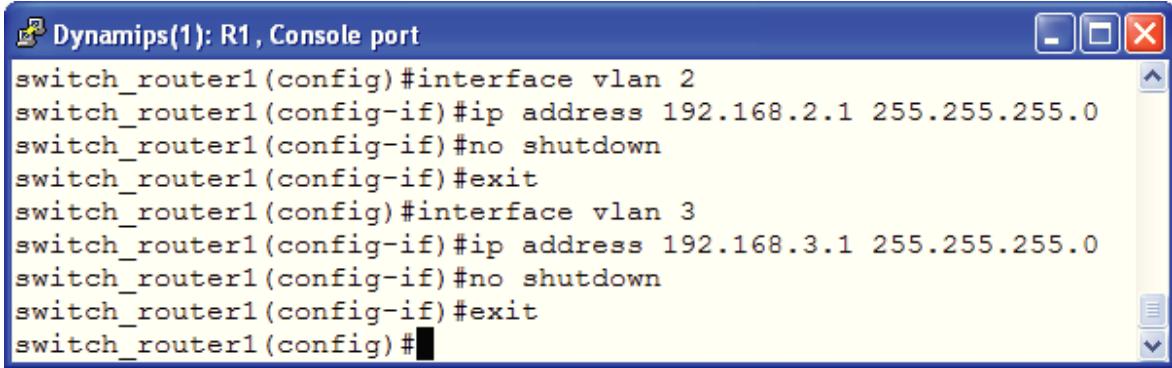
```
switch_router1(config)# ip routing
```

A tal efecto, para establecer las configuraciones de red *vía* consola de los dispositivos mediante comandos Cisco *IOS*, y tomando como base el estudio reflejado en el anexo 1, se procede a configurar, en primer lugar y para el correcto desarrollo del capítulo 3, la arquitectura de red principal (subred **155.210.157.248 /30**, nodos *Etherswitch router 1* y 2).

Configuración VLAN (VLAN2 y VLAN3) en nodo emulado *EtherSwitch router 1*

Atendiendo al mapa de direcciones IP y máscaras de subred establecido para nuestro diseño mostrado en la *figura 3.1*:

**VLAN2: 192.168.2.0 / 24
VLAN3: 192.168.3.0 / 24**



```

Dynamips(1): R1, Console port
switch_router1(config)#interface vlan 2
switch_router1(config-if)#ip address 192.168.2.1 255.255.255.0
switch_router1(config-if)#no shutdown
switch_router1(config-if)#exit
switch_router1(config)#interface vlan 3
switch_router1(config-if)#ip address 192.168.3.1 255.255.255.0
switch_router1(config-if)#no shutdown
switch_router1(config-if)#exit
switch_router1(config)#

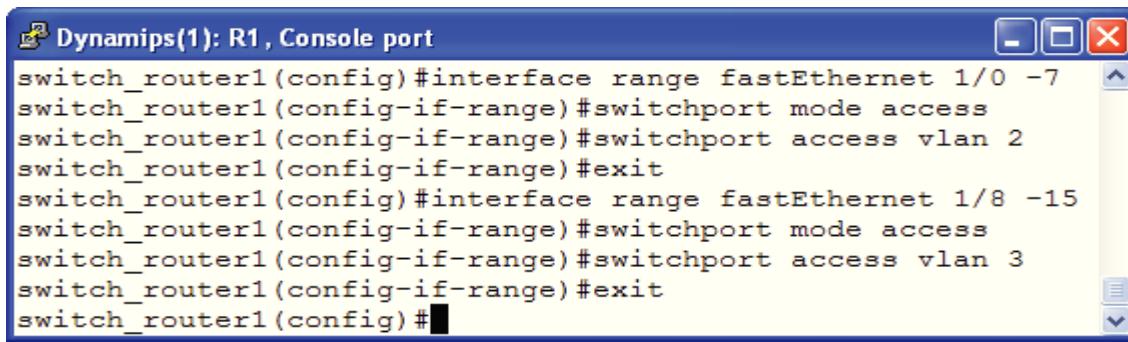
```

Figura 7.7. Creación y configuración de interfaces virtuales SVI para VLAN2 y VLAN3 en el nodo emulado *EtherSwitch router 1*.

Desde el modo de *configuración global*, como refleja la *figura 7.7*, se crean las VLAN e interfaces virtuales SVI [4], y se les asigna una dirección IP y máscara de subred, según el mapa de direcciones de red de la *figura 3.1* anterior.

La configuración del interfaz virtual SVI en un dispositivo de red con capacidad para realizar conmutación a nivel 3 es absolutamente necesaria, ya que actuará como *default gateway* para poder implementar *InterVLAN routing* entre equipos que pertenecen a VLAN diferentes. Para efectuar la conmutación entre equipos locales en un nodo y que pertenecen a una misma VLAN, conmutación a nivel 2 según la tabla de direcciones MAC “aprendida” y conformada en el dispositivo *EtherSwitch router*, dichos interfaces virtuales SVI no son necesarios. A partir de los 16 interfaces físicos *FastEthernet* disponibles en el *slot 1* de nuestro dispositivo de red gracias al módulo *NM-16SW* previamente incorporado, asignamos en nuestro diseño particular, y conforme muestra la *figura 7.8 a)* de la página siguiente:

- Interfaces **0 - 7 → VLAN2**
- Interfaces **8 - 15 → VLAN3**



```
switch_router1(config)#interface range fastEthernet 1/0 -7
switch_router1(config-if-range)#switchport mode access
switch_router1(config-if-range)#switchport access vlan 2
switch_router1(config-if-range)#exit
switch_router1(config)#interface range fastEthernet 1/8 -15
switch_router1(config-if-range)#switchport mode access
switch_router1(config-if-range)#switchport access vlan 3
switch_router1(config-if-range)#exit
switch_router1(config)#[
```

Figura 7.8 a). Configuración VLAN: configuración y asignación VLAN de interfaces en nodos emulados *EtherSwitch router*.

Con ello, quedan configuradas las facilidades para realizar conmutación a nivel 2 entre equipos pertenecientes a una misma VLAN, y conmutación a nivel 3 entre VLAN diferentes (VLAN2 y VLAN3, en este caso); todo ello, y en este punto del análisis, en el ámbito VLAN de un mismo nodo *EtherSwitch Router*.

En este punto es reseñable, de cara a establecer una comparativa entre los entornos de red real (laboratorio) y emulado, que la misma asignación de interfaces realizada en la figura 7.8 a) anterior se implementaría en el equipo real *Cisco Catalyst 5500 Switch* por medio de comandos *Cisco CatOS* [3][7], conforme a las particularidades descritas en el apartado 2.4 y siguiendo el anexo 1, obteniendo funcionalmente el mismo resultado.

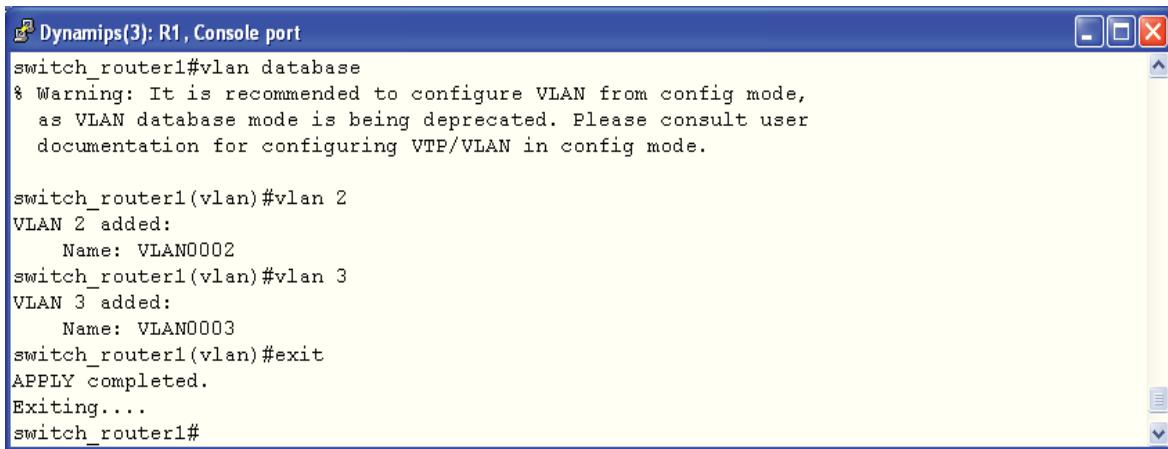
Configuración VLAN complementaria: configuraciones en modo *VLAN Database*.

Se da la circunstancia de que determinados sistemas *Cisco IOS* (es realmente muy difícil concretar cuáles por la gran cantidad de sistemas y versiones existentes de dispositivos *Cisco Systems* en el mercado, pero suelen ser sistemas relativamente antiguos) precisan de una configuración adicional a la hora de implementar correctamente la configuración VLAN. Esto se ejecuta en un submenú o *modo de configuración* denominado *VLAN Database*.

Los sistemas *Cisco IOS* antiguos precisaban de dicho *modo de configuración* para establecer toda la configuración VLAN. En sistemas *Cisco IOS* actuales, toda la configuración VLAN se establece desde el *modo de operación* o *configuración global*. Pero en esta transición entre ambas formas de operar existen sistemas que precisan de una cierta configuración adicional. A grandes rasgos, es una configuración suplementaria para “activar” la configuración VLAN previamente establecida desde el *modo de configuración global*. Es decir, para que la configuración VLAN se implemente correctamente, debemos acceder a dicho submenú y proceder a la activación. Este es un sencillo proceso, pero presenta el inconveniente funcional de que no se conserva en la memoria del dispositivo y

se borra cada vez que se apaga o reinicia el dispositivo (no como el resto de las configuraciones en *Cisco IOS*, si procedemos a guardarlas en memoria). Se han encontrado posibles respuestas en foros de Internet para solucionar este inconveniente y poder conservar en memoria la “activación”, pero ninguna ha resultado ser realmente efectiva.

En sistemas *Cisco IOS* actuales, dicha configuración *VLAN Database* ya no es precisa y es suficiente con el proceso de configuración VLAN general del punto anterior. El acceso a dicho submenú o *modo de configuración VLAN Database* también es necesario, si el sistema es relativamente antiguo, para configurar las funcionalidades del protocolo VTP que se ven en el siguiente apartado 7.2.6. Para acceder a dicho submenú *VLAN Database* y proceder a la “activación” VLAN que completa el proceso de configuración anterior, se accede desde el *modo de operación EXEC privilegiado*, procediendo como indica a continuación la figura 7.8 b), en base a comandos *Cisco IOS*. Se observa un “warning” del sistema recomendando usar el *modo de configuración global* para acometer tareas de configuración VLAN, acorde a las indicaciones antes descritas acerca de la relativa antigüedad del *modo de operación VLAN Database*, pero se comprueba empíricamente que si no operamos de esta forma los interfaces virtuales SVI no se activan convenientemente. Resumiendo, para cada sistema *Cisco IOS* deberá comprobarse la necesidad de proceder con dicha activación adicional.



```

Dynamips(3): R1, Console port
switch_router1#vlan database
% Warning: It is recommended to configure VLAN from config mode,
           as VLAN database mode is being deprecated. Please consult user
           documentation for configuring VTP/VLAN in config mode.

switch_router1(vlan)#vlan 2
VLAN 2 added:
  Name: VLAN0002
switch_router1(vlan)#vlan 3
VLAN 3 added:
  Name: VLAN0003
switch_router1(vlan)#exit
APPLY completed.
Exiting....
switch_router1#

```

Figura 7.8 b). Configuración VLAN: activación de interfaces SVI desde *VLAN Database*.

Verificación de la conectividad VLAN en el emulador GNS3/Dynamips.

Configuramos primero *equipo_real 1* (integrado éste en el entorno emulado según indica el anexo 3), asignando dirección IP **192.168.2.2 / 24** al interfaz de red *Ethernet* de *equipo_real 1* (PC portátil real y externo al emulador), según refleja la *figura 7.9*.

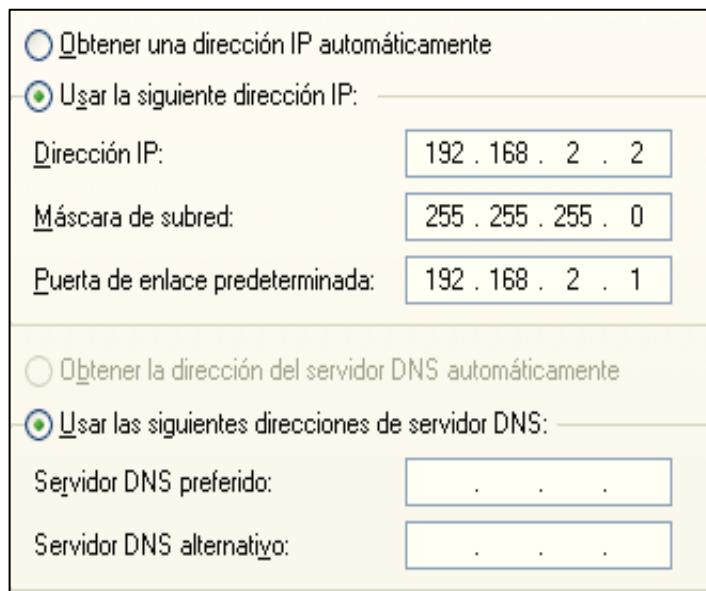


Figura 7.9. Configuración de interfaz *Ethernet* en *equipo_real 1*.

Del mismo modo, se configura *QEMU3*, asignando la dirección IP **192.168.3.2 / 24** al interfaz *Ethernet eth0* del *host* emulado Linux *QEMU3*, además de una puerta de enlace predeterminada través de dicho interfaz *eth0*:

```
tc@QEMU3:~$ sudo ifconfig eth0 192.168.3.2
tc@QEMU3:~$ sudo route add default gw 192.168.3.1 eth0
```

Resulta interesante en este punto del estudio ver que en la configuración del interfaz de red de *equipo_real 1* (*figura 7.9*) estamos considerando como *default gateway* para el mismo el interfaz virtual SVI de VLAN2, previamente configurado en el entorno emulado, procediendo así a la potencial interoperación entre equipos reales y virtualmente emulados. A continuación, mediante el comando Linux **traceroute**, y desde QEMU3 (**192.168.3.2 /24**, *host* virtual Linux adscrito a VLAN3), se comprueba la conectividad VLAN, así como el trayecto seguido por el tráfico de red implicado, según la *figura 7.10*.

```
tc@QEMU3:~$ traceroute 192.168.2.2
traceroute to 192.168.2.2 (192.168.2.2), 30 hops max, 38 byte packets
1 192.168.3.1 (192.168.3.1) 35.312 ms 23.111 ms 0.117 ms
2 192.168.2.2 (192.168.2.2) 22.160 ms 6.133 ms 6.745 ms
tc@QEMU3:~$
```

Figura 7.10. *Test de conectividad y trayectoria del tráfico en la conmutación VLAN a nivel 3.*

, observando como primero (*hop 1*) se “alcanza” el SVI de VLAN3, y desde allí (*hop 2*) se conmuta el tráfico a nivel 3 hacia el *host* de destino con dirección IP **192.168.2.2 /24** (*equipo portátil real 1 externo*, adscrito a VLAN2).

Es destacable que esta misma verificación en la conectividad VLAN permitiría establecer una primera conclusión en el estudio de la hasta ahora potencial capacidad que ofrecía el entorno de emulación *GNS3/Dynamips* para *interoperar* tráfico de red real y emulado: la comprobación de la anterior conectividad VLAN, se ha realizado mediante una correcta interoperación entre tráfico/equipos reales y emulados.

Configuración VLAN (VLAN2 y VLAN3) en nodo emulado EtherSwitch router 2

De modo similar a la configuración expuesta anteriormente, se procede para configurar VLAN2 y VLAN3 en el nodo *EtherSwitch router 2*, salvedad hecha con el espacio de direcciones de red asignado, siendo ahora:

VLAN2 : 192.168.4.0 / 24
VLAN3 : 192.168.5.0 / 24

Es conveniente clarificar aclarar en este caso que para la arquitectura actual de estudio (*WAN ppp*), el espacio de direcciones IP a configurar para una misma VLAN administrativa según nuestros intereses de trabajo, pero en entornos locales diferentes, no es el mismo (conexión VLAN a nivel 3).

¿Por qué?

La causa reside, como se verá en detalle más adelante, en que si esto no fuese así, al intercambiar los respectivos nodos *EtherSwitch router* la información de sus tablas de enrutamiento, éstas no se actualizarían convenientemente, pues los *router* creerían estar actualizando información que ya disponen en su tabla, por tener idéntica dirección IP y máscara.

Pero esta información es distinta de la que ya tienen, y se encuentra en un dominio *broadcast* diferente. Por lo tanto, dicha información es “aprendida” por los *router* a través del protocolo de enrutamiento dinámico correspondiente (RIP v2, en nuestro estudio) y con distancia administrativa mayor (distancia administrativa de RIP = 120) [8]. El dispositivo entendería que está actualizando una información que ya tiene en su tabla con distancia administrativa menor (la información que dispone de la misma VLAN “directamente conectada” tiene distancia administrativa = 0), por lo que no incorporaría la información nueva en sus tablas, discriminando así la misma sobre equipos y dispositivos adscritos a la misma VLAN administrativa, pero que están en otro dominio *broadcast*.

En el escenario VLAN commutado que se abordará seguidamente en este anexo, se configurará para las VLAN la misma dirección IP y máscara de red, aunque éstas se encuentren físicamente en nodos de red diferentes.

Conexión de los nodos EtherSwitch router vía enlace serie WAN point-to-point

Para configurar el enlace *serie WAN ppp* entre los nodos referidos, se deben asignar direcciones de red IP a los interfaces *serie* implicados (se configura el interfaz **serial 2/0** en ambos nodos, para facilitar el análisis), según el mapa de direcciones de red establecido en el diseño de la *figura 3.1*.

Al respecto:

- enlace serie *WAN ppp* ↔ subred ***155.210.157.248 /30***
- dirección IP del interfaz *serial 2/0* en el nodo 1 → ***155.210.157.249 /30***
- dirección IP del interfaz *serial 2/0* en el nodo 2: ***155.210.157.250 /30***.

Es necesario recalcar, con objeto de clarificar el mapa de direcciones IP de nuestro escenario de trabajo, que sobre la dirección IP de la *red mayor* del enlace serie (***155.210.0.0/16***) se ha realizado *subnetting* [1], *con* máscara de subred ***/30*** (equivalente a la máscara ***255.255.255.252***), para un óptimo aprovechamiento de las direcciones IP disponibles en un escenario telemático real. Este diseño sería aplicable, por ejemplo, en un entorno de trabajo que precise implementar numerosas conexiones/subredes independientes, con muy pocos nodos implicados en su infraestructura WAN de interconexión, enlaces *serie* fundamentalmente: dicho mapa de direcciones sólo permite dos nodos por conexión/subred WAN, en principio [1]. Adicionalmente, sobre cada subred, se procedería a implementar sus respectivos entornos de trabajo LAN/VLAN.

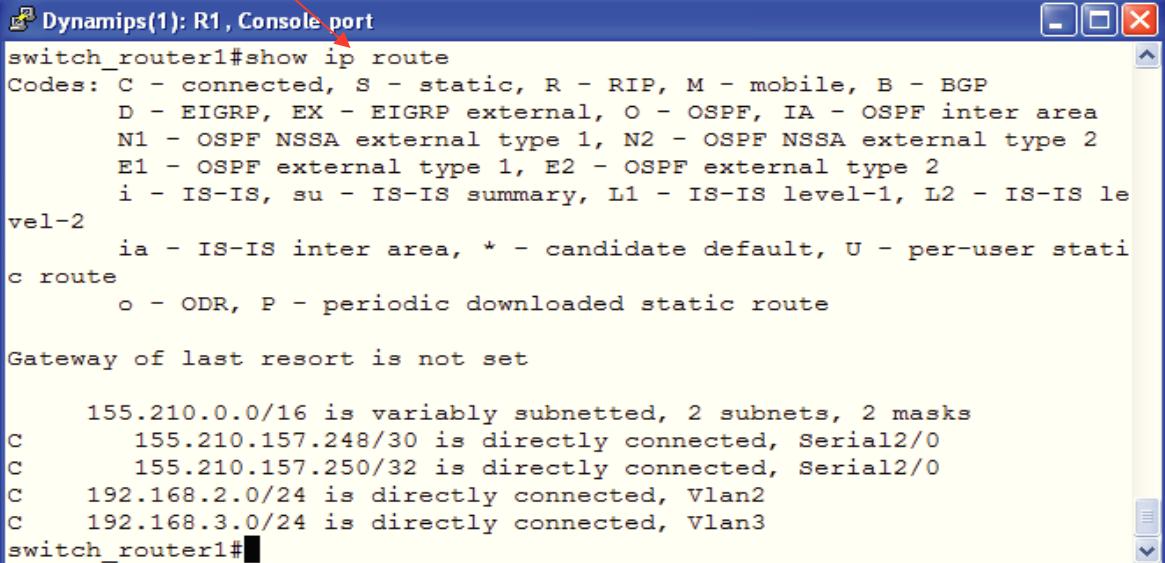
Seguidamente, procedemos a activar el protocolo a configurar para el enlace serie, en este caso *ppp* (*point-to-point protocol*). Todo este proceso, con los comandos *Cisco IOS* ejecutados, se ilustra de modo explícito a continuación en la figura 7.11.

```
Dynamips(0): R1, Console port
switch_router1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch_router1(config)#interface serial 2/0
switch_router1(config-if)#ip address 155.210.157.249 255.255.255.252
switch_router1(config-if)#no shutdown
switch_router1(config-if)#encapsulation ppp
switch_router1(config-if)#exit
switch_router1(config)##
```

Figura 7.11. Configuración del enlace serie WAN ppp en nodo *EtherSwitch router 1*.

De modo análogo se procedería para configurar el interfaz *serial 2/0* en el nodo *EtherSwitch router 2*. A la hora de configurar cualquier interfaz no debe olvidarse en dispositivos *Cisco Systems* el comando destinado a “activar” el mismo, pues no basta solo con asignarle una dirección de red para que se muestre activo (en sistemas basados en *Cisco IOS*, por defecto, los interfaces se encuentran desactivados).

En este punto, se realiza una primera consulta de las tablas de enrutamiento que los nodos emulados *EtherSwitch router* tienen configuradas hasta el momento, con el ingreso del comando *Cisco IOS* definido para ello y desde el modo *EXEC privilegiado*, según la figura 7.12 , y el anexo 1.



```
Dynamips(1): R1, Console port
switch_router1#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
      c route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      155.210.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        155.210.157.248/30 is directly connected, Serial2/0
C        155.210.157.250/32 is directly connected, Serial2/0
C        192.168.2.0/24 is directly connected, Vlan2
C        192.168.3.0/24 is directly connected, Vlan3
switch_router1#
```

Figura 7.12. Tabla de enrutamiento (situación inicial) en el nodo *EtherSwitch router 1*.

Según refleja la tabla de enrutamiento anterior, el nodo *EtherSwitch router 1* no tiene constancia de los dispositivos conectados al nodo *EtherSwitch router 2* (y viceversa).

¿Cuál es el motivo?

En este punto de la configuración de red no se ha producido intercambio de información a nivel 3 entre dispositivos porque no se ha configurado para ello un protocolo (dinámico) de enrutamiento en los mismos, o bien como alternativa a esto, el administrador de red no ha procedido a la configuración de rutas estáticas que permitan la comunicación entre los equipos y dispositivos implicados en el escenario de trabajo. Conforme a esto, se procede a configurar un protocolo de enrutamiento dinámico en el entorno emulado.

Configuración del protocolo de enrutamiento RIP v2 en el entorno emulado.

Se procede a la configuración y activación del protocolo de enrutamiento dinámico RIP v2 en el entorno emulado de trabajo. Para ello, tal como se mencionó anteriormente, en sistemas *Cisco IOS* los protocolos de enrutamiento deben activarse a nivel de interfaz (*modo de configuración global → interfaz* , anexo 1), estando por defecto desactivados.

Con esta “activación”, se habilita el envío y recepción de información y actualizaciones de las tablas de enrutamiento en un dispositivo, a través del interfaz activado.

```

Dynamips(1): R1 , Console port
switch_router1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch_router1(config)#router rip
switch_router1(config-router)#version 2
switch_router1(config-router)#network 155.210.0.0
switch_router1(config-router)#network 192.168.3.0
switch_router1(config-router)#network 192.168.2.0
switch_router1(config-router)#exit
switch_router1(config)#

```

Figura 7.13. Configuración del protocolo RIP v2 en el nodo *EtherSwitch router* emulado.

Según describe la figura 7.13 anterior, se activa el protocolo RIP v2 en aquellos interfaces que deseemos presenten implicación completa a nivel 3 en el entorno de red de trabajo (es decir, que envíen actualizaciones de su tabla de enrutamiento y también reciban actualizaciones de las tablas de otros dispositivos implicados en el escenario). Como se observa, en sistemas *Cisco IOS*, dicha activación se realiza “informando” de la dirección IP de la red mayor a la que pertenece el interfaz a activar. Como RIP v2 es un protocolo *classless* [8] (a diferencia de RIP que es *classfull* y no “anuncia” las máscaras de sus tablas), el protocolo se encargará de informar adecuadamente de la mascara de red asociada a las subredes implicadas en la comunicación. Por ello, el uso de RIP v2 es más adecuado que RIP en este escenario de trabajo, ya que hemos realizado *subnetting* para conectar los nodos *EtherSwitch router* por medio del enlace *serie WAN ppp* (recordemos, subred **155.210.157.248 /30**), evitando así el riesgo de que la información relativa a subredes (de la red mayor **155.210.0.0 /16**) no se difunda y actualice correctamente.

Al respecto, conviene apuntar que no existe inconveniente en haber usado RIP en el diseño, pero en este caso debería tenerse especial atención en desactivar el comportamiento *classfull* del protocolo, desde el *modo de configuración global → protocolo*, por medio del comando *Cisco IOS*:

```
switch_router1(config-router)#ip classless
```

, evitando así los riesgos de pérdida de información anteriormente descritos [8].

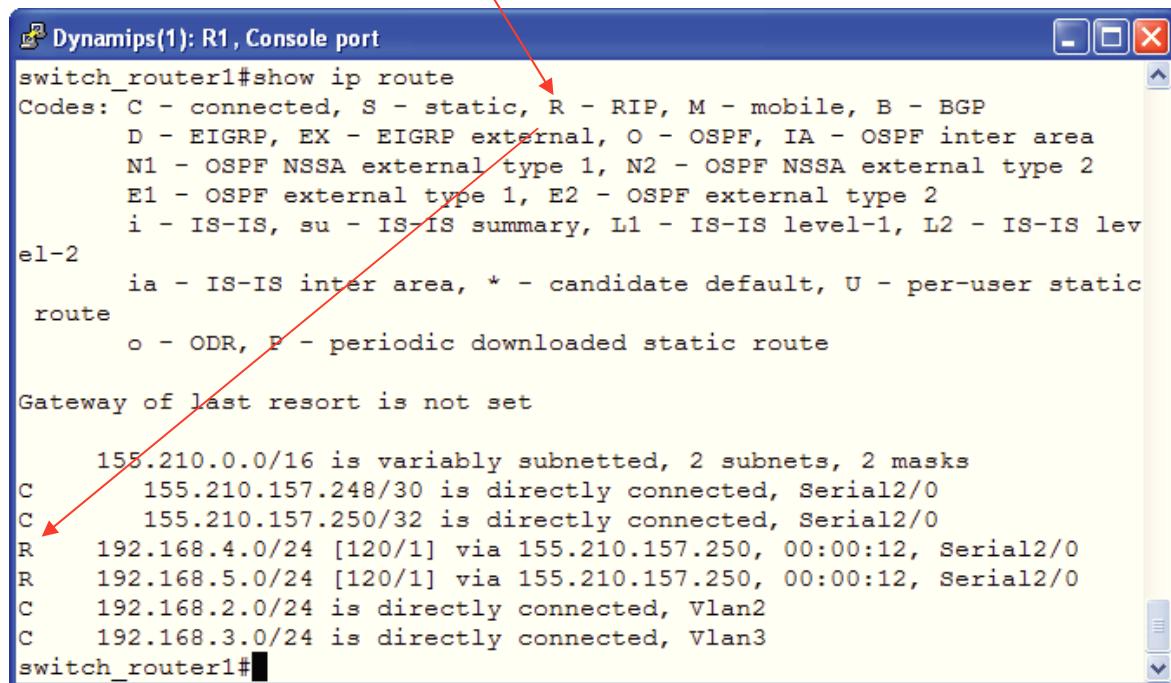
Llegados a este punto del análisis, configurado ya todo el entrampado de interfaces (*FastEthernet*, *serie* y *SVI*), VLAN y protocolos implicados en la comunicación, se realiza consulta de las tablas de enrutamiento configuradas (y supuestamente aprendidas y

actualizadas a través del protocolo dinámico RIP v2) en los nodos emulados *EtherSwitch router*, obteniendo así información relevante de la conectividad y el tráfico de red que debería circular actualmente por el entorno de red emulado (*figura 7.14*). Así mismo, podemos extraer importantes conclusiones acerca de la comunicación establecida para el intercambio y actualización de la información de *routing* entre nodos virtuales emulados en *GNS3/Dynamips*, y analizar así el rigor funcional del emulador.

Mediante el ingreso desde el *modo de configuración EXEC privilegiado* del comando *Cisco IOS*:

```
switch_router1#show ip route
```

, se observa en la *figura 7.14* la tabla de enrutamiento que una vez activado el protocolo RIP v2 presenta el nodo emulado *Etherswitch router 1*. En la misma se aprecia como el dispositivo aprende vía RIP la información de red que dispone el nodo 2: es decir, se ha establecido comunicación entre ambos nodos virtuales para proceder al intercambio y actualización de sus tablas de enrutamiento. A tal efecto, por el entramado virtual debería circular tráfico de red relacionado con la actualización y mantenimiento de la información asociada al protocolo RIP v2. Esto lo vemos en el *apartado 3.1.2* del capítulo 3 y en el siguiente *apartado 7.2.2*.



```
Dynamips(1): R1, Console port
switch_router1#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 lev
el-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

      155.210.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        155.210.157.248/30 is directly connected, Serial2/0
C        155.210.157.250/32 is directly connected, Serial2/0
R        192.168.4.0/24 [120/1] via 155.210.157.250, 00:00:12, Serial2/0
R        192.168.5.0/24 [120/1] via 155.210.157.250, 00:00:12, Serial2/0
C        192.168.2.0/24 is directly connected, Vlan2
C        192.168.3.0/24 is directly connected, Vlan3
switch_router1#
```

Figura 7.14. Tabla de enrutamiento final después de habilitar RIP v2 en el entorno emulado del análisis.

7.2.2. Análisis exhaustivo del tráfico de red emulado en el enlace serie WAN ppp.

Completando el contenido del *apartado 3.1.2* del correspondiente capítulo 3, para profundizar en el análisis del tráfico de red emulado se contrasta, por ejemplo, la trama nº 19 de la *figura 3.3* (figura que reproducimos nuevamente por claridad de la exposición), trama relacionada con el protocolo de enrutamiento dinámico RIP v2. La *figura 7.15* siguiente muestra esta información.

No.	Time	Source	Destination	Protocol	Length	Info
14	20.515000	N/A	N/A	PPP LCP	16	Echo Reply
15	30.718000	N/A	N/A	PPP LCP	16	Echo Request
16	30.750000	N/A	N/A	PPP LCP	16	Echo Reply
17	30.750000	N/A	N/A	PPP LCP	16	Echo Request
18	30.781000	N/A	N/A	PPP LCP	16	Echo Reply
19	37.328000	155.210.157.250	224.0.0.9	RIPV2	76	Response
20	40.953000	N/A	N/A	PPP LCP	16	Echo Request
21	40.968000	N/A	N/A	PPP LCP	16	Echo Reply
22	41.000000	N/A	N/A	PPP LCP	16	Echo Request
23	41.015000	N/A	N/A	PPP LCP	16	Echo Reply
24	47.312000	155.210.157.249	224.0.0.9	RIPV2	76	Response
25	51.187000	N/A	N/A	PPP LCP	16	Echo Request
26	51.203000	N/A	N/A	PPP LCP	16	Echo Reply
27	51.250000	N/A	N/A	PPP LCP	16	Echo Request
28	51.265000	N/A	N/A	PPP LCP	16	Echo Reply
29	52.156000	N/A	N/A	CDP	333	Device ID: switch_router2 Port ID: serial2/0
30	53.140000	N/A	N/A	CDP	333	Device ID: switch_router1 Port ID: serial2/0
31	61.437000	N/A	N/A	PPP LCP	16	Echo Request
32	61.453000	N/A	N/A	PPP LCP	16	Echo Reply
33	61.468000	N/A	N/A	PPP LCP	16	Echo Request
34	61.500000	N/A	N/A	PPP LCP	16	Echo Reply
35	63.843000	155.210.157.250	224.0.0.9	RIPV2	76	Response
36	71.672000	N/A	N/A	PPP LCP	16	Echo Request

Figura 3.3. Tráfico de red en el enlace serie WAN ppp del entorno emulado.

Frame 19: 76 bytes on wire (608 bits), 76 bytes captured (608 bits)
Point-to-Point Protocol
Internet Protocol version 4, Src: 155.210.157.250 (155.210.157.250), Dst: 224.0.0.9 (224.0.0.9)
User Datagram Protocol, Src Port: router (520), Dst Port: router (520)
Routing Information Protocol
Command: Response (2)
Version: RIPV2 (2)
+ IP Address: 192.168.4.0, Metric: 1
+ IP Address: 192.168.5.0, Metric: 1

Figura 7.15. Trama capturada en el enlace serie WAN ppp con información RIP v2.

Se aprecia como la trama capturada contiene información acerca de la tabla de enrutamiento actual del nodo *EtherSwitch router 2*, y se publica con dirección IP de destino multicast 224.0.0.9, desde el interfaz *serial 2/0*: dirección IP **155.210.157.250 /30**. Esta información permite a dispositivos de red que operan en el nivel 3 del modelo de

referencia OSI, actualizar, si procede, las tablas de enrutamiento con información aprendida dinámicamente vía RIP v2. Así quedará registrado en sus tablas, como muestra la *figura 7.14* de este mismo anexo 2. Siendo así, es manifiesto que el compendio de los análisis del capítulo 3, apoyados en las configuraciones y verificaciones implementadas en este anexo 2, revela una estricta correspondencia funcional del emulador *GNS3/Dynamips* a nivel 2 y 3 del modelo OSI, reproduciendo y emulando el comportamiento que hardware y tráfico de red reales presentarían en un entorno de red real análogo.

A pesar de esta interesante correspondencia funcional entre el entorno hardware real y su correspondiente “imagen” en el entorno emulado de *GNS3/Dynamips*, un análisis detallado en los tiempos de envío de actualizaciones de enrutamiento RIP v2 revela que estos no se ajustan estrictamente a lo que sería esperable teóricamente: RIP v2 envía actualizaciones de las tablas enrutamiento (para su mantenimiento, y si no ha habido cambios) cada 30 segundos; pero un análisis del tráfico capturado muestra desviaciones máximas de hasta 4 segundos entre estas actualizaciones, aproximadamente. Aún así, el rendimiento general y funcionalidad de la arquitectura de red emulada a nivel 2 y 3, y los protocolos implicados, no se ven afectados por ello, pues manejan márgenes de tiempo más elevados en la transferencia de información a nivel 3, no viéndose afectados por la desviación temporal anterior. Pero esta incidencia “temporal” obliga a focalizar el análisis del entorno de emulación sobre el comportamiento de parámetros de red dependientes de la variable “*tiempo*”, como la latencia, que podrían presentar diferencias con respecto al entorno de red real (esto es, un comportamiento en tiempo real del emulador no estricto).

Por lo demás, en la captura del tráfico de la *figura 3.3* anterior se aprecia como el entorno emulado realiza un tratamiento de la información trama a trama, ajustándose con mayor o menor exactitud a un funcionamiento general en tiempo real de la arquitectura de red.

7.2.3. Configuración de Listas de Control de acceso (ACL).

La configuración establecida hasta el momento en el escenario de trabajo emulado permite una accesibilidad total entre equipos y dispositivos pertenecientes a VLAN diferentes. Pero esta situación puede no ser óptima en la práctica, desde un punto de vista de la seguridad. De hecho, en entornos reales profesionales de trabajo no es nada recomendable. Para impedir el acceso no autorizado a una red o servicios prestados en ella, y/o permitir el acceso selectivo de determinados usuarios a equipos y servicios ofrecidos, los dispositivos de red de nivel 3 ofrecen funciones de filtrado mediante el uso de listas de control de acceso (ACL).

Una ACL [1][11] es una lista secuencial de sentencias de permiso o rechazo que se aplican a direcciones o protocolos de capa superior. Estas a su vez se clasifican en ACL *estándar* y *extendidas*, siendo estas últimas más completas: verifican no sólo direcciones de paquetes de origen y destino, sino también los protocolos y números de puerto. Esto ofrece mayor flexibilidad y opciones de filtrado.

En el caso que nos ocupa, como supuesto práctico para evaluar la funcionalidad del emulador, se configuran ACL *estándar* con la finalidad de filtrar direcciones IP de origen y destino, restringiendo así la comunicación entre VLAN diferentes (VLAN2 y VLAN3, en este caso concreto de estudio).

Sin entrar en detalles excesivos para nuestro caso de estudio (para profundizar en el tema puede consultarse la bibliografía oportuna), el principio de funcionamiento de una ACL configurada y aplicada a un interfaz concreto sería: toda vez que una trama accede al interfaz, el router verifica si la dirección de nivel 2 coincide, o si es una trama de *broadcast*. Si la trama es aceptada, la información de la misma se elimina y el *router* busca una ACL en el interfaz de acceso. Si existe una ACL, verifica si el paquete cumple las condiciones de la lista. Si cumple las condiciones, se lleva a cabo la acción de aceptar o rechazar el paquete (*permit* o *deny*). Si se acepta el paquete en el interfaz, se compara con las entradas de la tabla de enrutamiento para determinar el interfaz destino y comutarlo a aquel interfaz. Seguidamente, el *router* verifica si el interfaz destino tiene configurada una ACL. Si esto es así, compara el paquete con las sentencias de la lista y si el paquete concuerda con una sentencia, se lleva a cabo la aceptación o rechazo del paquete. Si no hay ACL (o se acepta el paquete), el paquete se encapsula en el nuevo protocolo de nivel 2 y se envía por el interfaz adecuado para alcanzar el dispositivo siguiente [11].

Muy importante: toda ACL termina con una sentencia implícita: *deny any* (*denegación a cualquiera*), que no permitirá que ningún paquete que no coincida con alguna sentencia (*permit / deny*) de la ACL sea aceptada. Este proceso secuencial de funcionamiento se ilustra a continuación, en la *figura 7.16*.

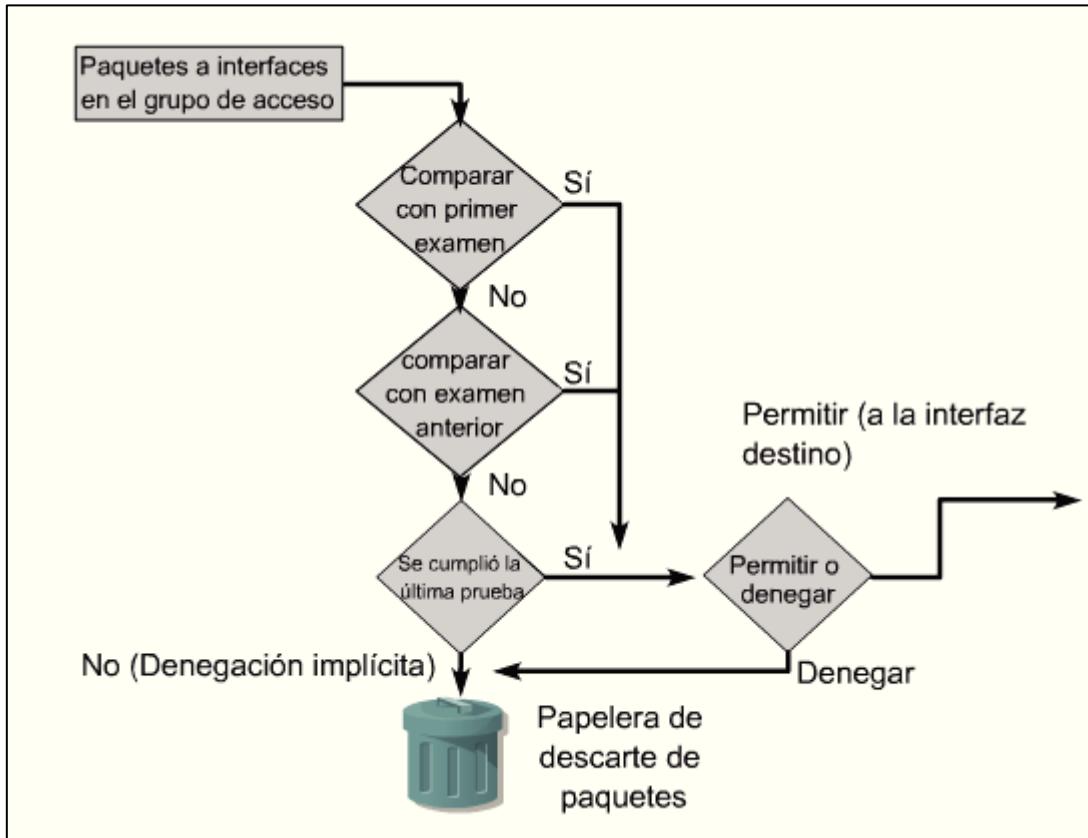


Figura 7.16. Esquema de funcionamiento general de una ACL.
Imagen extraída de “Programa de la Academia de Networking CISCO CCNA2 v3.1”[11]

En base a los fundamentos teóricos anteriores, la configuración de las pertinentes ACL en el escenario de estudio con objeto de que equipos y dispositivos de VLAN2 no puedan acceder a servicios y dispositivos de VLAN3, y viceversa, quedaría conformada como se indica a continuación mediante los oportunos comandos *Cisco IOS* introducidos secuencialmente (por no ser objetivo concreto del análisis, no se profundiza en la configuración de las ACL, y se realiza una configuración básica de las mismas desde un punto de vista funcional)

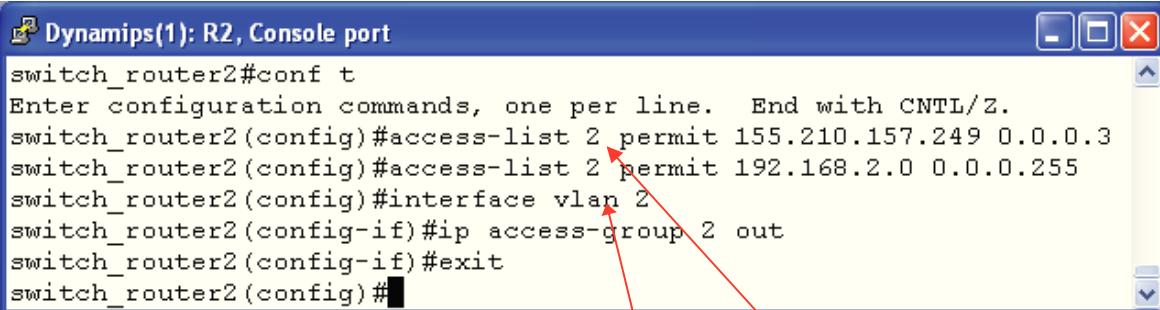
- **Nodo EtherSwitch router 1 (figura 7.17)**

```

Dynamips(0): R1, Console port
switch_router1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch_router1(config)#acce
switch_router1(config)#access-list 2 permit 155.210.157.250 0.0.0.3
switch_router1(config)#access-list 2 permit 192.168.4.0 0.0.0.255
switch_router1(config)#interface vlan 2
switch_router1(config-if)#ip access-group 2 out
switch_router1(config-if)#exit
switch_router1(config)#
  
```

Figura 7.17. Configuración de ACL en nodo emulado EtherSwitch router 1.

- **Nodo EtherSwitch router 2 (figura 7.18)**



```

Dynamips(1): R2, Console port
switch_router2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch_router2(config)#access-list 2 permit 155.210.157.249 0.0.0.3
switch_router2(config)#access-list 2 permit 192.168.2.0 0.0.0.255
switch_router2(config)#interface vlan 2
switch_router2(config-if)#ip access-group 2 out
switch_router2(config-if)#exit
switch_router2(config)#

```

Figura 7.18. Configuración de ACL en nodo emulado *EtherSwitch router 2*.

Completando la temática sobre la configuración de ACL en un escenario telemático, y como aclaración a los comandos *Cisco IOS* de configuración introducidos, se clarifican los siguientes aspectos:

- Toda ACL se configura con un número; en nuestro caso, número 2 (ACL *numeradas*; aunque también hay ACL *nombradas*, no se entra en más detalle por no ser necesario). Esta numeración se rige según indica la figura 7.19.
- Una vez configurada una ACL, ésta debe aplicarse sobre un interfaz concreto (en el caso de estudio, el interfaz de aplicación es el SVI de la VLAN) y con una dirección de aplicación de entrada (*in*) o de salida (*out*), como refleja la figura 7.18.

Protocolo	Intervalo
IP	1-99, 1300-1999
IP extendido	100-199, 2000-2699
AppleTalk	600-699
IPX	800-899
IPX extendido	900-999
Protocolo de publicación de servicio IPX	1000-1099

Figura 7.19. Esquema de numeración para ACL numeradas. Imagen extraída de “*Programa de la Academia de Networking CISCO CCNA2 v3.1*“ [11]

- La máscara que acompaña/completa la dirección IP, y que aparece en las sentencias *permit / deny* de la ACL, es una máscara *wildcard* [11]. Las máscaras *wildcard* no guardan relación funcional con las máscaras de subred. Se utilizan con distintos propósitos y siguen distintas reglas. Las máscaras *wildcard* se usan para filtrar direcciones IP individuales o en grupos, permitiendo o rechazando el acceso según el valor de las mismas. Dicho de

otro modo, puede concretarse que en una máscara *wildcard* presente en la sentencia de una ACL, los bit “0” significan que “*dejo pasar el valor para verificarlo a continuación*”.

Con las configuraciones anteriores de ACL, se pretende que desde un nodo emulado *EtherSwitch router*, equipos y dispositivos adscritos a una VLAN, además de poder conmutar entre sí a nivel 2, tendrán acceso a equipos y dispositivos pertenecientes a su misma VLAN en el nodo *EtherSwitch router* remoto. Pero distinta VLAN será inaccesible, tanto en el nodo local como en el remoto.

The screenshot shows a PuTTY terminal window titled "127.0.0.1 - PuTTY". The session is running on a QEMU host, with the command line showing tc@QEMU3:~\$. The terminal displays several ping commands:

- ping 192.168.2.2: Shows 100% packet loss.
- ping 155.210.157.250: Shows 100% packet loss.
- ping 192.168.5.2: Shows 0% packet loss.
- ping 192.168.4.2: Shows 100% packet loss.

Red boxes highlight the output for the failed pings (155.210.157.250 and 192.168.4.2) to indicate the expected behavior based on the configured ACLs.

Figura 7.20. *Test* de conectividad VLAN sobre el entorno de red emulado, una vez configuradas las ACL (al respecto, el *código de colores* utilizado se identifica describe en la página siguiente)

Hay que señalar que los propósitos telemáticos de las ACL configuradas pueden carecer de sentido, o la misma configuración podría implementarse de un modo optimizado, pero el objetivo es únicamente verificar la funcionalidad del entorno de emulación. Al respecto, la *figura 7.20* anterior muestra los *test* realizados con el propósito de comprobar la existencia o no de conectividad VLAN en el entorno de red emulado, y así verificar el rigor funcional del entorno de emulación según el propósito de las ACL configuradas.

Se observa como entre distintas VLAN **no existe conectividad**, tanto en el ámbito del **nodo local** como del **nodo remoto**, pero **sí existiría conectividad** entre equipos adscritos a una misma VLAN (comunicación a nivel 2) y en el **nodo remoto** (comunicación a nivel 3, no confundir con comunicación a nivel 2). Los *test* anteriores se efectúan desde el *host* virtual emulado Linux *QEMU3* (**192.168.3.2 /24**), perteneciente a VLAN3 en el nodo emulado *EtherSwitch router 1*. De igual modo, se obtienen los mismos resultados en la verificación de la conectividad VLAN desde *QEMU2*, *equipo_real 1* y *equipo_real 2*. Para evitar la saturación de la memoria del proyecto con resultados redundantes, se muestran únicamente los resultados del *test* efectuados desde *QEMU3* (**192.168.3.2 /24**)

La temática de configuración de ACL puede llegar a ser muy amplia y densa, según las características y configuraciones a practicar en un escenario de trabajo. Con objeto de ampliar dicha información, puede consultarse la bibliografía oportuna [1][2] [11].

7.2.4. Análisis del tráfico de red en el enlace *equipo real 1* ↔ *nodo virtual emulado 1* para el análisis de la *interoperabilidad* de los entornos de red.

Haciendo uso de la herramienta software *Wireshark* se realiza captura del tráfico de red en el enlace indicado, enlace en el que se efectúa la *interoperación* de los entornos de red real y emulado. Se pretende verificar en este momento el rigor y la correspondencia funcional del entorno de emulación en cuanto a la estructura de las tramas, y por consiguiente, del tráfico de red implicado en el experimento.

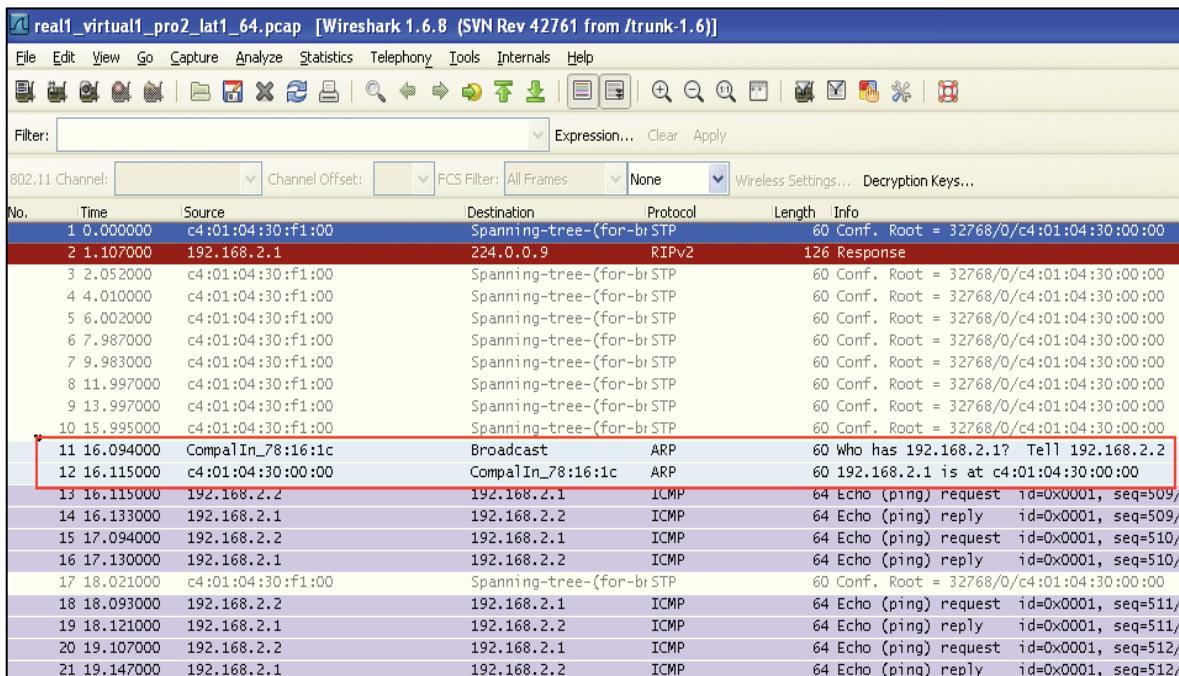


Figura 7.21.a) Tráfico de red capturado en el enlace *equipo_real 1* → *nodo virtual 1*, para una longitud de trama *Ethernet* de 64 bytes (longitud mínima, MTU = 46 bytes) en la arquitectura WAN ppp.

Una primera aproximación al tráfico capturado de las *figuras 7.21 a)* y *7.21 f)* nos permite comprobar el rigor y correspondencia funcional del emulador, verificando que realmente circulan por el entorno de red emulado tramas estructuralmente acordes al tráfico real generado con el comando ICMP (*ping*) desde *equipo real 1* externo. En segundo lugar, nos permite extraer una conclusión importante y decisiva en el análisis de la *interoperación entorno real ↔ entorno emulado*, objetivo prioritario del proyecto:

el interfaz de red *Ethernet* (del PC emulador) utilizado para integrar entornos de red reales en *GNS3/Dynamips* presenta un funcionamiento asimilable a un *proxy* "transparente" entre ambos entornos. Este interfaz "comutaría" automáticamente el tráfico de red real hacia el entorno emulado, y viceversa. Analicemos lo que sucede en la *figura 7.21 a)*. Cuando se produce la primera petición de eco (*ping*) desde el *equipo real 1* (**192.168.2.2**) con destino el interfaz virtual SVI del nodo *EtherSwitch router1* (**192.168.2.1**), al desconocer *equipo real 1* la dirección física (MAC) correspondiente a la dirección IP destino, se realiza una petición ARP en el dominio *broadcast* en que nos encontramos. Esta petición ARP es

respondida por el interfaz virtual SVI, informando de su dirección física (MAC). Al respecto, esta dirección MAC “virtual” es dinámicamente asignada por el nodo *EtherSwitch router* en el proceso de configuración VLAN y/o cada vez que ponemos en funcionamiento el dispositivo, y todos los interfaces virtuales SVI configurados tienen la misma dirección MAC virtual. Tras responder a la petición *broadcast ARP*, el *equipo real 1* tiene conformadas ya las correspondencias físicas y lógicas con las que asignar las direcciones de origen y destino a nivel 2 correspondientes a las direcciones IP. Y si comprobamos esta tabla ARP vemos que en la misma no hay ninguna entrada que haga referencia al interfaz *Ethernet* utilizado en la *interoperación*. Únicamente se hace referencia al interfaz SVI emulado. Del mismo modo, si comprobamos la tabla ARP del nodo *EtherSwitch router* emulado, vemos que esta sólo hace referencia a la dirección MAC del interfaz *Ethernet* del *equipo real 1*. Estas tablas ARP se muestran en las *figuras 7.20 b) y 7.20 c)* siguientes. Así, en todo esto subyace que el interfaz *Ethernet* empleado para implementar la *interoperación* funcionalmente puede desglosarse en un interfaz hardware real y un interfaz software emulado, produciéndose una “comutación” incondicional entre ambos. Dicho de otra manera, su funcionamiento puede asimilarse al de un *proxy* de obligado e incondicional paso/comutación entre ambos entornos de red. Esta “comutación” *entorno real ↔ entorno emulado* influirá decisivamente en la latencia de red en la *interoperación* entre ambos entornos de trabajo.

```

Dynamips(1): R1, Console port
nged state to up
*Mar 1 00:01:32.211: %LINEPROTO-5-UPDOWN: Line protocol on Interface V1
nged state to up
switch_router1#show arp
Protocol Address Age (min) Hardware Addr Type Interface
Internet 192.168.2.2 0 1c75.0878.161c ARPA Vlan2
Internet 192.168.3.1 - c401.0b84.0000 ARPA Vlan3
Internet 192.168.2.1 - c401.0b84.0000 ARPA Vlan2
switch_router1#

```

Figura 7.21.b) Tabla ARP en nodo emulado *EtherSwitch router 1*.

```

Administrator: C:\Windows\system32\cmd.exe
C:\Users\javiheavy>mac
"mac" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Users\javiheavy>arp -a
Interfaz: 192.168.2.2 --- 0xb
Dirección de Internet   Dirección física   Tipo
192.168.2.1           c4-01-04-30-00-00   dinámico
192.168.2.255         ff-ff-ff-ff-ff-ff   estatico
224.0.0.22             01-00-5e-00-00-16   estático
224.0.0.252            01-00-5e-00-00-fc   estatico

C:\Users\javiheavy>ping 192.168.4.2 -l 22 -t

```

Figura 7.21.c) Tabla ARP en *equipo real 1*.

Un análisis más detallado de una trama de la *figura 7.21 a*), por ejemplo la trama nº 15, nos muestra cómo para alcanzar **192.168.2.1** (interfaz SVI emulado del nodo 1) desde **192.168.2.2** (*equipo real 1*) la dirección MAC de destino es la dirección MAC virtual del interfaz SVI. Del mismo modo, la trama nº 16 muestra cómo para alcanzar **192.168.2.2** desde **192.168.2.1** la dirección MAC de destino es la dirección física del interfaz *Ethernet* de *equipo real 1*. Esto se ve reflejado en las *subfiguras 7.21 d) y 7.21 e)* siguientes.

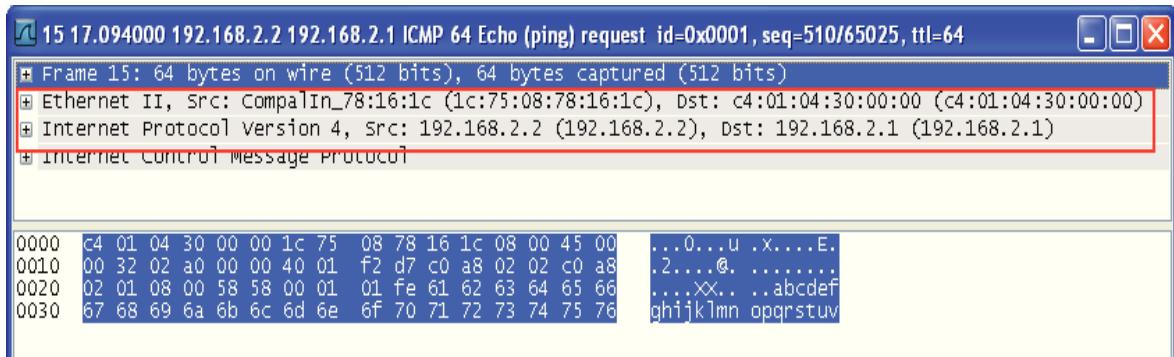


Figura 7.21 d) .Trama nº 15: ping 192.168.2.2 (*equipo real 1*) → 192.168.2.1 (interfaz SVI)

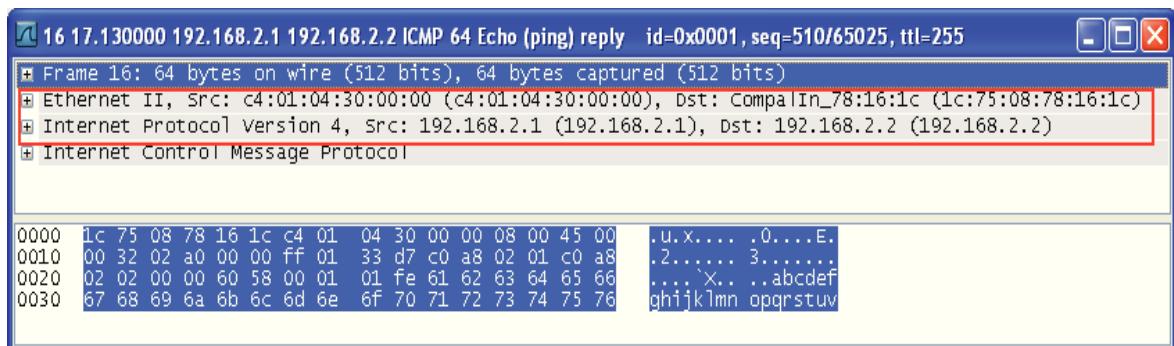


Figura 7.21 e) .Trama nº 16: ping 192.168.2.1(interfaz SVI) → 192.168.2.2 (*equipo real 1*)

Para comprobar las direcciones MAC en el nodo *EtherSwitch router1* hacemos uso del comando *Cisco IOS*: `show mac`

, desde el *modo de operación EXEC privilegiado* de la consola del dispositivo, confirmando el valor mostrado en las figuras anteriores (C4:01:04:30:00:00). También podemos observar las entradas en la tabla ARP mediante el comando: `show arp` , confirmando la entrada en la tabla que hace referencia al interfaz *Ethernet* de *equipo real 1* (1C:75:08:78:16:1C).

En conclusión, el interfaz *Ethernet* empleado en la *interoperación* puede considerarse “transparente” a los entornos real y emulado. Eso sí, tendrá una influencia manifiesta en la latencia de red en la *interoperación entorno real ↔ entorno emulado*.

1	0.000000	c4:00:04:a8:f1:0	Spanning-tree-(for-bristP)		60 Conf. Root = 32768/0/c
2	2.016000	c4:00:04:a8:f1:0	Spanning-tree-(for-bristP)		60 Conf. Root = 32768/0/c
3	2.266000	CompalIn_78:16:1	Broadcast	ARP	60 who has 192.168.2.1?
4	2.328000	c4:00:04:a8:00:0	CompalIn_78:16:1c	ARP	60 192.168.2.1 is at c4:0
5	2.344000	192.168.2.2	192.168.2.1	ICMP	1514 Echo (ping) request i
6	2.391000	192.168.2.1	192.168.2.2	ICMP	1514 Echo (ping) reply i
7	3.281000	192.168.2.2	192.168.2.1	ICMP	1514 Echo (ping) request i
8	3.328000	192.168.2.1	192.168.2.2	ICMP	1514 Echo (ping) reply i
9	4.016000	c4:00:04:a8:f1:0	Spanning-tree-(for-bristP)		60 Conf. Root = 32768/0/c
10	4.266000	192.168.2.2	192.168.2.1	ICMP	1514 Echo (ping) request i
11	4.328000	192.168.2.1	192.168.2.2	ICMP	1514 Echo (ping) reply i
12	5.266000	192.168.2.1	224.0.0.9	RIPV2	126 Response
13	5.281000	192.168.2.2	192.168.2.1	ICMP	1514 Echo (ping) request i
14	5.328000	192.168.2.1	192.168.2.2	ICMP	1514 Echo (ping) reply i
15	6.016000	c4:00:04:a8:f1:0	Spanning-tree-(for-bristP)		60 Conf. Root = 32768/0/c
16	6.281000	192.168.2.2	192.168.2.1	ICMP	1514 Echo (ping) request i
17	6.328000	192.168.2.1	192.168.2.2	ICMP	1514 Echo (ping) reply i
18	7.281000	192.168.2.2	192.168.2.1	ICMP	1514 Echo (ping) request i
19	7.344000	192.168.2.1	192.168.2.2	ICMP	1514 Echo (ping) reply i
20	8.016000	c4:00:04:a8:f1:0	Spanning-tree-(for-bristP)		60 Conf. Root = 32768/0/c
21	8.281000	192.168.2.2	192.168.2.1	ICMP	1514 Echo (ping) request i
22	8.328000	192.168.2.1	192.168.2.2	ICMP	1514 Echo (ping) reply i
23	9.297000	192.168.2.2	192.168.2.1	ICMP	1514 Echo (ping) request i

Figura 7.21 f). Tráfico de red capturado en el enlace equipo_rea1 1 → nodo virtual 1, para una longitud de trama *Ethernet* de 1514 bytes (MTU = 1500 bytes) en la arquitectura WAN ppp.

7.2.5. Configuraciones de red adicionales para el análisis de prestaciones y rendimiento del emulador *GNS3/Dynamips* en la arquitectura WAN ppp.

Para configurar el escenario global de trabajo de la *figura 3.1* que nos permite abordar el análisis de la capacidad y rendimiento del sistema emulador del capítulo 4 procedemos como se indica a continuación. Por comodidad y claridad en la exposición, se muestra de nuevo la *figura 3.1* en la página siguiente.

Siendo así, y de cara a establecer una relación más significativa entre el número de nodos activos participantes en el escenario emulado de trabajo (*router-switch multilayer*) y el rendimiento/consumo de recursos del sistema emulador, sobre nuestra arquitectura de red principal se añaden 2 enlaces *serie ppp* que interconectan los nodos *EtherSwitch router 1* y *EtherSwitch router 2* con sendos nodos emulados *EtherSwitch router*, que denominamos nodos Auxiliar 1 y 2, sin ningún propósito especial de diseño telemático. El único objetivo es evaluar la repercusión directa sobre el rendimiento del sistema emulador según el número de nodos emulados activos en un escenario (para evaluar la potencial escalabilidad del diseño de la arquitectura).

Desde la perspectiva teórica de administración & gestión de redes a nivel 3, estamos interconectando (mediante enlaces *serie WAN ppp*) nuestra arquitectura de trabajo principal, la subred **155.210.157.248 / 30**, con las subredes **155.210.157.240 / 30** y **155.210.157.244 / 30**, todas ellas subredes de la red mayor **155.210.0.0 / 16** (esquema general administrativo reflejado en la *figura 7.22*).

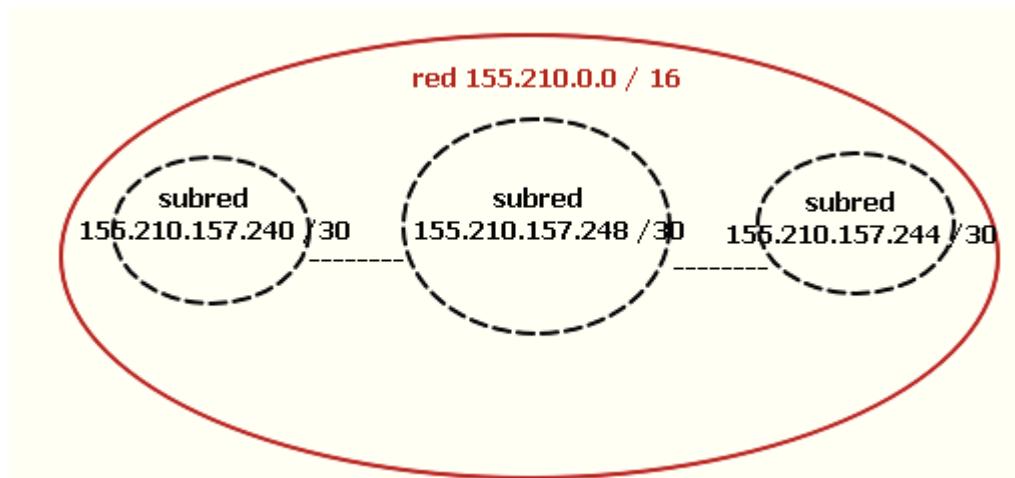


Figura 7.22. Esquema administrativo de diseño a nivel 3, según el modelo de referencia OSI, configurado para el análisis de la capacidad y rendimiento del sistema emulador.

Este “rediseño” ampliado de la arquitectura de red principal, con las correspondientes direcciones IP y máscaras de subred asignadas (recordemos, sin ningún propósito especial telemático de diseño), permite extraer los resultados mostrados en las *tablas 4.2 y 4.3* del análisis de la capacidad y rendimiento del sistema emulador del capítulo 4.

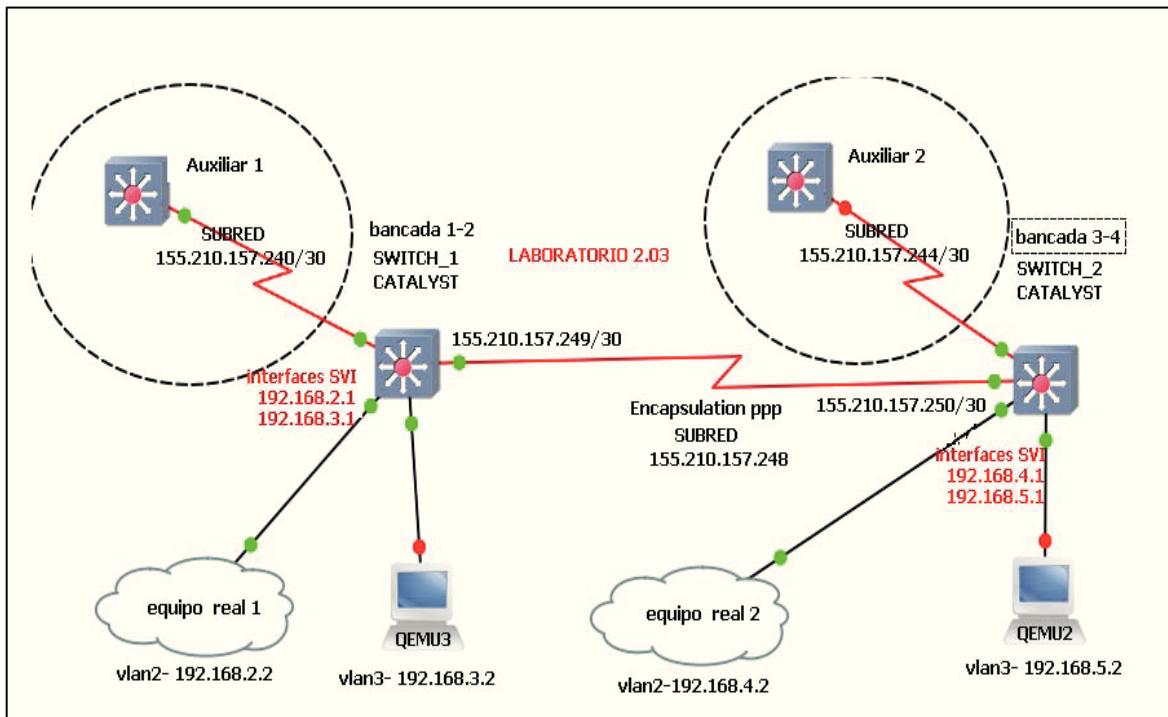


Figura 3.1. Topología/simulación gráfica en *GNS3* de la arquitectura de red WAN *ppp* (Nodos/conexiones/subredes incorporados en la arquitectura de red principal *WAN ppp* para analizar el rendimiento y prestaciones del sistema emulador).

Configuramos a continuación los nodos *EtherSwitch router* Auxiliar 1 y 2.

Para configurar las nuevas subredes de este “nuevo” escenario ampliado de trabajo accedemos *vía consola* en los nodos emulados al entorno de configuración *Cisco IOS*, asignando las direcciones de red necesarias en los interfaces que interconectan los nodos, según las *figuras 7.23 y 7.24* mostradas en la página siguiente. Recordemos que desde la perspectiva teórica de administración & gestión de redes a nivel 3 según el modelo de referencia OSI, estamos interconectando (mediante enlaces *serie ppp*) nuestro escenario de trabajo principal, la subred **155.210.157.248 / 30**, con las subredes **155.210.157.240 / 30** y **155.210.157.244 / 30**, todas ellas subredes de la red mayor **155.210.0.0 / 16** (esquema general administrativo reflejado en la *figura 7.22*).

- Nodos EtherSwitch router 1 / auxiliar 1 (subred 155.210.157.240 / 30)

The figure consists of two separate terminal windows. The top window is titled 'Dynamips(2): R1, Console port' and shows the configuration of a serial interface on router R1. The bottom window is titled 'Dynamips(0): R3, Console port' and shows the configuration of a serial interface on router R3. Both windows have standard window controls (minimize, maximize, close) at the top right.

```
Dynamips(2): R1, Console port
switch_router1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch_router1(config)#interface serial 2/1
switch_router1(config-if)#ip address 155.210.157.241 255.255.255.252
switch_router1(config-if)#no shutdown
switch_router1(config-if)#exit
switch_router1(config)#

Dynamips(0): R3, Console port
R3(config)#hostname auxiliar1
auxiliar1(config)#interface serial 0/0
auxiliar1(config-if)#ip address 155.210.157.242 255.255.255.252
auxiliar1(config-if)#no shutdown
auxiliar1(config-if)#exit
auxiliar1(config)#

```

Figura 7.23. Comandos Cisco IOS para la configuración de la subred 155.210.157.240 / 30.

- Nodos EtherSwitch router 2 / auxiliar 2 (subred 155.210.157.244 / 30)

The figure consists of two separate terminal windows. The top window is titled 'Dynamips(1): R2, Console port' and shows the configuration of a serial interface on router R2. The bottom window is titled 'Dynamips(4): R4, Console port' and shows the configuration of a serial interface on router R4. Both windows have standard window controls (minimize, maximize, close) at the top right.

```
Dynamips(1): R2, Console port
switch_router2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch_router2(config)#interface serial 2/1
switch_router2(config-if)#ip address 155.210.157.245 255.255.255.252
switch_router2(config-if)#no shutdown
switch_router2(config-if)#exit
switch_router2(config)#

Dynamips(4): R4, Console port
R4#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)#hostname auxiliar2
auxiliar2(config)#interface serial 0/0
auxiliar2(config-if)#ip address 155.210.157.246 255.255.255.252
auxiliar2(config-if)#no shutdown
auxiliar2(config-if)#exit
auxiliar2(config)#

```

Figura 7.24. Comandos Cisco IOS para la configuración de la subred 155.210.157.244 / 30.

Una vez configurada completamente la arquitectura de red para el análisis global de los propósitos de los capítulos 3 y 4, y para profundizar un poco más sobre nuestros análisis, recuperamos de nuevo, por ejemplo, la *tabla 4.2* del capítulo 4. Utilizamos esta tabla para hacer ciertas consideraciones teórico-prácticas sobre los *host* emulados Linux QEMU.

Configuración hardware 1	Carga de CPU (<i>offmode</i>) ³	Memoria RAM en uso en MB (<i>offmode</i>)	Carga media de CPU (<i>pseudoactivo</i>) ⁴	Memoria RAM en uso en MB (<i>pseudoactivo</i>)
1 <i>switch-router</i> activo	1% max.	318	1-2 %	780 → 540
2 <i>switch-router</i> activos	1% max.	318	4 % → 2-3 %	940 → 720
2 <i>switch-router</i> activos + 1 <i>QEMU</i>	1% max.	318	16 % → 14 %	950 → 750
2 <i>switch-router</i> activos + 2 <i>QEMU</i>	1% max.	318	17 % → 15 %	960 → 760
2 <i>switch-router</i> activos + 1 auxiliar (sin <i>QEMU</i>)	1% max.	318	6 % → 4-5 %	970 → 760
2 <i>switch-router</i> activos + 2 auxiliar (sin <i>QEMU</i>)	1% max.	318	11% → 9-10 %	980 → 790

Tabla 4.2. Consumo de recursos del sistema emulador para la *configuración hardware 1*, bajo S.O. Windows XP 32 bits y arquitectura de red WAN ppp.

Conclusiones sobre el uso de *host* emulados Linux QEMU

- Los terminales *QEMU* son *host* virtuales que emulan sistemas operativos Linux. A la vista de los resultados mostrados en la *tabla 4.2*, si comparamos paralelamente el consumo de CPU entre las filas nº2 ↔ nº3 se aprecia que el uso de dichos terminales eleva significativamente (un 12 % aproximadamente) la carga de CPU.

³ Estado inicial en el que ningún nodo y/o terminal QEMU está en funcionamiento.

⁴ Estado en el que el tráfico de red presente es el relacionado con el mantenimiento y actualización de los protocolos configurados en el escenario de trabajo (información dinámica de nivel 2 y 3 → escaso tráfico)

- Aunque igualmente se constata que el incremento en el consumo de CPU *relativo* al pasar de 1 QEMU activo a 2 QEMU activos (**fila nº3→ fila nº4**) supone únicamente el 1%, el uso de dichos terminales parece no ser recomendable en demasía para los entornos de emulación *GNS3/Dynamips*, toda vez que cargas elevadas del procesador y consumos elevados de memoria RAM pueden repercutir negativamente en la ejecución en tiempo real de los procesos asociados con la emulación, incidiendo directamente sobre parámetros de red como la latencia. En estudios futuros, y sobre equipos emuladores de una gama tecnológica superior, cabría incluirse el estudio de más terminales QEMU para analizar esa tendencia aparentemente “asintótica” en la evolución del consumo de CPU al aumentar su número. Siendo así, el uso de los mismos se justifica en el análisis realizado en el capítulo 3, paralelamente con el anexo 2, como *host* para verificar la conectividad VLAN, y así analizar la funcionalidad y rigor del emulador *GNS3/Dynamips*. Otra justificación de su uso, es mostrar la posibilidad de utilización en diseños emulados.

Dado que la *interoperación* entre el tráfico de red real y emulado no eleva significativamente el consumo de recursos del sistema, y así se comprueba empíricamente, sobre el escenario actual, en el que básicamente el tráfico de red presente en el mismo es el relacionado con la actualización y mantenimiento de los protocolos de nivel 2 y 3 configurados en la arquitectura de red (básicamente *point-to-point* y RIP v2), parece adecuado el uso de *equipos reales externos* mejor que los *host virtuales* QEMU, como *host* integrantes de la arquitectura y que subsidiariamente nos permitan analizar y verificar las configuraciones de diseño y la conectividad VLAN (y por consiguiente, la funcionalidad y rigor del entorno de emulación), además de la *interoperación entorno real ↔ entorno emulado*. Pero esta deseable disposición de diseño y trabajo puede presentar varios inconvenientes logísticos: primero, en cuanto a la disponibilidad de *equipos reales externos* (interfaces de red *Ethernet* externos al emulador); en segundo lugar, en cuanto a que todo equipo real que se desee incorporar al entorno emulado precisa de un interfaz físico libre y específico en el PC emulador para su integración (esto es, un interfaz de red *Ethernet*).

7.2.6. Configuración y verificación de la arquitectura LAN Ethernet/VTP.

La figura 7.25 nos recuerda la topología de la arquitectura LAN Ethernet/VTP, objeto de análisis y previamente planteada en el desarrollo del capítulo 3.

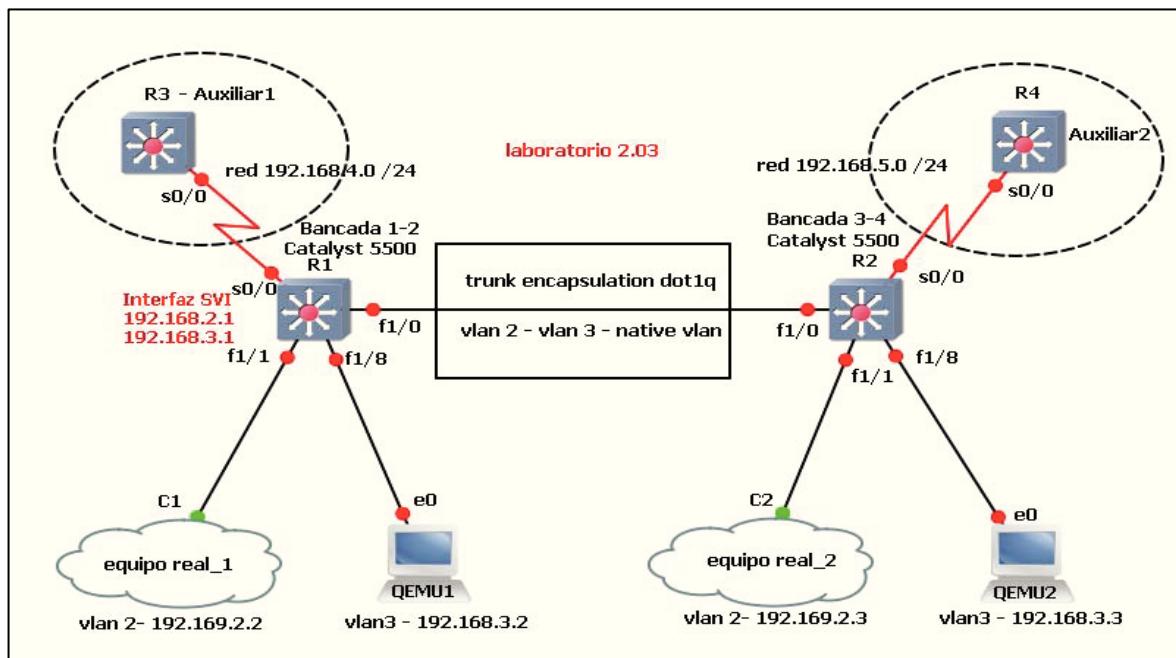


Figura 7.25. Topología/simulación gráfica en GNS3 de la arquitectura LAN /VTP.

Las consideraciones iniciales para la configuración en *GNS3/Dynamips* de la arquitectura de red actual, y respecto a la arquitectura WAN ppp, podemos establecerlas:

- Entorno de red LAN Ethernet, conmutado a nivel 2 y 3 (recordemos, gracias a las características presentes en dispositivos *switch multilayer* como el *Cisco Catalyst 5500 Switch* del laboratorio que permiten implementar *InterVLAN routing*) [3][10].
- Las *bancadas 1-2* y *3-4* del laboratorio 2.03 se interconectan por medio de un enlace *FastEthernet* que actúa como enlace troncal (*trunk link*) por el que circulan, mediante el protocolo 802.1q, VLAN2, VLAN3 y *native VLAN*. Todo esto se explica en detalle a continuación.
- En cuanto al interfaz gráfico/simulación de la topología en *GNS3*: interconexión de los nodos *EtherSwitch router* por medio de un enlace *FastEthernet*, sustituyendo el enlace *serie point-to-point* de la arquitectura WAN ppp anterior.

- Los equipos reales que se integran en el escenario emulado mediante enlaces *FastEthernet* se adscriben administrativamente a VLAN2, con direcciones IP y máscaras asignadas según el diseño de la *figura 7.25*. Los equipos QEMU se consideran a efectos de diseño integrantes de VLAN3, en el espacio de direcciones indicado.

Se implementa sobre *GNS3/Dynamips*, la configuración de los dispositivos de red emulados imagen/similares a los *Cisco Catalyst 5500 Switch* del laboratorio, ejecutando para ello los comandos *Cisco IOS* apropiados, haciendo hincapié de nuevo en que esta configuración en el entorno emulado es funcionalmente la misma que se implementaría en el entorno de red real. Al igual que en el *apartado 7.2.1*, es preciso incorporar en los nodos *EtherSwitch router* del entorno emulado de trabajo los módulos de interfaces virtuales apropiados que nos permitan reproducir equipos emulados similares a los reales, así como configurar la arquitectura de red de trabajo. Con respecto a la arquitectura de red WAN ppp, no es preciso incorporar ningún módulo hardware adicional.

Tras “iniciar” los dispositivos de red *EtherSwitch router 1* y *2*, y acceder a la consola de los mismos, procedemos a su configuración conforme al diseño planteado en la *figura 7.25* anterior. La interconexión de los equipos *EtherSwitch router*, que en el entorno de red real conectarían las *bancadas 1-2* y *3-4* del laboratorio de telemática *2.03*, se realiza a través de un enlace *FastEthernet*. Éste será el enlace troncal que permite realizar *switching* a nivel 2 e *InterVLAN routing* (*switching* a nivel 3) entre VLAN, mediante el protocolo de enlace troncal 802.1q. A diferencia de otros protocolos, 802.1q es un protocolo que usa etiquetado de tramas (*tagged* y *untagged frames*) [10] que facilita la administración y confiere mayor rapidez en la transmisión que la tecnología troncal de filtrado de tramas.

Sobre la topología anterior, procedemos a implementar las configuraciones en base a comandos *Cisco IOS*, que permiten abordar su análisis funcional en el entorno emulado. Primeramente, se aborda la configuración de los nodos emulados *EtherSwitch router 1* y *2* que constituyen la infraestructura principal de esta arquitectura de red. Posteriormente, en el *apartado 7.2.7*, se configuran los nodos *EtherSwitch router Auxiliar 1* y *Auxiliar 2* para el análisis de los propósitos del capítulo 4.

Configuración VLAN (VLAN2 y VLAN3) en EtherSwitch router 1 (servidor VTP)

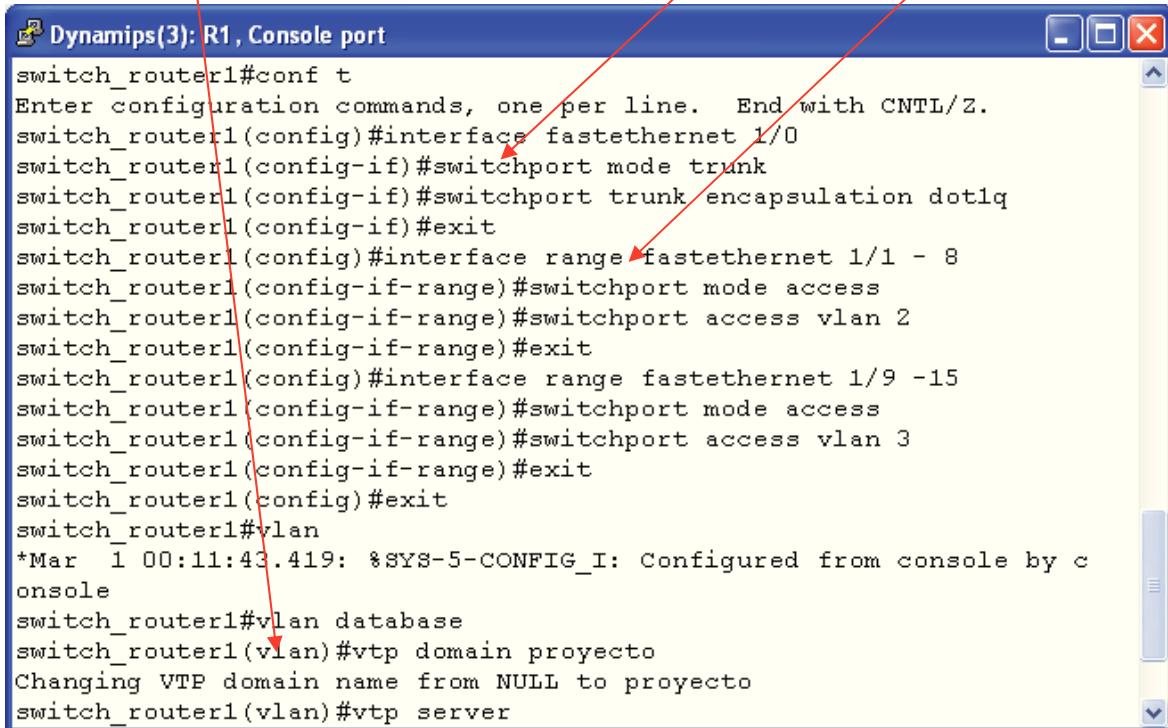
Con respecto a la arquitectura de red WAN ppp se introducen varias modificaciones.

Con el objetivo de conferir facilidades de *InterVLAN routing* a todo el entorno emulado de trabajo, el nodo *EtherSwitch router 1* será el único que implementará funcionalidad de conmutación a nivel 3 (*switch multilayer*), teniendo el nodo *EtherSwitch router 2* únicamente funcionalidad a nivel 2 (mismo dispositivo, pero sólo con la función de *switch general activa*), precisando así éste último del enlace troncal para poder realizar *switching* entre equipos adscritos a la misma VLAN, pero en nodos diferentes, e *InterVLAN routing* entre VLAN diferentes.

Con ello, el mapa de direcciones IP para el escenario, según la *figura 7.25*, será:

**VLAN2 : 192.168.2.0 / 24
VLAN3 : 192.168.3.0 / 24**

En las mismas observamos, respectivamente, la configuración del enlace troncal sobre el interfaz *FastEthernet 1/0* según el protocolo 802.1q, la asignación física de interfaces FastEthernet a VLAN2 y VLAN3 en el dispositivo, así como la creación y asignación del dominio y funcionalidad VTP del dispositivo (*figura 7.26*); posteriormente, se muestra la creación /configuración de VLAN2 y VLAN3 (*figura 7.27*), según el diseño establecido previamente.

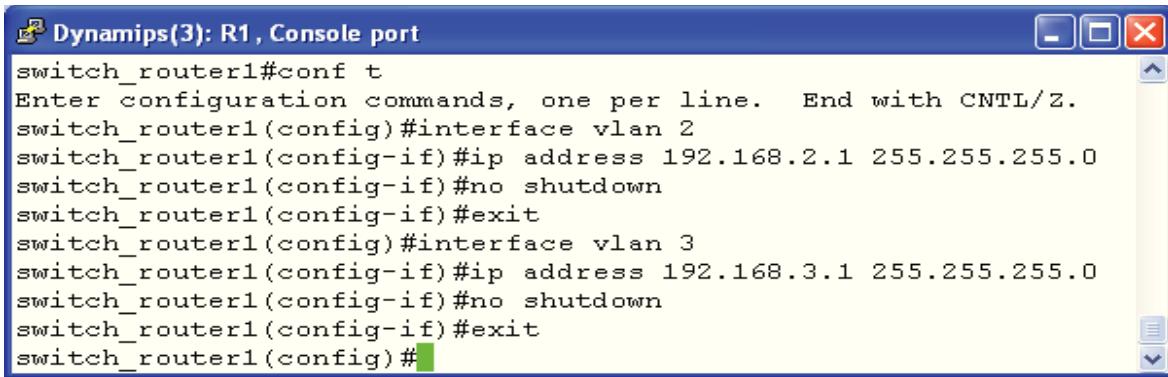


The screenshot shows a terminal window titled "Dynamips(3): R1 , Console port". The window contains the following configuration commands:

```
switch_router1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch_router1(config)#interface fastethernet 1/0
switch_router1(config-if)#switchport mode trunk
switch_router1(config-if)#switchport trunk encapsulation dot1q
switch_router1(config-if)#exit
switch_router1(config)#interface range fastethernet 1/1 - 8
switch_router1(config-if-range)#switchport mode access
switch_router1(config-if-range)#switchport access vlan 2
switch_router1(config-if-range)#exit
switch_router1(config)#interface range fastethernet 1/9 - 15
switch_router1(config-if-range)#switchport mode access
switch_router1(config-if-range)#switchport access vlan 3
switch_router1(config-if-range)#exit
switch_router1(config)#exit
switch_router1#vlan
*Mar 1 00:11:43.419: %SYS-5-CONFIG_I: Configured from console by c
onsole
switch_router1#vlan database
switch_router1(vlan)#vtp domain proyecto
Changing VTP domain name from NULL to proyecto
switch_router1(vlan)#vtp server
```

Figura 7.26. Configuración del nodo *emulado EtherSwitch router 1*: configuración del enlace troncal, asignación de interfaces *FastEthernet* a VLAN2 y VLAN3, y características VTP.

Como en este escenario de trabajo los dispositivos de red no operan a nivel de red, sino que realizan commutación a nivel 2 y 3 (*switching* e *InterVLAN routing*), no habrá intercambio de información a nivel 3 entre dispositivos, y por lo tanto no se configurarán direcciones IP en los interfaces *Fastethernet* asignados al enlace troncal: recordemos que hablamos de un escenario LAN *Ethernet* commutado. Solo se asignan características de nivel 3 a los interfaces virtuales SVI a configurar en las VLAN, para poder implementar *InterVLAN routing*. Idéntica configuración en el entorno de red real, sobre los dispositivos de red *Cisco Catalyst 5500 Switch*, precisaría de comandos *Cisco CatOS* tal como se indicó en el apartado 2.4, siguiendo para ello las indicaciones del anexo 1.



```

Dynamips(3): R1, Console port
switch_routerr1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch_routerr1(config)#interface vlan 2
switch_routerr1(config-if)#ip address 192.168.2.1 255.255.255.0
switch_routerr1(config-if)#no shutdown
switch_routerr1(config-if)#exit
switch_routerr1(config)#interface vlan 3
switch_routerr1(config-if)#ip address 192.168.3.1 255.255.255.0
switch_routerr1(config-if)#no shutdown
switch_routerr1(config-if)#exit
switch_routerr1(config)#

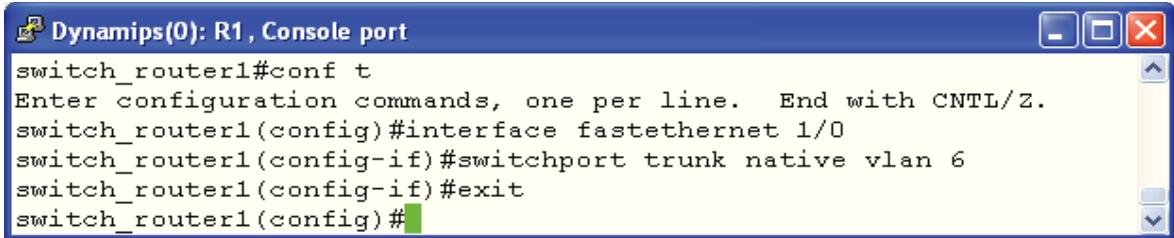
```

Figura 7.27. Creación de VLAN2 y VLAN3 (y sus respectivos interfaces virtuales SVI).

Se observa en la figura anterior la creación de los interfaces virtuales SVI asociados a cada VLAN que permitirán realizar *InterVLAN routing* entre las VLAN del entorno emulado de trabajo. Sobre este escenario, las tramas adscritas a VLAN2 y VLAN3 que circulen por el enlace troncal lo harán identificadas como *tagged frames*, mediante un campo a tal efecto que las identificará únicamente. Tramas que no pertenezcan a ninguna de estas VLAN, circularían por el enlace troncal como *untagged frames*, utilizando para ello el identificador de la VLAN *nativa*. Y aquí radica la importancia de la definición de una VLAN nativa en un escenario commutado VLAN. En este sentido, se consideran fundamentalmente las siguientes particularidades:

- Por defecto, en dispositivos de red *Cisco Systems*, la VLAN *nativa* por defecto es VLAN1, no existiendo inconveniente administrativo para su cambio.
- Un puerto de un enlace troncal soporta únicamente una VLAN *nativa*. Para un enlace troncal, debe existir concordancia en la VLAN *nativa* configurada en ambos extremos del mismo, produciéndose errores en caso contrario.

En el supuesto caso de utilizar algún interfaz asignado a VLAN1 en un dispositivo de red para fines administrativos (no es el caso que nos ocupa, pero puede ser bastante frecuente en la práctica), deberá considerarse el hecho de configurar una VLAN *nativa* diferente, en vez de la asignada por defecto (VLAN1), y con la misma denominación en ambos extremos del enlace troncal para evitar errores. Se procede, como refleja la figura 7.28 de la página siguiente, a cambiar la configuración de la VLAN *nativa* (asignamos VLAN6 como tal) en el enlace troncal emulado, mediante comandos *Cisco IOS*.

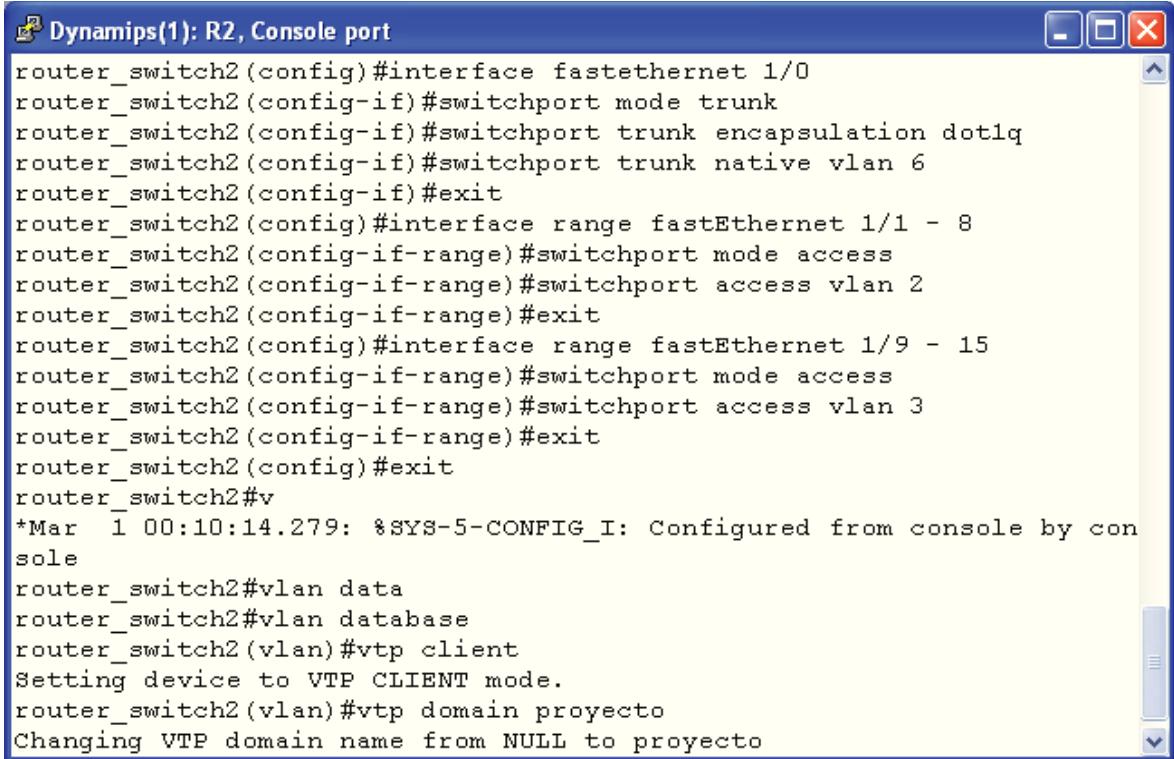


```
switch_router1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch_router1(config)#interface fastethernet 1/0
switch_router1(config-if)#switchport trunk native vlan 6
switch_router1(config-if)#exit
switch_router1(config)#[
```

Figura 7.28. Configuración de VLAN nativa en el interfaz *Fastethernet 1/0* asignado al enlace troncal en el nodo emulado *EtherSwitch router 1*.

Configuración VLAN (VLAN2 y VLAN3) en *EtherSwitch router 2* (cliente VTP)

Como refleja la figura 7.29, y habiendo configurado previamente sobre el dispositivo *servidor* el dominio VTP del escenario comutado de trabajo, así como la segmentación administrativa según el diseño en VLAN2 y VLAN3 (además de la VLAN *nativa*), asignaremos en el nodo *Etherswitch router 2* los interfaces *Fastethernet* a cada VLAN, según el diseño planteado. Igualmente, se configura el enlace troncal sobre el interfaz *Fastethernet 1/0*. Dotaremos al nodo de funcionalidad VTP *cliente* y lo incorporamos como tal al dominio de trabajo. Según esta funcionalidad VTP, este dispositivo no podrá crear VLAN ni generar cambios administrativos en el escenario de trabajo, pero sí actualizará la información que le llegue del *servidor* VTP a tal efecto, y la propagará a otros dispositivos del entorno comutado.



```
router_switch2(config)#interface fastethernet 1/0
router_switch2(config-if)#switchport mode trunk
router_switch2(config-if)#switchport trunk encapsulation dot1q
router_switch2(config-if)#switchport trunk native vlan 6
router_switch2(config-if)#exit
router_switch2(config)#interface range fastEthernet 1/1 - 8
router_switch2(config-if-range)#switchport mode access
router_switch2(config-if-range)#switchport access vlan 2
router_switch2(config-if-range)#exit
router_switch2(config)#interface range fastEthernet 1/9 - 15
router_switch2(config-if-range)#switchport mode access
router_switch2(config-if-range)#switchport access vlan 3
router_switch2(config-if-range)#exit
router_switch2(config)#exit
router_switch2#v
*Mar 1 00:10:14.279: %SYS-5-CONFIG_I: Configured from console by console
router_switch2#vlan data
router_switch2#vlan database
router_switch2(vlan)#vtp client
Setting device to VTP CLIENT mode.
router_switch2(vlan)#vtp domain proyecto
Changing VTP domain name from NULL to proyecto
```

Figura 7.29. Configuración del nodo *emulado EtherSwitch router 2*: configuración del enlace troncal, asignación de interfaces *Fastethernet* a VLAN2 y VLAN3, y características VTP.

Al respecto, la *tabla 7.1* muestra las características y prestaciones telemáticas principales del protocolo VTP.

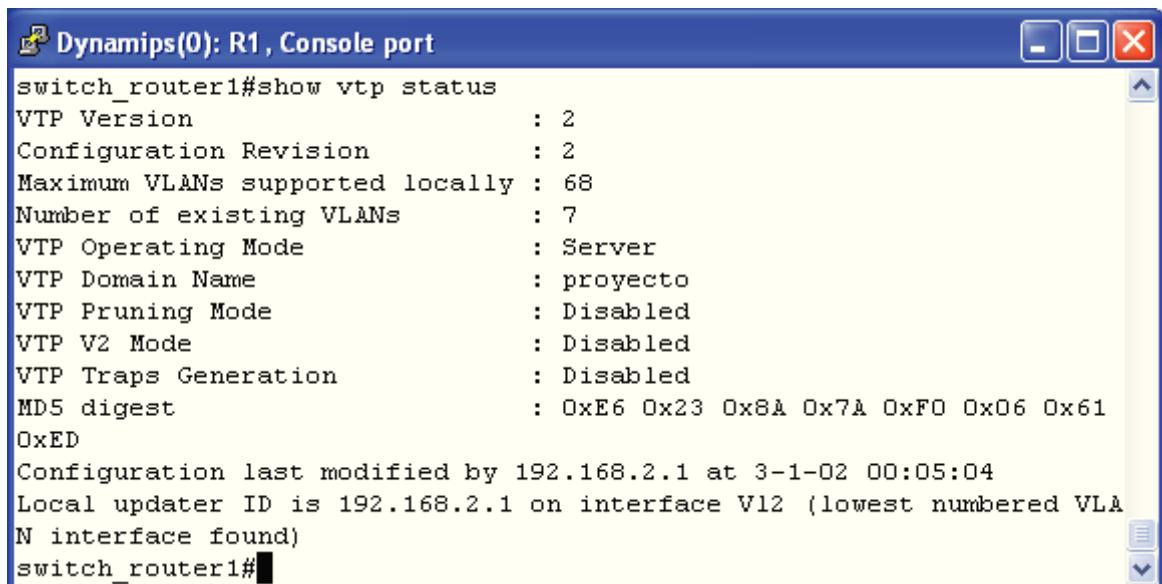
Protocolo VTP	Características / prestaciones de VTP
Objetivo/función principal del protocolo	<ul style="list-style-type: none"> La configuración VLAN mantiene <u>unidad</u> dentro de un <u>dominio</u> administrativo común. Permite <u>reducir la complejidad</u> de la administración y la monitorización de redes segmentadas en VLAN. <u>Coherencia</u> en la configuración VLAN de toda una red comutada.
Características principales del protocolo	<ul style="list-style-type: none"> Configuración <i>plug&play</i> al agregar nuevas VLAN. Generación de <u>informes dinámicos</u> de VLAN agregadas/modificadas/eliminadas en el servidor VTP. Usa <u>tramas de enlace troncal de nivel 2</u> para agregar y configurar administrativamente las VLAN en un dominio. Los <u>mensajes del protocolo</u> que circulan por un <u>dominio VTP</u> configurado se encapsulan en las tramas del protocolo de enlace <i>Inter-Switch (ISL)</i>, propietario de <i>Cisco Systems</i>, o <u>802.1q</u> (que es nuestro caso), y se envían mediante el uso de enlaces troncales a otros dispositivos de red (en el caso de tramas del protocolo 802.1q, se añadirá una cabecera “<i>Tag</i>” a la trama <i>Ethernet</i> para etiquetarla). Admite <u>cambios centralizados</u> en un dispositivo de red que ejerce el rol de <i>servidor</i> del dominio, y que se comunican a todos los demás <i>switch</i> de un mismo entorno de red.
Roles ejercidos por los dispositivos	<ul style="list-style-type: none"> Servidor: pueden crear, modificar y eliminar la VLAN y los parámetros de configuración de VLAN de todo un dominio, así como enviar mensajes VTP a través de los puertos de un enlace troncal. Cliente: no pueden crear, modificar ni eliminar la información de VLAN. La única función de los dispositivos en modo <i>cliente</i> VTP es procesar los cambios de VLAN y enviar mensajes VTP desde los puertos de un enlace troncal. Transparente: envían publicaciones VTP, ignorando la información contenida en el mensaje. Un <i>switch</i> en modo transparente no modifica su base de datos al recibir actualizaciones o al enviar una actualización indicando cambios en el estado de la configuración VLAN del dominio.

Tabla 7.1. Tabla resumen de las características y prestaciones principales del protocolo VTP y que se aplican en el diseño particular del escenario de trabajo.

Verificación de la conectividad VLAN y funcionalidades VTP del emulador

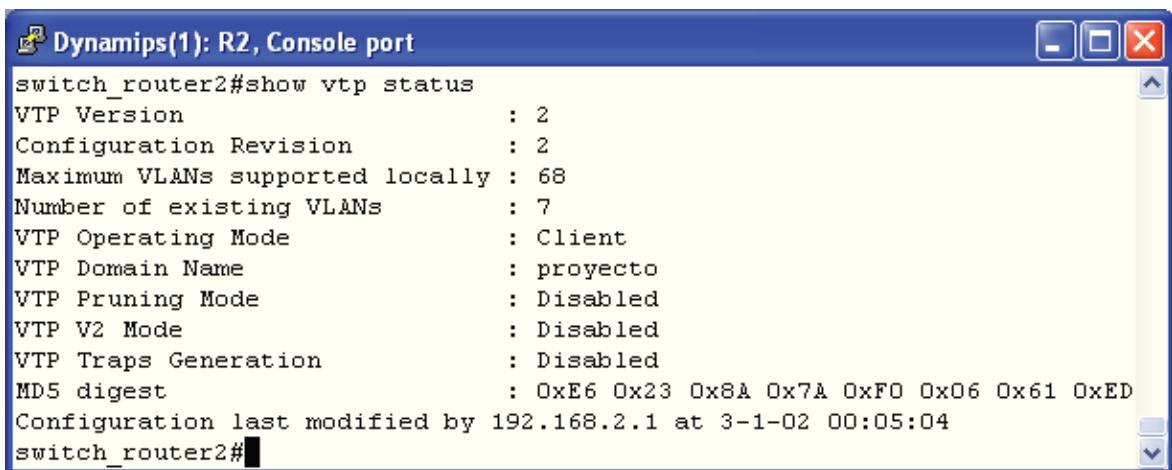
Una vez realizadas las configuraciones anteriores, procedemos a verificar la conectividad VLAN en el entorno emulado, verificando así la funcionalidad del emulador.

Según nuestro diseño, debe existir conectividad completa entre VLAN2 y VLAN3, generándose comunicación a nivel 3 en la comunicación entre ambas VLAN a través del nodo *EtherSwitch router 1*, que recordemos es un *switch multilayer*. Primeramente, comprobamos la funcionalidad y características VTP activas en ambos nodos emulados, desde el *modo de configuración EXEC privilegiado*:



```
Dynamips(0): R1, Console port
switch_router1#show vtp status
VTP Version : 2
Configuration Revision : 2
Maximum VLANs supported locally : 68
Number of existing VLANs : 7
VTP Operating Mode : Server
VTP Domain Name : proyecto
VTP Pruning Mode : Disabled
VTP V2 Mode : Disabled
VTP Traps Generation : Disabled
MD5 digest : 0xE6 0x23 0x8A 0x7A 0xF0 0x06 0x61
0xED
Configuration last modified by 192.168.2.1 at 3-1-02 00:05:04
Local updater ID is 192.168.2.1 on interface V12 (lowest numbered VLAN interface found)
switch_router1#
```

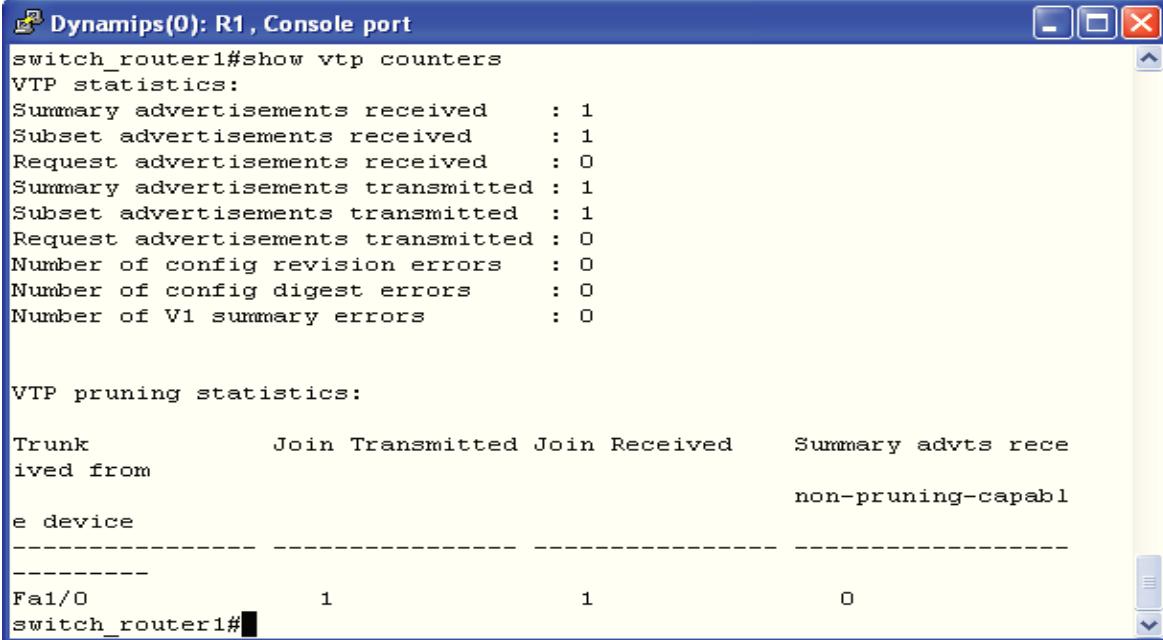
Figura 7.30. Estado VTP en el nodo emulado *EtherSwitch router 1* (servidor VTP).



```
Dynamips(1): R2, Console port
switch_router2#show vtp status
VTP Version : 2
Configuration Revision : 2
Maximum VLANs supported locally : 68
Number of existing VLANs : 7
VTP Operating Mode : Client
VTP Domain Name : proyecto
VTP Pruning Mode : Disabled
VTP V2 Mode : Disabled
VTP Traps Generation : Disabled
MD5 digest : 0xE6 0x23 0x8A 0x7A 0xF0 0x06 0x61 0xED
Configuration last modified by 192.168.2.1 at 3-1-02 00:05:04
switch_router2#
```

Figura 7.31. Estado VTP en el nodo emulado *EtherSwitch router 2* (cliente VTP).

Según las anteriores *figuras 7.30* y *7.31*, los nodos 1 y 2 estarían ejecutando en el entorno emulado los roles previamente asignados en la configuración emulada VTP (*servidor* y *cliente* respectivamente). Seguidamente, las *figuras 7.32* y *7.33* constatan que se ha producido la transmisión de la información relativa a la configuración VLAN implementada en el *servidor*, mediante el protocolo VTP.



```

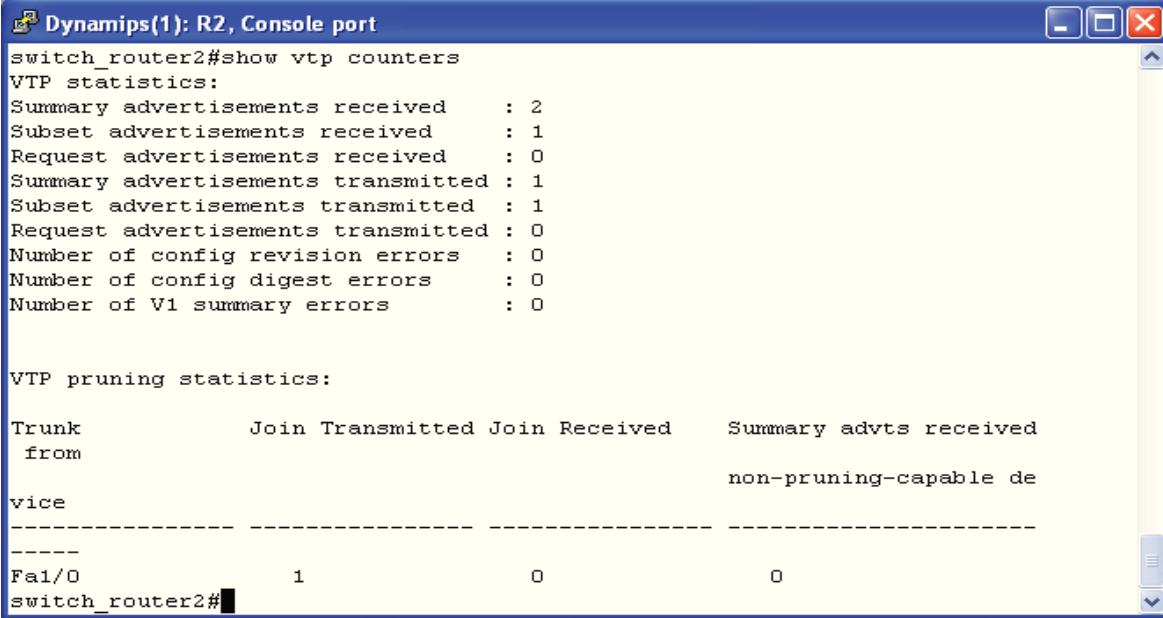
Dynamips(0): R1, Console port
switch_router1#show vtp counters
VTP statistics:
Summary advertisements received      : 1
Subset advertisements received       : 1
Request advertisements received     : 0
Summary advertisements transmitted   : 1
Subset advertisements transmitted    : 1
Request advertisements transmitted  : 0
Number of config revision errors   : 0
Number of config digest errors     : 0
Number of V1 summary errors        : 0

VTP pruning statistics:

Trunk          Join Transmitted Join Received   Summary advts rece
ived from                               non-pruning-capabl
e device
-----
----- Fa1/0           1                 1             0
switch_router1#

```

Figura 7.32. Relación de la información transmitida y recibida por el servidor emulado VTP.



```

Dynamips(1): R2, Console port
switch_router2#show vtp counters
VTP statistics:
Summary advertisements received      : 2
Subset advertisements received       : 1
Request advertisements received     : 0
Summary advertisements transmitted   : 1
Subset advertisements transmitted    : 0
Request advertisements transmitted  : 0
Number of config revision errors   : 0
Number of config digest errors     : 0
Number of V1 summary errors        : 0

VTP pruning statistics:

Trunk          Join Transmitted Join Received   Summary advts received
from                               non-pruning-capable de
vice
-----
----- Fa1/0           1                 0             0
switch_router2#

```

Figura 7.33. Relación de la información transmitida y recibida por el cliente emulado VTP.

A continuación, verificamos la conectividad y el trayecto que sigue el tráfico en la conmutación emulada entre equipos de una misma VLAN (comutación a nivel 2), y entre VLAN diferentes (comutación a nivel 3) en el nodo *EtherSwitch router 2*. Para ello, habremos configurado previamente *equipo_real 1*, *equipo_real 2*, *QEMU2* y *QEMU3*, según el mapa de direcciones IP del diseño VLAN mostrado en la figura 7.25.

Así, ejecutamos desde *equipo_real 2* (192.168.2.3):

```
C:\WINDOWS\system32\cmd.exe
Adaptador Ethernet Conexión de área local 2 :
  Sufijo de conexión específica DNS :
  Dirección IP . . . . . : 192.168.2.3
  Máscara de subred . . . . . : 255.255.255.0
  Puerta de enlace predeterminada : 192.168.2.1

C:\WINDOWS\system32>tracert 192.168.2.2
Traza a 192.168.2.2 sobre caminos de 30 saltos como máximo.
  1  30 ms    1 ms    1 ms  192.168.2.2
Traza completa.

C:\WINDOWS\system32>tracert 192.168.3.3
Traza a 192.168.3.3 sobre caminos de 30 saltos como máximo.
  1  37 ms    8 ms   11 ms  192.168.2.1
  2  60 ms   14 ms   21 ms  192.168.3.3
Traza completa.
```

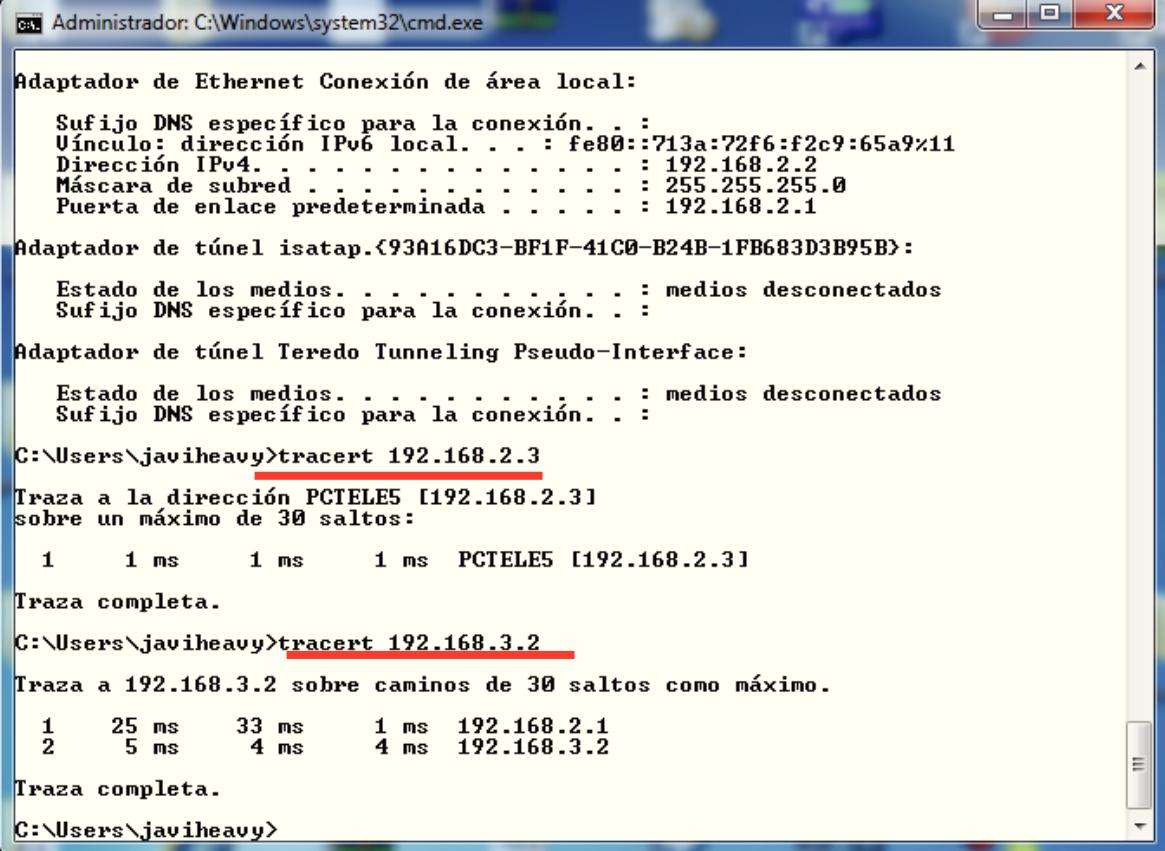
Figura 7.34. Conectividad y trayectoria en la conmutación VLAN
(en switch emulado, nodo 2)

En la figura 7.34 se observa como desde 192.168.2.3/24, para alcanzar el host 192.168.2.2/24 (equipo_real 2 → entorno emulado → equipo_real 1, diferente VLAN en distintos nodos virtuales) se realiza conmutación a nivel 2 (1 hop). Para alcanzar 192.168.3.3/24 (diferente VLAN en distinto nodo emulado), es necesario “alcanzar” primero el SVI configurado para su VLAN, y desde el interfaz virtual SVI alcanzar finalmente el host de destino final 192.168.3.3/24 (2 hops, conmutación a nivel 3). En ambos casos, se ha hecho uso del enlace troncal 802.1q configurado.

Hay que destacar que en las verificaciones anteriores, se ha realizado *interoperación* entre equipos reales y equipos emulados. No se pierde de vista este aspecto, uno de los objetivos analizados en el proyecto.

Completando el análisis y verificación de la conectividad VLAN, conviene apuntar que en el nodo *EtherSwitch router 1* el enlace troncal no se utilizará en el caso de querer establecer conmutación a nivel 3 entre **VLAN2 ↔ VLAN3**, pues ésta se realiza en el mismo *switch multilayer*, a través de los interfaces virtuales SVI.

Finalizando el análisis de la conectividad VLAN en el entorno emulado *GNS3/Dynamips*, verificamos la misma en el nodo *EtherSwitch router 1*:



```

Administrator: C:\Windows\system32\cmd.exe

Adaptador de Ethernet Conexión de área local:
  Sufijo DNS específico para la conexión. . . .
  Vínculo: dirección IPv6 local. . . . : fe80::713a:72f6:f2c9:65a9%11
  Dirección IPv4. . . . . : 192.168.2.2
  Máscara de subred . . . . . : 255.255.255.0
  Puerta de enlace predeterminada . . . . . : 192.168.2.1

Adaptador de túnel isatap.{93A16DC3-BF1F-41C0-B24B-1FB683D3B95B}:
  Estado de los medios. . . . . : medios desconectados
  Sufijo DNS específico para la conexión. . . .

Adaptador de túnel Teredo Tunneling Pseudo-Interface:
  Estado de los medios. . . . . : medios desconectados
  Sufijo DNS específico para la conexión. . . .

C:\Users\javiheavy>tracert 192.168.2.3
Traza a la dirección PCTELE5 [192.168.2.3]
sobre un máximo de 30 saltos:
  1      1 ms      1 ms      1 ms  PCTELE5 [192.168.2.3]
Traza completa.

C:\Users\javiheavy>tracert 192.168.3.2
Traza a 192.168.3.2 sobre caminos de 30 saltos como máximo.
  1    25 ms    33 ms      1 ms  192.168.2.1
  2     5 ms     4 ms      4 ms  192.168.3.2
Traza completa.

C:\Users\javiheavy>

```

Figura 7.35. Conectividad y trayectoria en la conmutación VLAN (en *switch multilayer 1* emulado)

La figura 7.35 nos muestra cómo para alcanzar **192.168.2.3/24** desde **192.168.2.2/24** (*equipo_real 1* → *entorno emulado* → *equipo_real 2*, misma VLAN en distintos nodos virtuales) se está realizando conmutación a nivel 2, utilizando para ello el enlace troncal (*1 hop*).

Para alcanzar **192.168.3.2/24** desde **192.168.2.2/24** (*equipo_real 1* → *host emulado QEMU*, distinta VLAN en el mismo nodo), se realiza conmutación a nivel 3 (*2 hops*) en el mismo *switch multilayer* como se indicó antes, sin hacer uso del enlace troncal.

Las verificaciones anteriores de la conectividad sobre este escenario segmentado VLAN, configurado en base a comandos *Cisco IOS* tal como se implementan sobre un escenario de red real similar, confirma la funcionalidad y rigor del emulador, así como la correspondencia funcional entre el entorno emulado y el real. Así mismo, recordemos que las verificaciones han sido realizadas vía *interoperación* entre ambos entornos de red, real y emulado, confirmándose así la funcionalidad de esta prestación del emulador *GNS3/Dynamips*, uno de los objetivos del proyecto.

7.2.7. Configuraciones de red adicionales para el análisis de prestaciones y rendimiento del emulador GNS3/Dynamips en la arquitectura LAN/VTP

Para configurar globalmente el escenario de trabajo de la *figura 7.25* que nos permite analizar los propósitos del capítulo 4, procedemos como se indica a continuación. Por comodidad y claridad en la exposición se muestra de nuevo la *figura 7.25*.

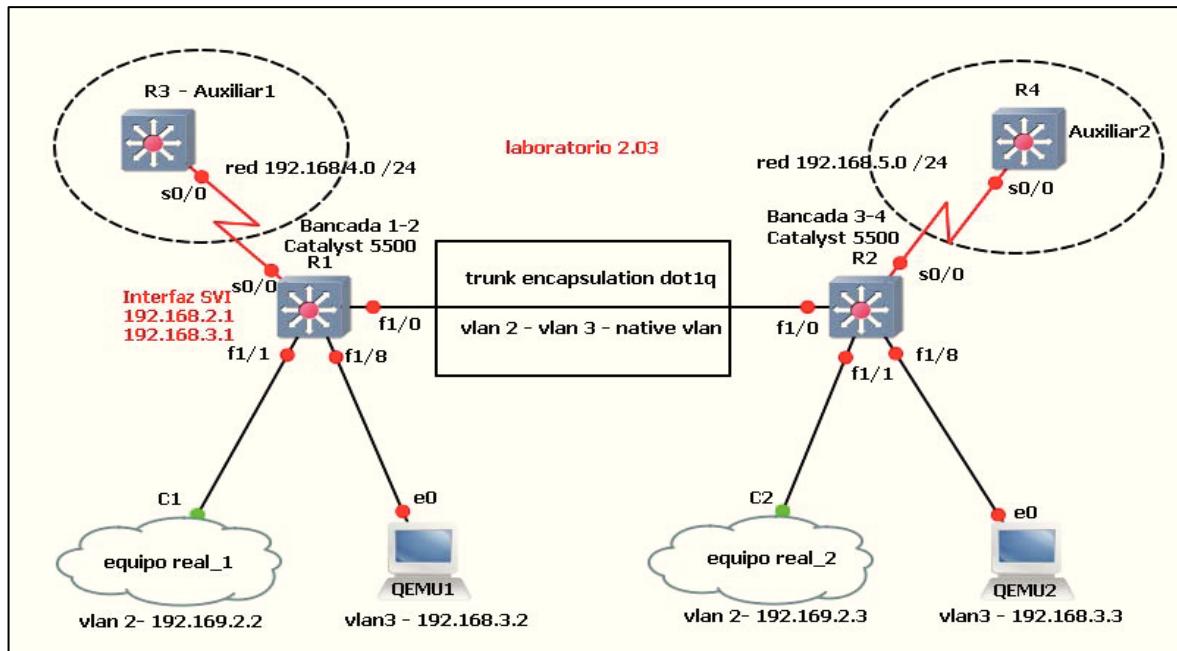
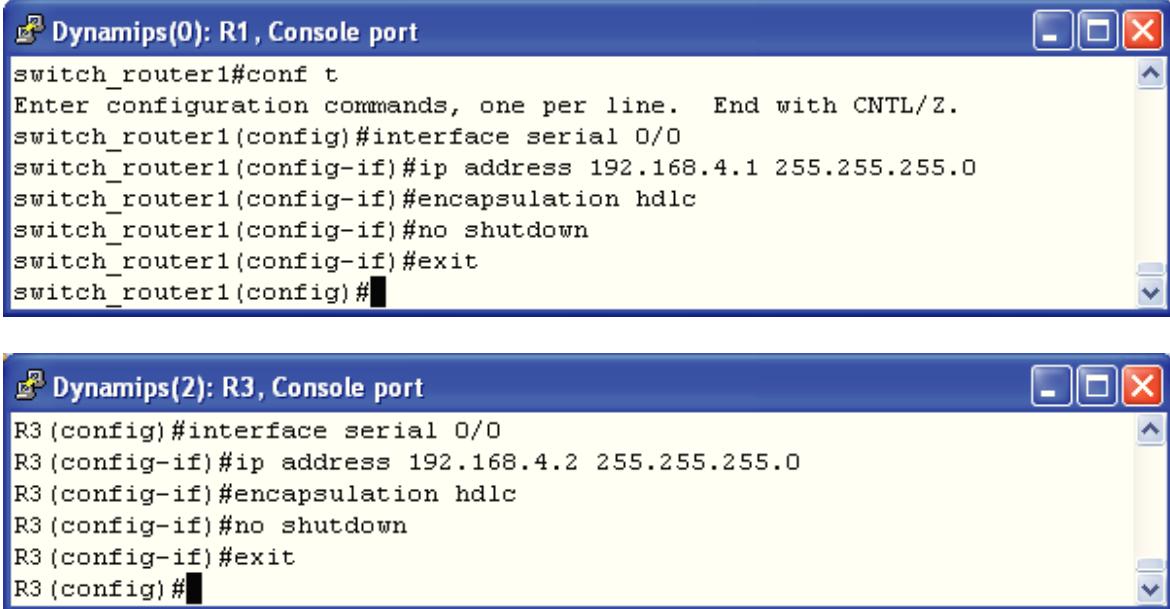


Figura 7.25. Topología/simulación gráfica en GNS3 de la arquitectura LAN /VTP.

Según esta topología, conectamos los nodos de red *EtherSwitch router* 1 y 2 de la arquitectura de red LAN Ethernet/VTP principal con sendos nodos *EtherSwitch router* (R3 y R4 respectivamente) mediante enlaces serie encapsulados HDLC, vía los interfaces ***serial 0/0*** disponibles en cada uno de los nodos emulados. Estos nuevos enlaces serie se adscriben administrativamente a las redes locales **192.168.4.0 / 24** y **192.168.5.0 / 24**, sin ningún propósito telemático y no participan del escenario comutado LAN Ethernet /VTP.

La configuración de los interfaces ***serial 0/0*** en cada de los nodos emulados (dirección IP y máscara de red, encapsulamiento HDLC y activación del interfaz) se realiza mediante los comandos *Cisco IOS* pertinentes, reflejado ello en las *figuras 7.36* y *7.37* de la página siguiente. Se realiza encapsulamiento HDLC para mostrar las posibilidades del entorno emulador.

- Nodos *EtherSwitch router 1 / R3* (red 192.168.4.0 / 24)



The figure displays two terminal windows side-by-side. Both windows have a title bar labeled "Dynamips(0): R1, Console port" and "Dynamips(2): R3, Console port" respectively. The windows show command-line interfaces for configuring serial interfaces.

```

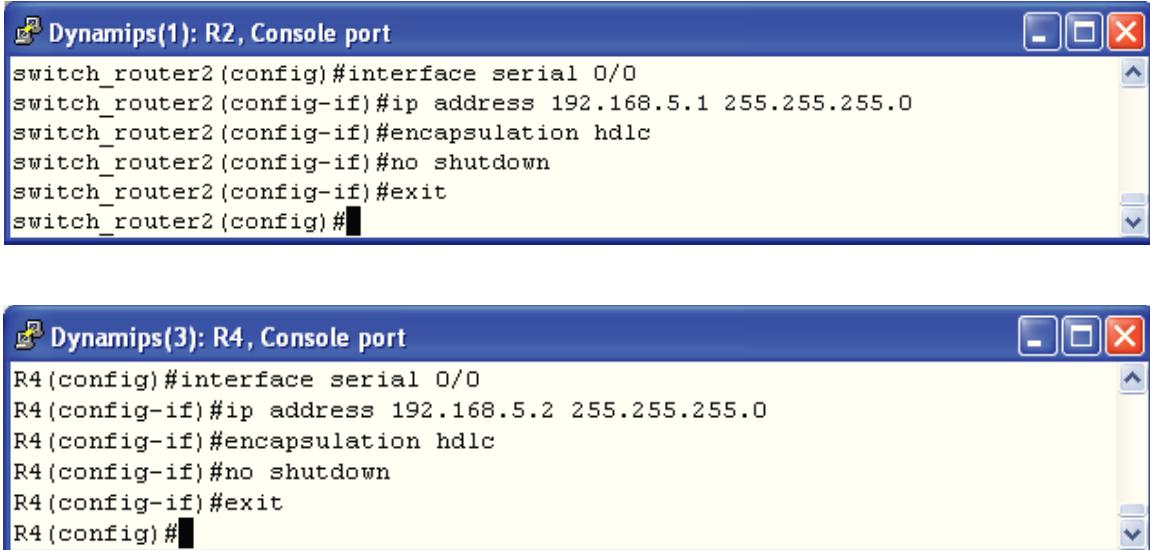
Dynamips(0): R1, Console port
switch_router1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch_router1(config)#interface serial 0/0
switch_router1(config-if)#ip address 192.168.4.1 255.255.255.0
switch_router1(config-if)#encapsulation hdlc
switch_router1(config-if)#no shutdown
switch_router1(config-if)#exit
switch_router1(config)#

Dynamips(2): R3, Console port
R3(config)#interface serial 0/0
R3(config-if)#ip address 192.168.4.2 255.255.255.0
R3(config-if)#encapsulation hdlc
R3(config-if)#no shutdown
R3(config-if)#exit
R3(config)#

```

Figura 7.36. Configuración de interfaces *serial 0/0* asignados a la red 192.168.4.0 / 24 en los nodos *EtherSwitch router 1* y R3 de la arquitectura de red LAN/VTP.

- Nodos *EtherSwitch router 2 / R4* (red 192.168.5.0 / 24)



The figure displays two terminal windows side-by-side. Both windows have a title bar labeled "Dynamips(1): R2, Console port" and "Dynamips(3): R4, Console port" respectively. The windows show command-line interfaces for configuring serial interfaces.

```

Dynamips(1): R2, Console port
switch_router2(config)#interface serial 0/0
switch_router2(config-if)#ip address 192.168.5.1 255.255.255.0
switch_router2(config-if)#encapsulation hdlc
switch_router2(config-if)#no shutdown
switch_router2(config-if)#exit
switch_router2(config)#

Dynamips(3): R4, Console port
R4(config)#interface serial 0/0
R4(config-if)#ip address 192.168.5.2 255.255.255.0
R4(config-if)#encapsulation hdlc
R4(config-if)#no shutdown
R4(config-if)#exit
R4(config)#

```

Figura 7.37. Configuración de interfaces *serial 0/0* asignados a la red 192.168.5.0 / 24 en los nodos *EtherSwitch router 2* y R4 de la arquitectura de red LAN/VTP.

7.2.8. Complemento teórico-práctico a las conclusiones del proyecto.

Explicitando más profundamente las conclusiones de la *tabla 5.1* del capítulo 5 del proyecto, en cuanto a la funcionalidad, rigor y correspondencia funcional del entorno emulador es destacable que las configuraciones de red se implementan en base a idénticos comandos *Cisco IOS* que serían ejecutados en análogos entornos reales de trabajo, sin ninguna cortapisa tecnológica y funcional por parte del software emulador. Las configuraciones de red para las arquitecturas WAN ppp y LAN Ethernet/VTP y los diseños propuestos según las *figuras 3.1* y *3.3* son convenientemente descritas y analizadas en el anexo 2. Una vez implementadas, se verifica su funcionalidad (en este mismo anexo) siendo los resultados acordes con los objetivos de diseño: funcionalmente rigurosos e idénticos a los que se obtendrían sobre entornos reales análogos. Recordemos que este entorno software emulador debe principalmente sus prestaciones y potencia de emulación al empleo en la misma de idénticos sistemas *Cisco IOS* que integran dispositivos de red reales. Derivado de esto, ha surgido la necesidad de elaborar el anexo 3 sobre la extracción de los sistemas *Cisco IOS* de dispositivos de red reales y su posterior integración en el entorno software *GNS3/Dynamips*.

A la hora de implementar las configuraciones en *GNS3/Dynamips*, se presenta al usuario idéntico interfaz gráfico que visualizaría en caso de configurar el dispositivo de red real *vía* puerto consola y cable de acceso *rollover*, siendo ésta otra ventaja con respecto a otras plataformas de emulación. Esta característica permite, llegado el momento, una cierta “aclimatación funcional” a un determinado entorno real de trabajo que debamos analizar en el emulador previamente a acometer una actuación telemática sobre él. Tal como se menciona en el capítulo 1, uno de los objetivos a evaluar en el proyecto es el uso potencial del entorno emulador *GNS3/Dynamips* en entornos profesionales, como herramienta software que permita un análisis previo de un escenario real de trabajo sobre el que se planea acometer una determinada actuación, con el objetivo de minimizar riesgos estructurales y costes asociados al proceso. Por ello, todas las herramientas y prestaciones que permitan reproducir adecuadamente un escenario real de trabajo sobre uno emulado es un aporte sustancialmente beneficioso. Además, *GNS3/Dynamips* permite implementar topologías/arquitecturas en base a dispositivos de red genéricos (*switch Ethernet*, *switch Frame Relay*, *switch ATM...*), dispositivos no asociados a ningún modelo corporativo concreto, haciendo interesante y recomendable su uso en ámbitos académicos.

La *tabla 7.2* de la página siguiente nos ofrece a modo de resumen las principales ventajas e inconvenientes que se extraen de los análisis realizados en los capítulos 3 y 4 del proyecto.

Tabla 7.2	Ventajas / inconvenientes del entorno emulador GNS3/Dynamips (extraídas del análisis del capítulo 3↔ anexo 2 y del análisis de interoperabilidad del capítulo 4)
Ventajas	<ul style="list-style-type: none"> • <u>Objetivo fundamental del proyecto</u>: permite la correcta <i>interoperación</i> funcional a nivel 2 y 3 de tráfico y equipos de red reales y emulados. Estricta correspondencia funcional del tráfico de red implicado en dicha <i>interoperación</i>. • <u>Facilidad</u> de implementación y uso que lo hace recomendable para ámbitos profesionales y académicos. <u>Adeuada usabilidad</u>. • Configuración de escenarios de trabajo emulado conforme a escenarios reales análogos, con las consiguientes ventajas para el campo profesional en cuanto a la adquisición de habilidades sobre los escenarios reales, minimización de riesgos y costes. Correspondencia funcional entre entornos reales y emulados. • Implementación de topologías en base a dispositivos de red generales no asociados a dispositivos hardware concretos que incrementa las posibilidades y <u>versatilidad del diseño</u> en entornos académicos, así como minimizar consumo de recursos. • Permite <u>segmentar</u> topologías extensas de trabajo en una “parte real” y una “parte emulada”, vía interfaz <i>Ethernet</i>, en el caso de disponer recursos hardware limitados, y si procede en el diseño.
Inconvenientes	<ul style="list-style-type: none"> • Actualmente no permite emular estrictamente dispositivos de red de la familia <i>Catalyst</i> de <i>Cisco Systems</i>, debiendo hacer uso de ciertos recursos de diseño para conformar dispositivos de red emulados similares. • Los dispositivos de red emulados se circunscriben actualmente a los entornos corporativos <i>Cisco Systems</i> y <i>Juniper</i>. • No permite emular sistemas operativos <i>Cisco CatOS</i> (ni es previsible que lo haga). • Actualmente, sólo se implementa <i>interoperación entorno real ↔ entorno emulado</i> a través de interfaz <i>Ethernet</i>. • Deficiente comportamiento de la latencia en la <i>interoperación entorno real ↔ entorno emulado</i> para algunas arquitecturas de red, como es el caso de la <i>interoperación</i> tráfico real <i>Ethernet</i> ↔ <i>interfaz virtual SVI</i> (comunicación a nivel 3).

Tabla 7.2. Tabla resumen de ventajas e inconvenientes del emulador *GNS3/Dynamips* extraídas del análisis de los capítulos 3↔anexo 2, y del análisis de la *interoperabilidad* del capítulo 4.

Complemento a las conclusiones de los análisis de la interoperación entorno real ↔ entorno emulado.

Ahondando en las conclusiones del objetivo prioritario del proyecto, se analizan exhaustivamente los resultados de la *interoperación entorno real ↔ entorno emulado*, haciendo referencia a los *apartados 3.1.2, 3.2.2*. En ellos se realiza captura y análisis del tráfico de red emulado sobre escenarios en los que se implementa una *interoperación* entre entornos de red real y emulado. Siendo así, se aprecia cómo disponemos de idéntica información a la que obtendríamos sobre análogos escenarios reales de red (existe estricta correspondencia funcional). Esta información está relacionada con los protocolos configurados y activos, caso del *apartado 3.1.2*. En el caso del *apartado 3.2.2* (y 7.2.2 de este anexo 2), el tráfico de red capturado aporta, además, información significativa sobre la *interoperación* entre los entornos de red real y emulado, información ésta que permite verificar la correspondencia funcional entre ambos entornos. Concretamente, se verifica que la estructura de las tramas del tráfico de red emulado es idéntica a la teórica, y en definitiva, a la del tráfico de red real. Esto constata empírica y definitivamente que el tráfico de red en *GNS3/Dynamips*, entre dispositivos emulados, y en el caso de escenarios en los que *interoperan* equipos reales y emulados guarda correspondencia funcional con el tráfico de red real.

En el *apartado 7.2.4* de este anexo 2, se ha realizado un análisis mediante *Wireshark* del tráfico de red existente en el enlace *equipo real 1 ↔ nodo virtual 1*, para distintas longitudes de trama *Ethernet*. En este enlace se procede a la *interoperación* entre el entorno real y el emulado. Siendo así, se muestra y analiza el comportamiento telemático de la interoperación al respecto. Los resultados de esta captura muestran cómo existe una absoluta correspondencia funcional entre el tráfico de red real y emulado a “ambos lados” del interfaz *Ethernet*, conservándose la estructura y longitud de trama. Consecuentemente, no sólo es factible la *interoperación* entre entornos de red reales y emulados en *GNS3/Dynamips*, sino que el rigor estructural del tráfico de red en la misma es máximo.

Complemento a las conclusiones del proyecto sobre la latencia en la interoperación.

1. Complemento al análisis de resultados de la arquitectura de red WAN ppp.

Para el primero de los experimentos propuestos en el análisis de la latencia, las *tablas 4.4 y 4.5* obtenidas en el *apartado 4.2.2* del capítulo 4, revelan un valor de la latencia de red para todas las longitudes de trama *Ethernet* muy elevado con respecto al valor que se obtiene en un entorno real análogo. Este valor no sería asumible en la práctica en un escenario de red real similar. Recordemos que en este primer ensayo implementábamos peticiones de eco (*ping*) desde un *equipo real externo* al emulador (**192.168.2.2**) con destino el interfaz virtual SVI (**192.168.2.1**), interfaz configurado en el nodo *multilayer Etherswitch router* (esto es, comutación a nivel 3). A la vista de los resultados, y para esta arquitectura de red, el valor de la latencia en la *interoperación entorno real ↔ entorno*

emulado es considerablemente deficiente debiendo considerarse indefectiblemente en los posibles análisis que pudiésemos efectuar en el emulador sobre la prestación de servicios telemáticos en esta arquitectura. Así, el elevado valor de la latencia en este caso sugiere un mal comportamiento por parte del emulador en la *interoperación* en el caso de conmutar tráfico de red real *Ethernet* con destino un interfaz virtual SVI (comutación a nivel 3).

Aunque estrictamente no es el caso, esta elevada latencia sugiere el hecho de que el potencial valor de la latencia media en una supuesta conexión *extremo-extremo* en topologías emuladas más complejas (mayor número de nodos intermedios o *hops* en la trayectoria) incumpliese la recomendación ITU-T G.114 al respecto (valor máximo < 150 ms) [19] a la hora de implementar servicios telemáticos, como VoIP o vídeo multimedia. Estos valores de latencia se han obtenido en una situación de consumo máximo de recursos, siendo igualmente deficientes los valores obtenidos en la situación estable de trabajo asociada al consumo mínimo de recursos. A la hora de extraer conclusiones y paralelismos en el estudio de un escenario de red real reproduciendo éste sobre el emulador *GNS3/Dynamips*, podríamos considerar como una cota superior amplia para la latencia media real el valor de latencia mínimo obtenido en cada experimento emulado concreto (para cada longitud de trama y consumo de recursos).

Por ejemplo, en el supuesto práctico de la *configuración hardware 1*, longitud de trama Ethernet 802.3 de 1514 bytes y 2 *switch-router* activos, el valor medio de la latencia es de 43 ms (*tabla 4.4*, muy elevado en este caso). Pero en el cálculo de este valor medio, ponderación de los valores de las 200 peticiones de eco para las 20 repeticiones del experimento, intervienen varios valores, siendo el menor de ellos 2 ms (los valores mínimos, máximos y medios de la latencia son reportados al finalizar la ejecución del comando *ping*). Empíricamente se comprueba que la latencia media medida sobre el mismo escenario de red real queda siempre por debajo de este valor, y en alguna de las 200 repeticiones del ensayo (1 – 2 iteraciones) lo alcanza. Por lo tanto, podríamos considerar como una cota superior (suficientemente amplia) para la latencia media en el análogo escenario de red real el valor mínimo de la latencia medido en el escenario emulado, valor que depende para cada supuesto práctico. Esto nos permitiría trasladar desde el entorno emulado al escenario de red real, y para cada supuesto práctico, un margen de trabajo considerablemente fiable para la latencia.

De los resultados de las *tablas 4.4* y *4.5* del capítulo 4, extraídos en condiciones de consumo máximo de recursos, extraemos otras conclusiones decisivas asociadas al objetivo principal del proyecto. La primera de ellas es que el valor de la latencia en la *interoperación entorno real ↔ entorno emulado* es muy dependiente del número de nodos emulados, y por lo tanto, del consumo de recursos en el sistema emulador. Ello nos conduce a la decisiva conclusión de que el comportamiento en tiempo real del emulador puede estar severamente condicionado por la arquitectura de red y su topología asociada (principalmente, el nº de nodos emulados de la arquitectura).



Figura 5.2. Comportamiento del emulador para la latencia de red respecto al consumo de recursos y la longitud de trama para las *tablas 4.4 , 4.5, 4.11 y 4.12* referentes al primero de los experimentos de la latencia en mabas arquitecturas de red.

Del análisis conjunto de estas últimas valoraciones de las *tablas 4.4 y 4.5* se observa que la dependencia del comportamiento del emulador para la latencia es más crítica y sensible en el sentido de las columnas y hacia la derecha, que en el de las filas y en sentido descendente. De un modo gráfico, esto se muestra en la *figura 5.2* anterior: el comportamiento del emulador para la latencia de red en este experimento y para esta arquitectura de red (y consecuentemente trasladable en cierta manera a su comportamiento en tiempo real) resulta más sensible al consumo de recursos del sistema emulador que al tamaño de trama *Ethernet*.

2. Complemento al análisis de resultados para la arquitectura LAN Ethernet / VTP

Los resultados de las *tablas 4.11 y 4.12* del *apartado 4.2.4* del capítulo 4 son muy similares a los obtenidos en el mismo experimento sobre la arquitectura WAN ppp, y en la situación de consumo máximo de recursos. Con respecto a estos últimos, se observa un ligero incremento en el valor de la latencia de red, incremento que podemos asociar al ligero aumento del consumo de recursos en el emulador para la arquitectura LAN Ethernet/VTP.

No obstante, tanto estos valores como los referentes al consumo de recursos del sistema emulador podrían equiparse llegado el caso para ambas arquitecturas, pues las diferencias son muy poco significativas. Además, y al igual que en la arquitectura WAN ppp, la dependencia del comportamiento del emulador respecto de la latencia es más sensible y crítica en el sentido de las columnas y hacia la derecha, que en el de las filas y en sentido descendente. Por consiguiente, el comportamiento del emulador para la latencia de red (y consecuentemente trasladable a su comportamiento en tiempo real) es más

sensible al consumo de recursos del sistema emulador que a la longitud de trama. Igualmente es reseñable, que la diferencia en los valores de la latencia para situaciones de consumo máximo de recursos y consumo mínimo presenta la misma dependencia antes descrita, apreciándose un mayor efecto en la optimización de recursos del sistema en el sentido de las columnas hacia la derecha, es decir, el efecto de tales medidas es mayormente contrastable en situaciones de mayor estrés del sistema, llegando a ser fundamentales en tal caso para el correcto desempeño de los procesos de la emulación.

Para el segundo de los experimentos, las *tablas 4.13 y 4.14* muestran un descenso rotundo y constatable en sus valores con respecto al mismo experimento en la arquitectura WAN ppp. En este caso, los resultados son bastante más equiparables a los obtenidos sobre un mismo escenario real, apreciándose igualmente que estos empeoran conforme aumentamos el consumo de recursos y el tamaño de la trama. Es decir, la dependencia en los valores obtenidos en este experimento es la misma que antes, pero los resultados en la latencia media han mejorado manifiestamente. Sobre el escenario emulado LAN Ethernet/VTP comutado, y para este experimento, estamos implementando comutación a nivel 2 entre ambos equipos, equipos adscritos a la misma VLAN, y el emulador presenta un mejor comportamiento en esta *interoperación*. Siendo así, el funcionamiento del emulador en tiempo real parece ser bastante correcto en este caso, con ciertos márgenes de precaución. Los valores de latencia media obtenidos, aunque superiores a los obtenidos sobre idénticos escenarios reales (aproximadamente entre 1-2 ms), se circunscriben mejor a la recomendación ITU-T G.114 [19], de una manera absoluta, pues su valor está ampliamente dentro de los márgenes de seguridad establecidos, y de una manera relativa, pues la conexión *extremo-extremo* implica dos nodos de red tan sólo, en este caso 2 *switch* (un solo *hop*).

7.3. Anexo 3: Extracción de sistemas operativos *Cisco IOS* de hardware de red *Cisco Systems* y posterior integración en *GNS3/Dynamips*. Configuración básica de *GNS3/Dynamips*: parámetros y prestaciones fundamentales.

7.3.1. Introducción.

La extracción y copia del sistema operativo *Cisco IOS* de un dispositivo de red *Cisco Systems* es un proceso ciertamente recomendable (evidentemente, se hace extensible esta recomendación a todo dispositivo hardware en general) a la hora de planificar un cierto mantenimiento para asegurar un correcto funcionamiento futuro de nuestro hardware: no son pocas las ocasiones en las que por error técnico y/o humano es necesario reinstalar el software en un máquina o acometer un proceso de actualización con el fin de implementar mejoras y nuevas funcionalidades. Añadido a esto, y como se describe a lo largo de la memoria del proyecto, los dispositivos emulados en el entorno de emulación *GNS3/Dynamips* operan con el mismo sistema que dispositivos de red reales análogos, por lo que para emular un determinado hardware de red es necesario, tras la extracción del sistema operativo, su posterior integración en el emulador. Por ello, el presente anexo pretende habilitar en las tareas antes descritas y constituyó en su momento una tarea decisiva y fundamental para la elaboración del proyecto.

7.3.2. Extracción del sistema *Cisco IOS* de dispositivos de red *Cisco Systems*.

Para extraer el sistema operativo de un dispositivo de red *Cisco Systems* de un modo general, operamos de la siguiente forma:

- Primeramente, sobre un PC se configura un servidor TFTP. Para ello, en el desarrollo del proyecto se ha hecho uso del software *Filezilla Server* [16], catalogado *open source software*.
- Con el objetivo de “volcar” posteriormente el sistema *Cisco IOS* sobre nuestro servidor TFTP, debemos configurar una sencilla conexión LAN *Ethernet* entre el dispositivo de red y el servidor TFTP anterior. Para ello, y de un modo muy general, asignamos por ejemplo al servidor la dirección IP **192.168.1.2 /24**. Con esto, ya tenemos configurado un extremo de la conexión *Ethernet* entre ambos dispositivos.
- Por medio de un cable *rollover* o consola (*figura 7.38* siguiente), y tal como se describe en el *apartado 7.1.3* del anexo 1, se accede al puerto consola del dispositivo de red haciendo uso de la configuración de emulador de Terminal de Windows de la *figura 7.39*. La configuración del cableado debería ser así para la mayoría de dispositivos *Cisco Systems* actuales, aunque como se muestra en el *apartado 7.1.3*, el dispositivo de red *Cisco Catalyst 5500 Switch* precisaba de una conexión de acceso diferente: un cable consola **Db-9→RJ-45** directo.



Figura 7.38. Cable *rollover* o consola .



Figura 7.39. Configuración de *HyperTerminal* para la conexión al puerto consola del dispositivo de red.

- Una vez se accede a la consola del dispositivo de red se configura el otro extremo de la conexión LAN *Ethernet* anterior, el correspondiente al dispositivo de red. Para ello, operamos de un modo general con comandos *Cisco IOS* :

```

router> enable

router# configure terminal ( ó simplemente conf t)

router(config)# interface fastethernet 1/1
router(config-if)# ip address 192.168.1.1 255.255.255.0
router(config-if)# no shutdown

```

Con la configuración anterior, en base a comandos Cisco *IOS*, y tras acceder al *modo de configuración global* → *de interfaz*, se configura la dirección IP **192.168.1.1 / 24** sobre el interfaz FastEthernet 1 dispuesto en el slot 1 (atención, dirección de red e interfaz para este caso concreto; en la práctica deben indicarse el número de interfaz y *slot/bahía* para cada caso particular, según la disposición hardware del dispositivo).

- A continuación, conectamos dispositivo de red y servidor TFTP por medio de un cable de red *Ethernet* (RJ-45) directo:

interfaz FastEthernet 1/1 en disp. de red ↔ *interfaz Ethernet en PC*

- Para conocer la versión (“nombre”) del sistema *Cisco IOS* a copiar ejecutamos, desde el *modo de operación EXEC privilegiado*, el siguiente comando :

```
router# show version
```

- Finalmente, procedemos al volcado del sistema operativo sobre nuestro servidor TFTP, tal como se indica secuencialmente :

```

router# copy flash: nombre_ios tftp
Remote host []? 192.168.1.2
Name of configuration file to write [nombre_ios]? Enter

```

De un modo general, se ha hecho constar en las instrucciones anteriores que el sistema *Cisco IOS* se encuentra normalmente alojado en la memoria *Flash* que los dispositivos *Cisco Systems* llevan a tal efecto.

7.3.3. Interoperación de dispositivos de red reales y emulados en GNS3/Dynamips.

Una de las prestaciones que confieren al entorno de emulación *GNS3/Dynamips* un atractivo especial frente a otras plataformas de emulación es la capacidad de *interoperar* tráfico de red y equipos reales con tráfico de red y equipos emulados. Éste, recordemos, es uno de los objetivos principales de análisis del proyecto.

Para proceder a la interoperación telemática: *entorno de red real* \leftrightarrow *entorno de red emulado*, el sistema emulador, generalmente un PC, debe disponer de tantos interfaces *Ethernet* libres como nodos o dispositivos reales se pretenda integrar, realizándose a través de dichos interfaces la *integración/interoperación* de ambos entornos (*figura 7.40*).

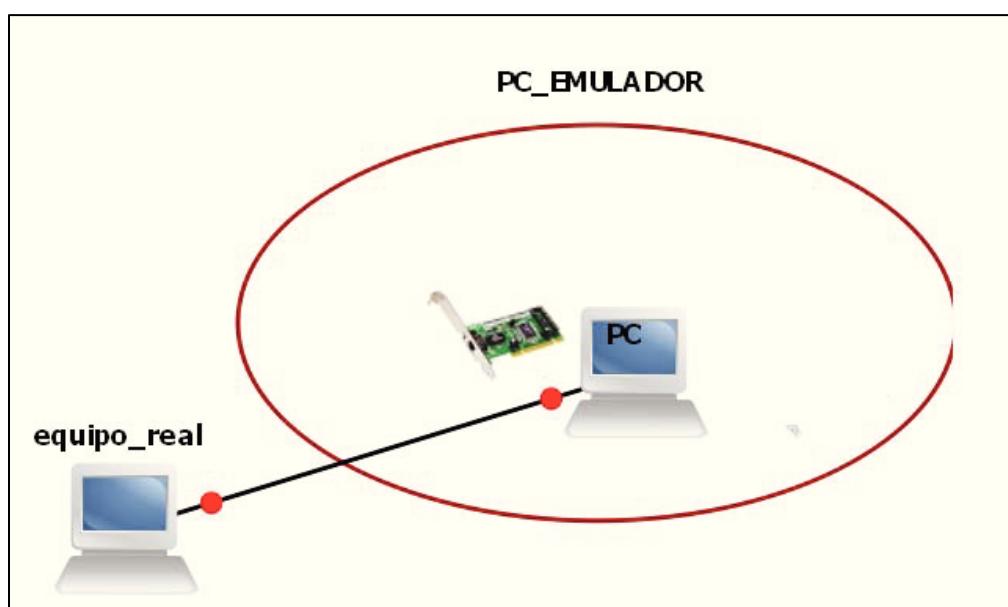


Figura 7.40. Esquema general para la integración de un equipo real en el entorno de emulación *GNS /Dynamips*, vía interfaz *Ethernet*.

Al respecto, es necesario concretar lo siguiente:

- Si el dispositivo real a integrar es un *host* (PC), para su integración se utilizará cableado de red *Ethernet RJ-45 directo*.
- Así mismo, si el dispositivo real a integrar es un *switch/router*, para su integración se utilizará cableado de red *Ethernet RJ-45 cruzado*.

Añadido a esto, se deben configurar los interfaces *Ethernet* del PC emulador empleados en la *interoperación* sin ninguna dirección de red, como si presentasen configuración DHCP.

Una vez se ha procedido con la “parte física” de la *interoperación*, desde el entorno de *GNS3/Dynamips* se procede a la configuración software de la misma. Pero antes de proseguir, hacemos un breve inciso.

Para la realización del presente proyecto se ha hecho uso de la versión *software 0.7.3* de GNS3 [12]. Si bien existen versiones más actualizadas del mismo, tras una primera toma de contacto y experimentación inicial, presentaron ciertas inestabilidades en su comportamiento por lo que se decidió trabajar con la versión antes mencionada, que tras su análisis presentó un comportamiento más estable. Tras una sencilla y convencional instalación debe procederse con la configuración básica y fundamental del que será nuestro entorno de trabajo emulado (hacemos referencia al entorno Windows, la instalación respectiva en entorno Linux se detalla en el anexo 4). Para ello, al hacer uso por primera vez del software, se ofrece un contexto gráfico para establecer las configuraciones prioritarias del emulador, reflejado en la *figura 7.41*.

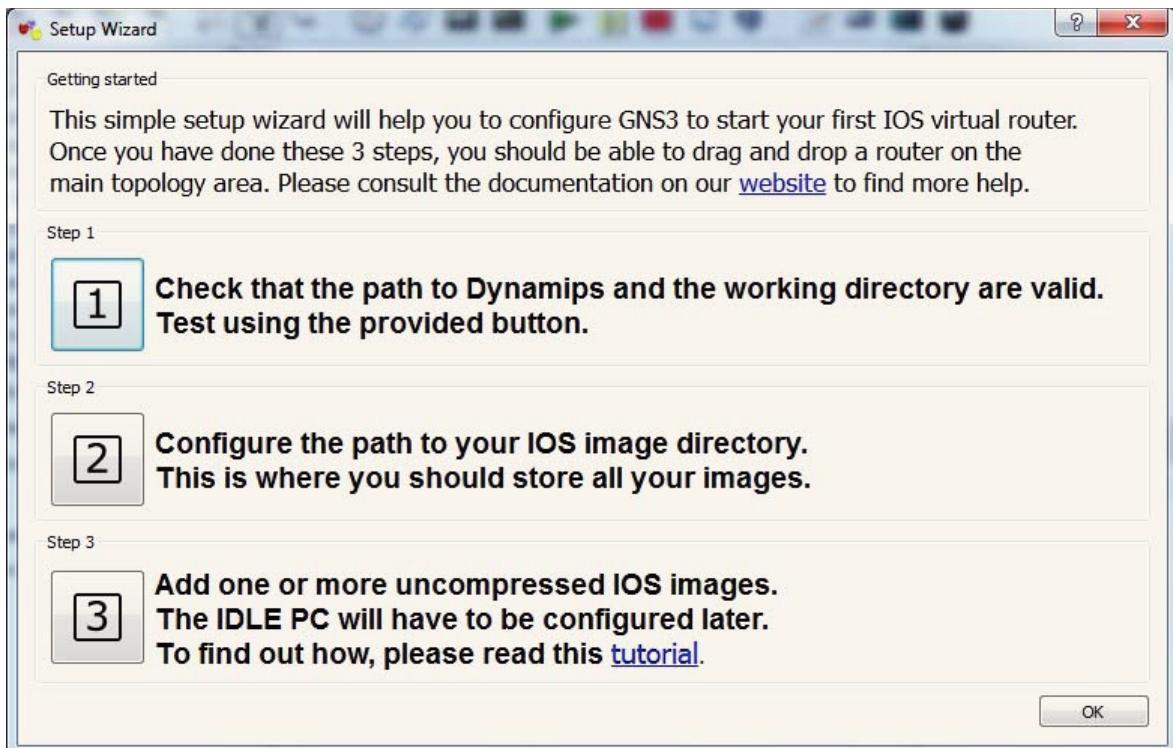


Figura 7.41. Contexto gráfico inicial para la configuración de los parámetros básicos y fundamentales de *GNS /Dynamips*.

Este menú inicial de configuración puede obviarse y optar por ejecutar manualmente la configuración de los parámetros y variables fundamentales del emulador. La configuración y descripción de estos parámetros básicos se aborda en apartados posteriores. Además, tal como indica la *figura 7.41*, el tercero de ellos debe ser configurado posteriormente, como se describe en el apartado 7.5.3 de este anexo 3.

Tras este inciso, continuamos con la configuración de la *interoperación entorno real ↔ entorno emulado*. Una vez nos encontramos en el interfaz gráfico principal de GNS3, seleccionamos del menú lateral izquierdo un elemento “*cloud*”, cuya función es representar (y obviamente integrar) dispositivos reales en el entorno emulado. Seguidamente, haciendo “*clic*” con el botón derecho del ratón sobre dicho elemento, seleccionamos del menú desplegable la opción **Configure**, según muestra la figura 7.42.

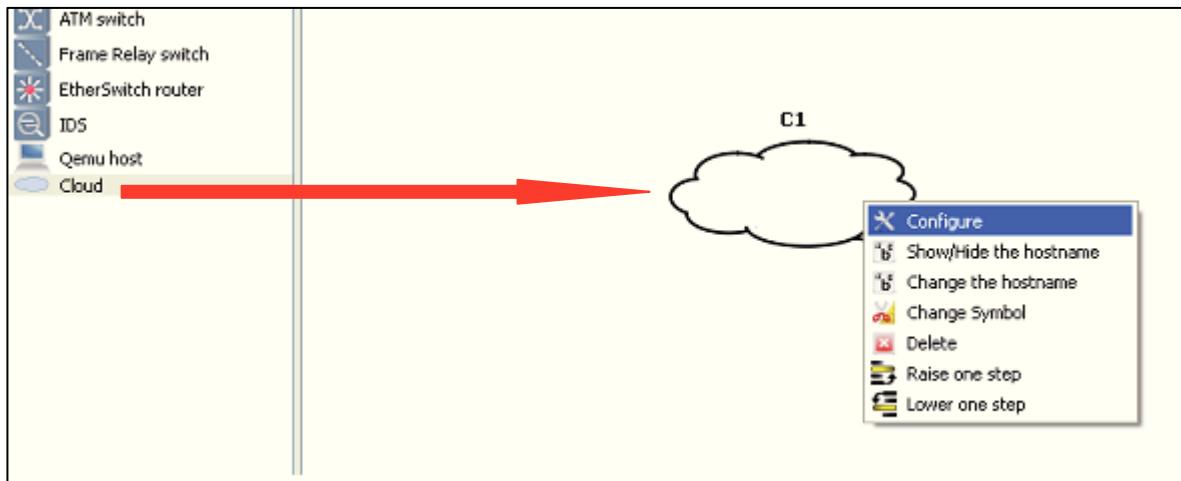


Figura 7.42. Contexto gráfico para la integración de dispositivos reales en *GNS3 / Dynamips*.

Finalmente, y sobre el nuevo contexto gráfico emergente de la figura 7.43 siguiente,

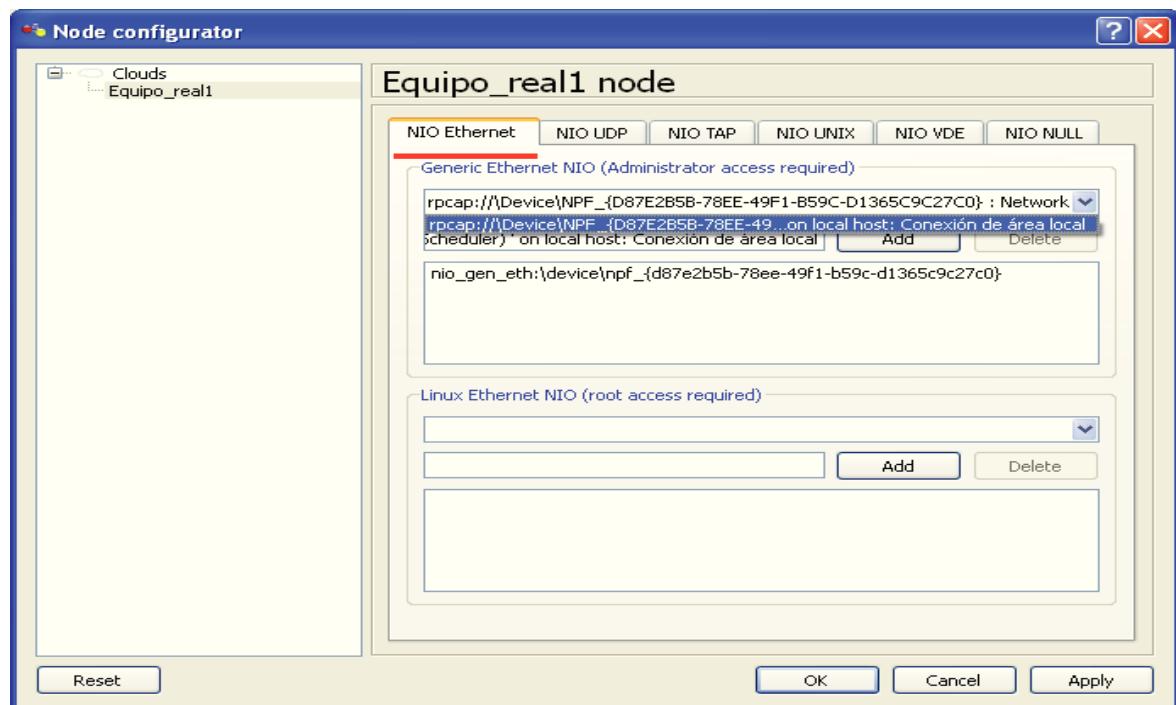


Figura 7.43. Contexto gráfico de configuración en la selección del interfaz de integración.

seleccionamos la *pestana* → **NIO Ethernet** ; esta *pestana*, a través de un nuevo menú *desplegable*, nos permite seleccionar de una lista de interfaces disponibles en el PC emulador el interfaz *Ethernet* que deseemos para integrar el dispositivo de red real (para ello, seleccionaremos el interfaz deseado de la lista y a continuación → **Add**)

Toda la configuración anterior nos permite disponer de un dispositivo de red real (*host, switch o router*) con capacidad de *interoperar* con un escenario telemático emulado en *GNS3/Dynamips*. Finalmente, solo resta la conexión en el interfaz gráfico de GNS3, *vía Ethernet*, del elemento “*cloud*” con el dispositivo de red emulado que deseemos.

7.3.4. Integración de *Cisco IOS* en el entorno emulador *GNS3/Dynamips*.

Tal como se ha comentado anteriormente, una vez extraído el sistema operativo del dispositivo de red, y con la finalidad de proceder a su emulación en un determinado escenario de trabajo en *GNS3/Dynamips*, debemos proceder a integrarlo/incorporarlo en el entorno software. Para ello, procedemos como se indica a continuación.

Primeramente, una vez realizada la instalación de *GNS3/Dynamips* es prioritario y fundamental configurar el *path* o ruta indicando los directorios donde se ubicarán nuestros proyectos de trabajo y las imágenes a emular por los dispositivos, esto es, los sistemas *Cisco IOS*. Este es uno de los parámetros básicos de configuración antes mencionados. La personalización al respecto es libre, pero sirva como ejemplo ilustrativo la opción de trabajo mostrada en la *figura 7.44*.

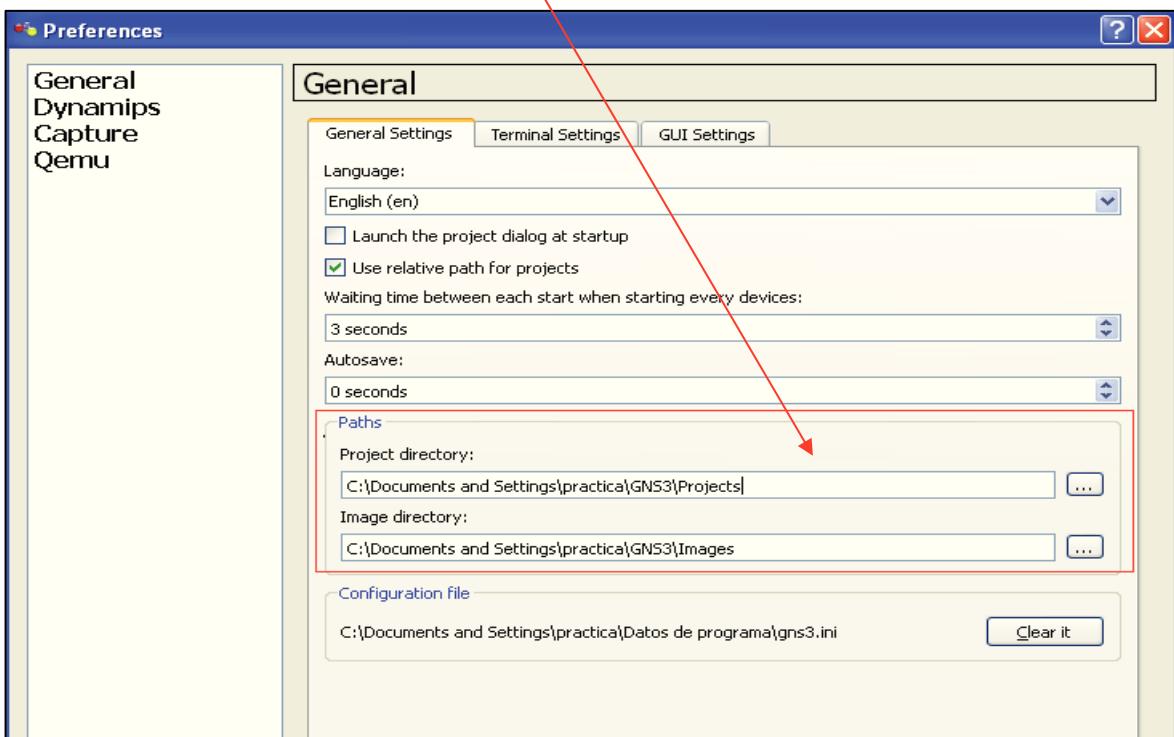


Figura 7.44. Contexto gráfico para la configuración del *path* o ruta de los directorios.

Seguidamente, se accede al menú de configuración: **Edit** → **Preferences** → **General**, desde el interfaz gráfico principal de *GNS3*, como muestra la figura 7.45.

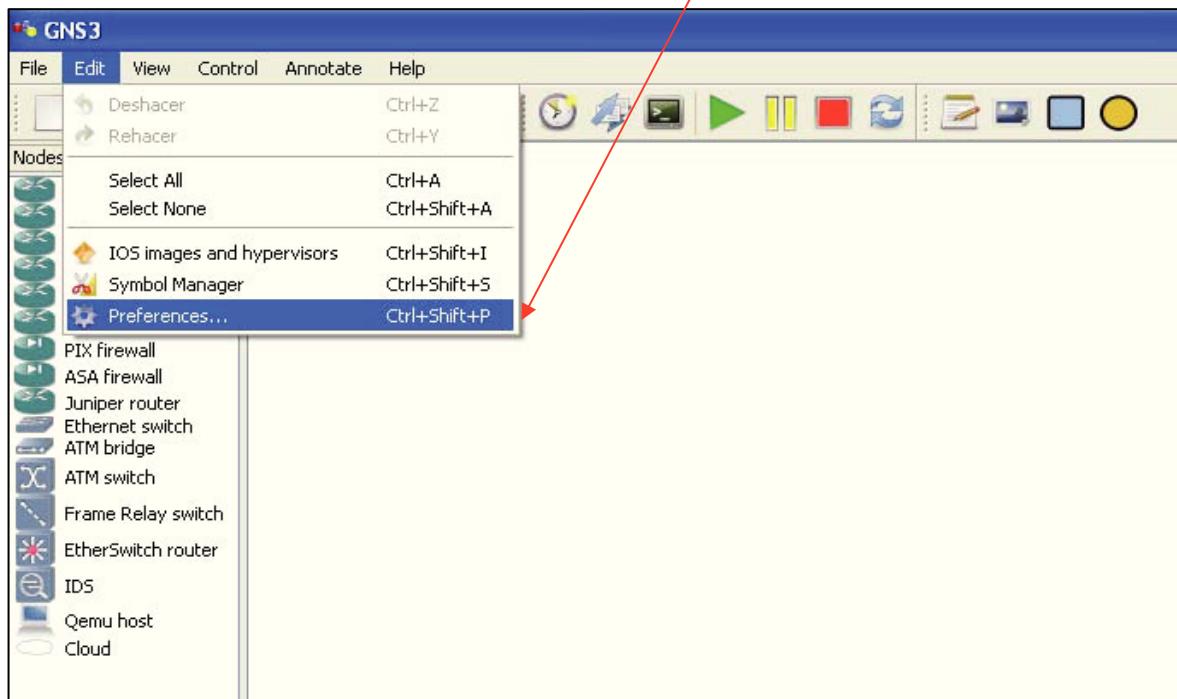


Figura 7.45. Acceso al menú de configuración de *GNS3/Dynamips*.

Los sistemas *Cisco IOS* (recordemos, extraídos de dispositivos de red reales) los copiaremos en el directorio destinado a las imágenes (directorio *images*, en el caso anterior de la figura 7.44), y los proyectos de trabajo realizados o activos se guardarán en el directorio configurado a tal efecto (directorio *projects* de la figura 7.44).

Finalmente, precisamos asociar dispositivos emulados con imágenes o sistemas *Cisco IOS*. Para ello, desde el interfaz gráfico principal de *GNS3*: **Edit** → **IOS images and hypervisors**, nos dirige a un contexto gráfico de configuración en el que únicamente resta, y según nuestras particularidades de trabajo y emulación, seleccionar las opciones que nos ofrecen los menús *desplegables* (esto es, los modelos de dispositivos e imágenes asociadas).

Recomendación práctica: los sistemas o imágenes *Cisco IOS* en su formato original, extraídos de los dispositivos de red, presentan formato *nombre_imagen.bin*. Cada vez que ponemos en funcionamiento en el emulador un dispositivo asociado con una imagen, ésta debe descomprimirse, con el tiempo que ello implica. Podemos agilizar este proceso descomprimiendo *nombre_imagen.bin* con el software *Winrar* por ejemplo, en el mismo directorio donde tenemos ubicadas las imágenes *Cisco IOS*, cambiar el nombre del archivo generado/descomprimido al formato *nombre_imagen.image* y reasignar esta nueva imagen a nuestro dispositivo, según indicaba el párrafo anterior. Con ello, cada vez que iniciemos el dispositivo emulado ahorraremos el tiempo empleado en la descompresión de la imagen *Cisco IOS* asociada al mismo.

Un parámetro de configuración fundamental cuyo funcionamiento es preciso verificar la primera vez que vayamos a hacer uso de *GNS3/Dynamips* (y configurar adecuadamente si es necesario) es el *path* o *ruta* seguida en la ejecución del proceso *dynamips.exe* (proceso sobre el que recaen fundamentalmente las tareas de emulación) y su directorio de trabajo. Este es uno de los parámetros mostrados con anterioridad en la figura 7.41. Para ello, desde el interfaz gráfico principal de GNS3: **Edit** → **Preferences** → **General** → **Dynamips**, muestra el contexto gráfico sobre el que se realiza esta verificación (figura 7.46). El resultado de la misma es satisfactorio si así se indica (en color verde). En caso contrario (color rojo), deben configurarse correctamente el *path* y/o el directorio temporal de trabajo.

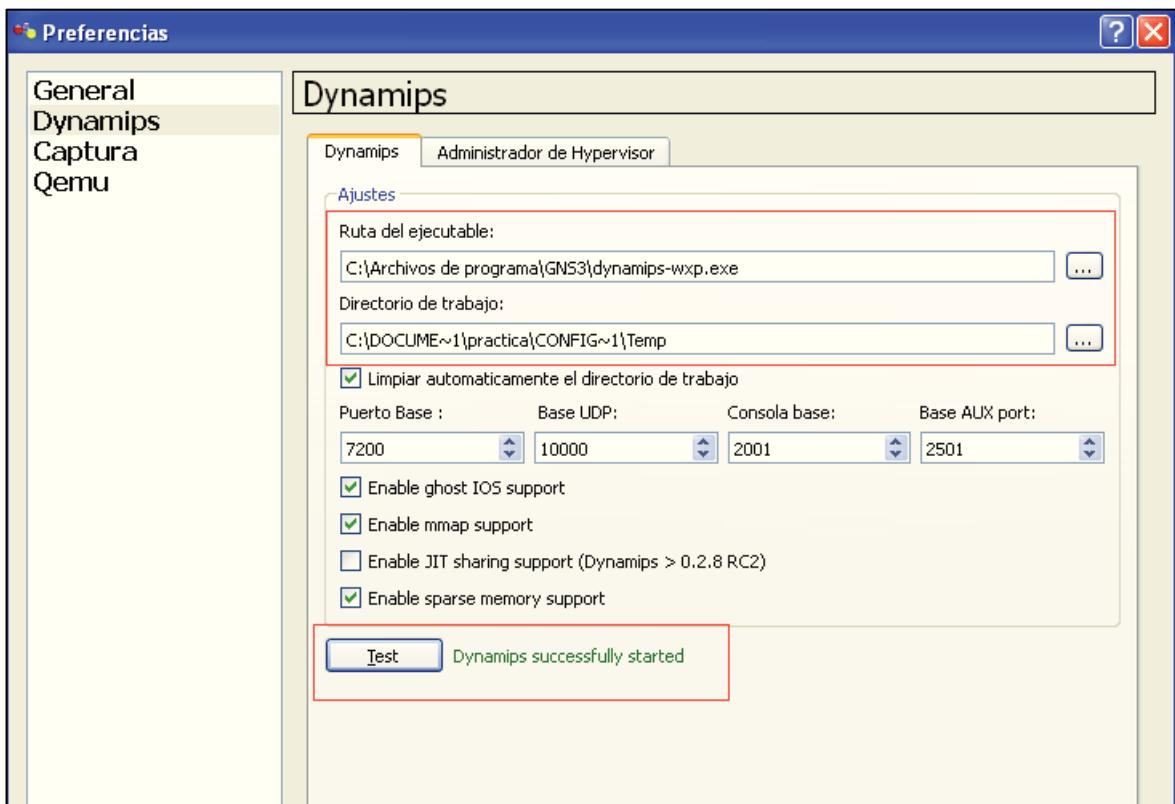


Figura 7.46. Contexto gráfico para la verificación del *path* o ruta en la ejecución del proceso *dynamips.exe* y su directorio de trabajo temporal.

7.3.5. Optimización del rendimiento de CPU: parámetro *idle PC*.

Una tarea de configuración absolutamente prioritaria y fundamental del entorno de emulación es el cálculo del denominado *idle PC*. Este es uno de los parámetros de configuración inicial mostrados en la *figura 7.41*, la primera vez que usamos el software. Pero la configuración de este parámetro es preciso abordarla manualmente llegado el momento de usar un dispositivo emulado concreto.

Sin una correcta configuración del mismo, el consumo de CPU puede alcanzar valores entre el 50 % y el 100 % que hacen prácticamente inviable la ejecución de los procesos asociados a la emulación. Esto es debido a que en la emulación de *Cisco IOS*, el procesador del sistema emulador desconoce cuándo el dispositivo emulado está realmente activo y cuándo está “desocupado”, y por ello está ejecutando continuamente las instrucciones que conforman el entrampado de su sistema operativo *Cisco IOS*. La configuración del parámetro *idle PC* es por lo tanto, a grandes rasgos, una manera de informar a nuestro procesador cuándo los dispositivos emulados se encuentran realmente activos, y así ejecutar las instrucciones precisas de su sistema operativo cuando corresponda, no de manera continua. En caso de no estar activos, los dispositivos emulados permanecen en un estado de “*pseudoreposo*” (*idle*). En principio, el valor *idle PC* debe calcularse para cada *Cisco IOS* concreto [12][13]. Pero la experiencia práctica acumulada a lo largo del desarrollo del proyecto nos recomienda calcularlo para cada Cisco IOS y para cada PC emulador concreto. Es decir, un valor de *idle PC* calculado para un determinado *Cisco IOS* en un PC concreto no sería un valor óptimo para el mismo sistema *Cisco IOS* en distinto PC emulador.

El cálculo del parámetro *idle PC* es relativamente sencillo. Una vez disponemos de un dispositivo emulado en nuestro panel principal de *GNS3* e iniciamos este dispositivo (dispositivo asociado a una determinada imagen *Cisco IOS* según las indicaciones anteriores), el procesador de nuestro PC emulador comenzará a ejecutar, vía proceso *dynamips.exe*, las instrucciones asociadas al sistema a emular. En este momento, haciendo “clic” con el botón derecho del ratón sobre el ícono del dispositivo, del menú desplegable seleccionamos la opción ***idle PC***. Tras unos segundos, se nos mostrará por pantalla una lista desplegable de posibles valores. De estos valores, los realmente válidos, son aquellos que van acompañados por el símbolo (*). Si al calcular el valor *idle PC* no obtuviésemos un valor con (*) deberíamos repetir el proceso. Recurriendo de nuevo a la experiencia acumulada en el análisis de esta plataforma de emulación, un pequeño consejo práctico para obtener un buen valor de *idle PC* es proceder al cálculo cuando estamos ejecutando alguna instrucción *Cisco IOS* desde la consola del dispositivo.

Concretando descriptivamente nuestro pequeño consejo práctico:

iniciamos el dispositivo y esperamos el tiempo suficiente para que se cargue correctamente *Cisco IOS*. En este momento, si no hemos calculado previamente *idle PC* o no disponemos de un valor óptimo del mismo, el valor de carga de CPU puede llegar al 100 %. Por esta circunstancia, nuestro sistema funcionará lentamente, pero sin llegar a bloquearse. Con relativa habilidad, ejecutamos alguna instrucción *Cisco IOS*, vía consola del dispositivo, y

seguida e inmediatamente calculamos *idle PC*. La experiencia práctica nos dice que los valores así calculados son óptimos, o próximos a serlo.

Este pequeño apunte práctico sería útil en caso de querer asignar un valor adecuado de *idle PC* antes de incorporar el dispositivo emulado en la topología de una determinada arquitectura, para así proceder con las configuraciones de red de un modo más “ágil” por parte del sistema emulador.

Otra forma de proceder para el cálculo de *idle PC* podría ser:

una vez diseñada la topología y activa una determinada arquitectura con sus dispositivos en funcionamiento, aunque sea con un valor inicial no óptimo de *idle PC*, calcular en este momento el valor hasta dar con un valor óptimo. Al estar los dispositivos de red activos, esto es, ejecutando instrucciones del sistema operativo asociado, los valores calculados serían óptimos, o próximos a serlo. El inconveniente de esta propuesta es que hasta concretar un valor adecuado el funcionamiento del sistema podría llegar a ser bastante deficiente (con cargas de CPU hasta 100 %), siendo un proceso lento y tedioso.

Para un determinado dispositivo de red, imagen *Cisco IOS* asociada al mismo y sistema emulador, el cálculo del valor óptimo de *idle PC* es un proceso que sólo es necesario hacer una vez .

7.3.6. Optimización del rendimiento de la memoria RAM.

Si la carga de CPU asociado a una plataforma de emulación es crítico, el consumo de memoria RAM no lo es menos. Tal como recogen los resultados del capítulo 4, el consumo de memoria es ciertamente decisivo y crítico en el correcto desempeño de los procesos de emulación. Para abordar esta contingencia, el entorno de emulación *GNS3/Dynamips* implementa e integra diversas mejoras, por defecto activadas, pero que con un buen dominio del entorno, y según la cantidad de recursos hardware disponibles, pueden desactivarse a voluntad para priorizar unas sobre otras. Al respecto cabe mencionar que en el desarrollo de este proyecto se ha hecho uso de la configuración por defecto, esto es, con las mejoras activadas. El acceso a estos recursos para la optimización del consumo de memoria RAM, reflejado en la figura 7.47 de la página siguiente, se realiza a través de: **Edit** → **Preferences** → **Dynamips** . Estos recursos de optimización [12], a grandes rasgos, operan como se indica:

- *Ghost IOS support*: genera una región de memoria compartida de forma que dispositivos que emulen idéntico sistema *Cisco IOS* accedan a esa única región de memoria común ahondando en un evidente ahorro de consumo de memoria RAM. Habilitada esta función se generan archivos *.ghost en un directorio *working* con el mismo *path* que los directorios de trabajo previamente descritos, archivos que corresponden a la región compartida en la memoria.

- *Enable sparse memory support:* no afecta al consumo de memoria RAM “real”. Este recurso, limita únicamente el uso de memoria virtual consumida por las instancias de los *router* considerando únicamente el consumo de memoria de aquellos dispositivos que están realmente activos. Este recurso afecta al consumo de memoria virtual [12].

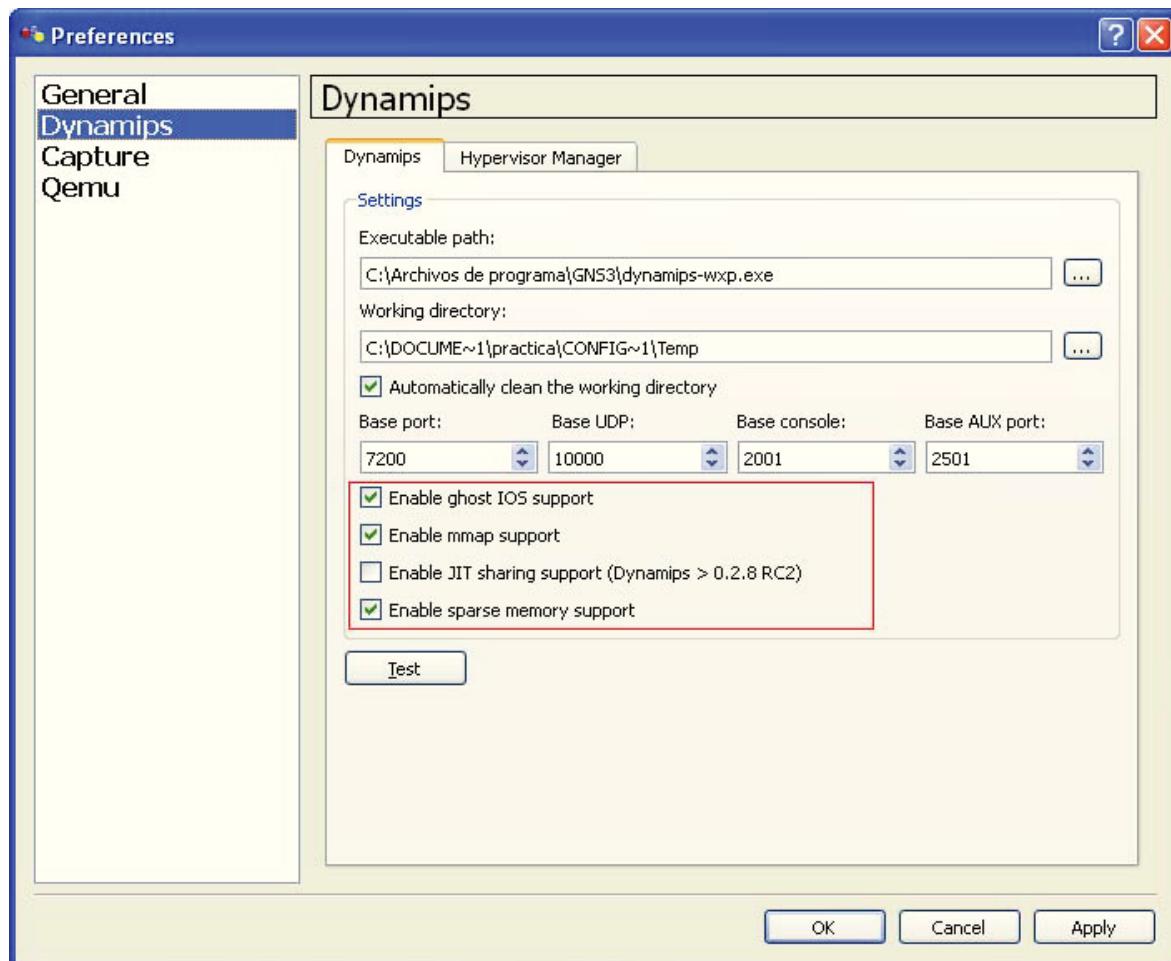


Figura 7.47. Contexto gráfico para la configuración de los recursos optimizadores de la memoria RAM del sistema emulado.

Para la correcta ejecución de ambos recursos RAM optimizadores, la configuración de *GNS3/Dynamips* obliga a activar también la opción *Enable mmap support*; en caso contrario, el funcionamiento de los recursos anteriores no será correcto.

Si bien el entorno emulador *GNS3/Dynamips* implementa otra serie de recursos para la optimización de los procesos asociados a la emulación, abordarlas escapa a los objetivos del proyecto y no ha sido necesario su concurso en el planteamiento de los escenarios de trabajo del mismo. Huelga decir que tales recursos pueden ser objeto de posteriores análisis para profundizar en el estudio de las potencialidades de esta plataforma de emulación, y así se postula en el capítulo 5.

7.4. Anexo 4: Análisis de prestaciones y rendimiento del sistema emulador en entornos de trabajo Linux (Distribución Ubuntu 12.04).

Habiéndose concretado todos los análisis del entorno emulador *GNS3/Dynamips* descritos en la presente memoria bajo el sistema operativo Windows, en concreto Windows XP de 32 bits (se comprueba además que el funcionamiento es visible y similar bajo el sistema operativo Windows 7 de 32 y 64 bits), este anexo se elabora *ex profeso* para mostrar ciertas tareas de configuración de *GNS3/Dynamips* manifiestamente diferentes en sistemas operativos Linux, en este caso la distribución *Ubuntu 12.04*. Además, y sin la profundidad de contenidos con respecto a los análisis efectuados bajo sistema operativo Windows, se analiza la capacidad y rendimiento del sistema emulador para la configuración hardware 1 descrita en el capítulo 4, así como su influencia en la latencia de red en la *interoperación entorno real ↔ entorno emulado*.

7.4.1. Instalación/Configuración de *GNS3/Dynamips* en sistemas Linux.

Las tareas de configuración en sistemas operativos Linux es básicamente la misma que en sistemas Windows, con ciertas particularidades relacionadas con la arquitectura software propia del sistema operativo. En primer lugar, descargaremos los archivos *GNS3-0.7.3-src.tar.gz* y *Dynamips 0.2.8-RC2-x86.bin for Linux (32-bit)* [12]. Vamos a instalar y configurar en el sistema/PC emulador las mismas versiones del software que hemos hecho uso en Windows, aunque existen versiones más actuales.

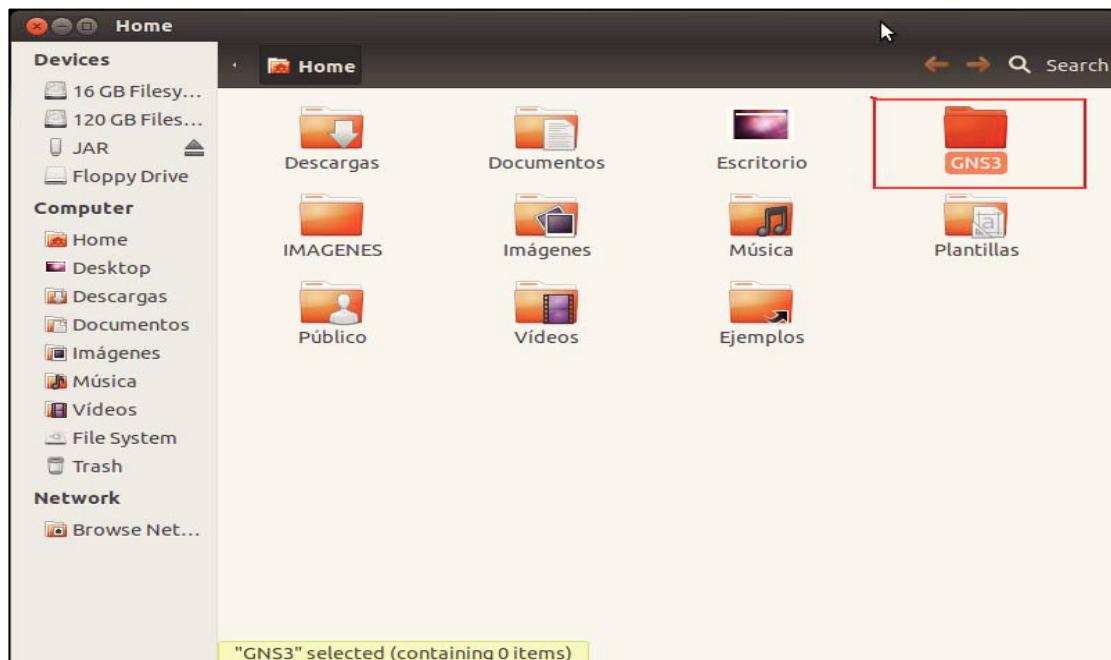


Figura 7.48. Vista detallada del directorio/carpeta creada en /home para la instalación de *GNS3/Dynamips* en Linux.

Después de descomprimir/installar el archivo *GNS3-0.7.3-src.tar.gz* en el directorio creado */home/GNS3*, como muestra la *figura 7.48* anterior, copiamos el archivo binario *Dynamips 0.2.8-RC2-x86.bin* en el mismo directorio donde se ubican los archivos fuente, */home/GNS3/GNS3-0.7.3-src* .

Posteriormente, utilizando la aplicación de Ubuntu *Synaptic Package Manager* buscamos el paquete software/librerías/dependencias *python-qt4*, marcándolo “*para instalación*” en la lista de resultados de la búsqueda, haciendo “*clic*” con el botón derecho del ratón. *Python-Qt* son librerías multiplataforma ampliamente usadas en el desarrollo de aplicaciones con interfaz gráfica de usuario, como es nuestro caso [20]. *Python-Qt* es utilizada, por ejemplo, en KDE, entorno de escritorio para sistemas Linux, entre otros. Tras la correcta instalación, se nos informa de la misma (*figura 7.49*).



Figura 7.49. Vista detallada de la instalación de las librerías *python-qt4*.

Seguidamente, se procede con el último de los pasos necesarios para la correcta instalación y configuración de *GNS3/Dynamips* en sistemas Linux. Así, haciendo “clic” con el botón derecho sobre el archivo *Dynamips 0.2.8-RC2-x86.bin*, marcamos al respecto la opción “*permitir ejecutar como programa*”, asegurándonos además activar los permisos de lectura y escritura (*figura 7.50*). A partir de ahora, todos los procesos de configuración y operación, así como el entorno gráfico para los mismos, son idénticos que en entornos de trabajo Windows.

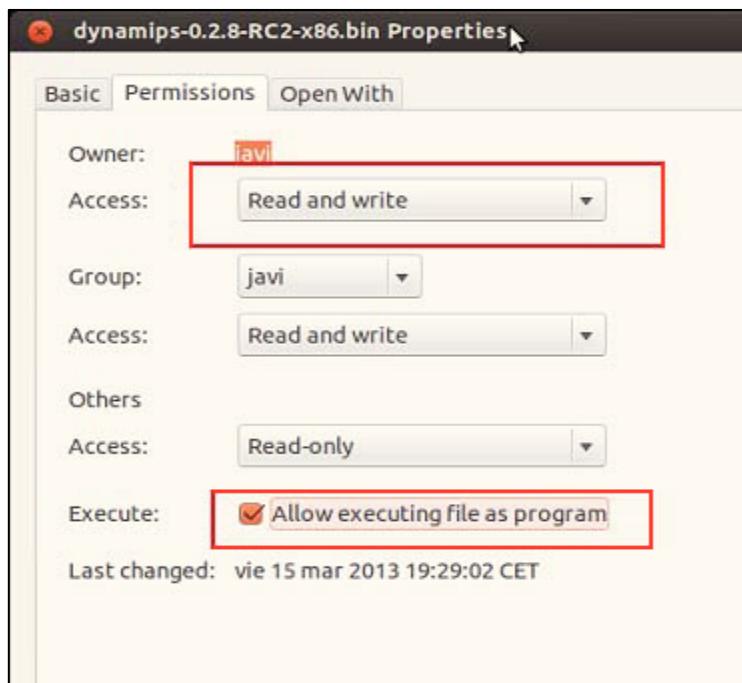


Figura 7.50. Contexto gráfico de propiedades del archivo binario de la emulación.

Creación/configuración de directorios de trabajo y *path* para la emulación

El objetivo de esta configuración es idéntico que en sistemas operativos Windows: disponer de carpetas/directorios donde ubicar nuestros proyectos y las imágenes *Cisco IOS* asociadas a los dispositivos emulados. Únicamente cambia la “estructura” del *path* o ruta de los mismos, asociado a sistemas Linux y su estructura de directorios. Así, configuraremos como directorio para ubicar nuestros proyectos `/home/javi/GNS3/projects` , y como directorio para las imágenes *Cisco IOS* `/home/javi/GNS3/ímágenes` , desde el mismo contexto gráfico que en Windows, como refleja la *figura 7.51* de la página siguiente. Igualmente, debe comprobarse que el *path* para la ejecución del archivo binario *Dynamips 0.2.8-RC2-x86.bin* es correcto, así como el directorio temporal de trabajo, fundamentales y prioritarios para la emulación (*figura 7.52* de la página siguiente). Al respecto, como se indica en el anexo 3 de la memoria, esta verificación sólo es necesario hacerla una vez, y debe configurarse correctamente en caso de que se informe de su inconveniencia (**en rojo**). Tal como se mencionó anteriormente, a partir de aquí todas las demás tareas de configuración (excepción hecha con la configuración del interfaz *Ethernet*

para implementar la *interoperación entorno real \leftrightarrow entorno emulado*, que merece una detallada descripción a continuación), operación y trabajo se implementan exactamente igual y con idéntico interfaz gráfico que bajo sistemas operativos Windows.

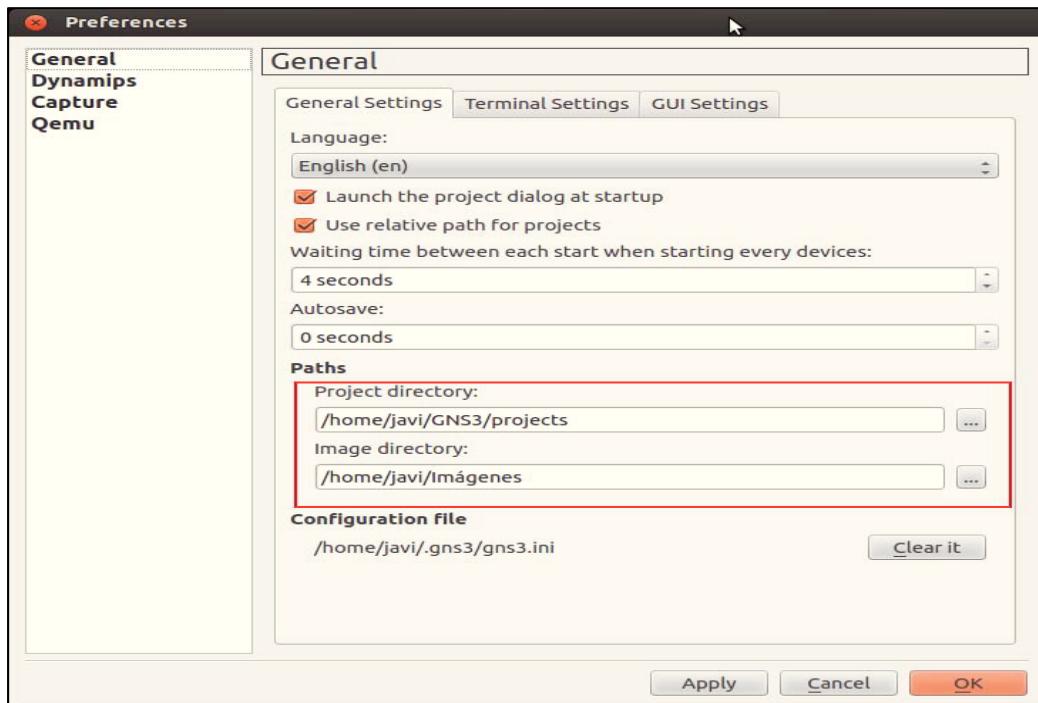


Figura 7.51. Configuración de los directorios de trabajo para *GNS3/Dynamips* en Linux.

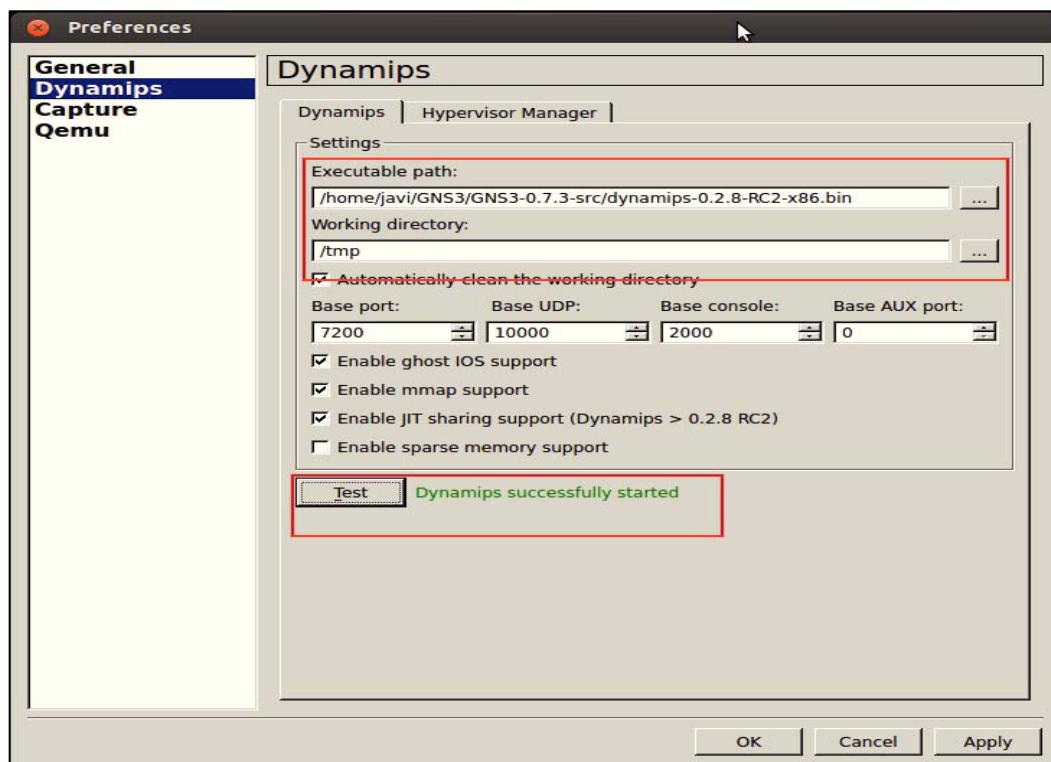


Figura 7.52. Contexto gráfico para la verificación/configuración del *path* para la emulación.

Configuración/selección en Linux del interfaz *Ethernet* para la *interoperación entorno real ↔ entorno emulado*.

Con respecto a sistemas operativos Windows, la configuración del interfaz *Ethernet* necesario para implementar la *interoperación entorno real ↔ entorno emulado* es ligeramente distinta. Aunque el contexto gráfico para realizar la misma es igual que en Windows, como muestra la figura 7.53, se da la circunstancia de que para poder implementarla correctamente es necesario ejecutar *GNS3/Dynamips* como usuario *root*, es decir, con todos los privilegios administrativos del sistema. En caso contrario, y aunque hayamos podido configurar el interfaz sin ningún inconveniente tal como hacíamos en entornos de trabajo Windows, el sistema mostrará un error (en concreto, el error nº 206) al intentar interconectar en la topología de diseño en *GNS3*, y vía enlace *Ethernet*, el dispositivo *cloud* o nube (que recordemos sirve para representar equipos reales que *interoperan* con el entorno emulado) con un dispositivo de red emulado.

Siendo así, para ejecutar *GNS3/Dynamips* como usuario *root*, abrimos el interfaz del intérprete de comandos o Terminal de Linux, y ejecutamos: `sudo gns3`, si estamos ubicados en el directorio donde están todos nuestros archivos fuente. A continuación, seleccionamos el interfaz *Ethernet* deseado de la lista de interfaces disponibles en el PC emulador tal como hacíamos en Windows, proceso descrito en el anexo 3 de la memoria. De un modo gráfico, todo el proceso descrito se muestra en la *subfiguras 7.54 a) y b)*.

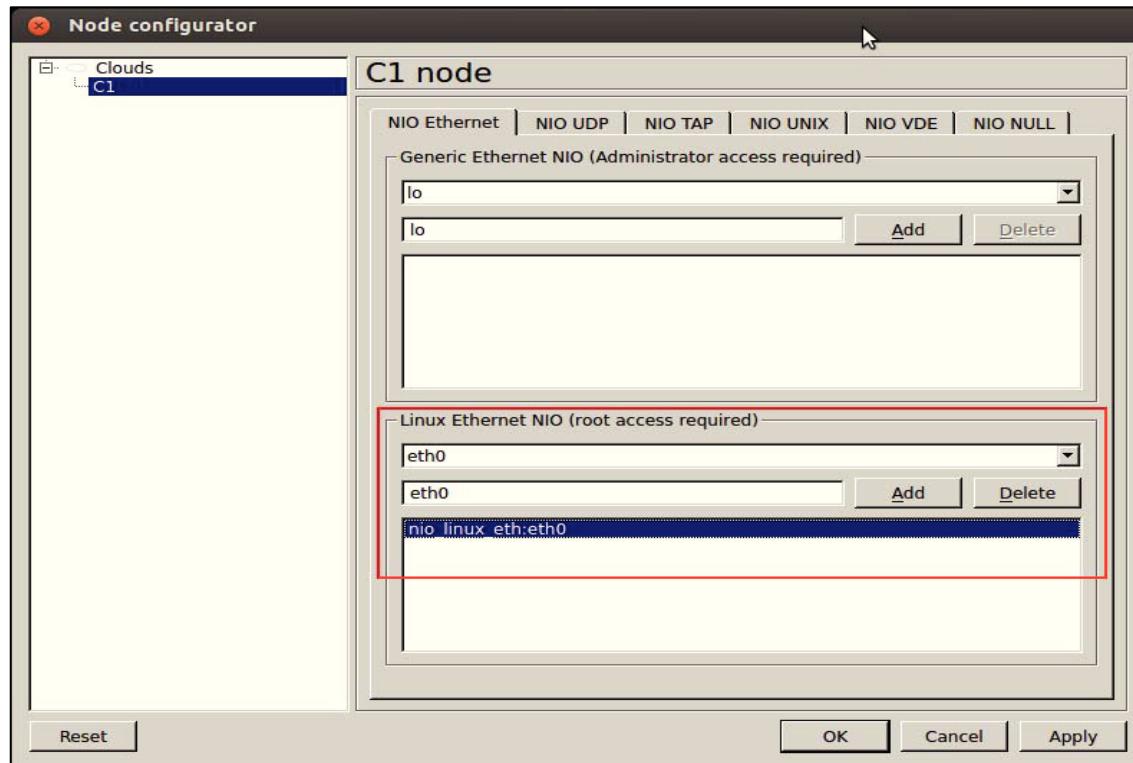


Figura 7.53. Contexto gráfico para la la selección /configuración del interfaz *Ethernet* (como usuario *root*) en la *interoperación entorno real ↔ entorno emulado* en sistemas Linux.

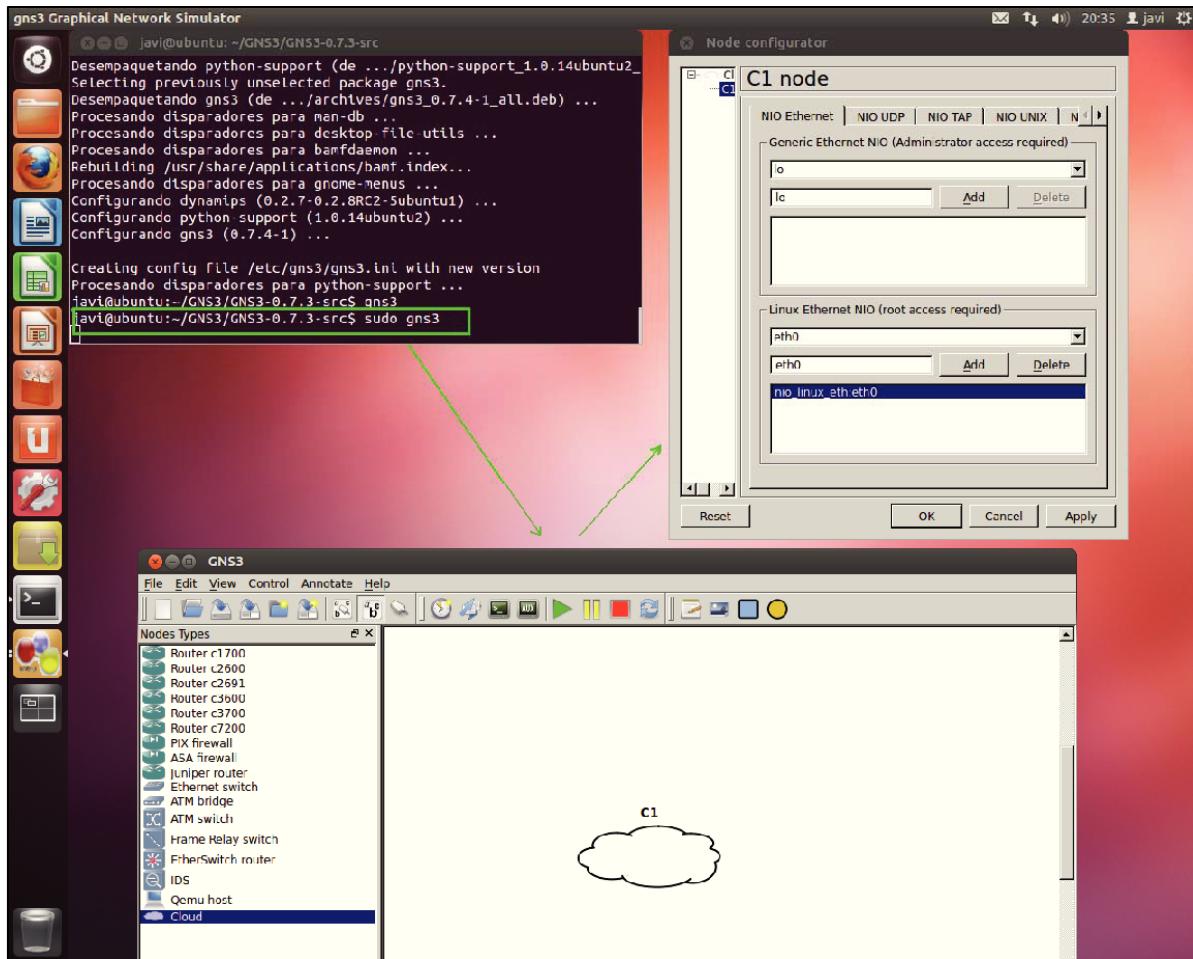


Figura 7.54 a) Proceso para la configuración/selección del interfaz *Ethernet* (como usuario *root*) en la *interoperación entorno real ↔ entorno emulado* en sistemas Linux.

```
gns3 Graphical Network Simulator
javi@ubuntu: ~/GNS3/GNS3-0.7.3-src
Desempaquetando python-support (de .../python-support_1.0.14ubuntu2_
Selecting previously unselected package gns3.
Desempaquetando gns3 (de .../archives/gns3_0.7.4-1_all.deb) ...
Procesando disparadores para man-db ...
Procesando disparadores para desktop-file-utils ...
Procesando disparadores para bamfdaemon ...
Rebuilding /usr/share/applications/bamf.index...
Procesando disparadores para gnome-menus ...
Configurando dynamips (0.2.7-0.2.8RC2-5ubuntu1) ...
Configurando python-support (1.0.14ubuntu2) ...
Configurando gns3 (0.7.4-1) ...

Creating config file /etc/gns3/gns3.ini with new version
Procesando disparadores para python-support ...
javi@ubuntu:~/GNS3/GNS3-0.7.3-src$ gns3
javi@ubuntu:~/GNS3/GNS3-0.7.3-src$ sudo gns3
```

Figura 7.54 b) Vista detallada del intérprete de comandos Linux de la figura 7.49 a).

7.4.2. Análisis de la capacidad y rendimiento del sistema emulador en entornos de trabajo Linux.

Del mismo modo que previamente se describió en los *apartados 7.2.6 y 7.2.7* del anexo 2, procedemos con la configuración de la arquitectura LAN Ethernet/VTP en sistemas Linux. Siendo así, las configuraciones son exactamente implementadas con respecto al entorno de trabajo Windows, configuraciones descritas en el anexo 2. Como el propósito de este apartado es establecer un análisis comparativo de la capacidad y rendimiento del sistema/PC emulador en Linux tomando como referencia el mismo escenario telemático que en Windows, y para no sobrecargar la memoria con resultados redundantes, únicamente se procede a la implementación y configuración de la arquitectura LAN Ethernet/VTP. Así, la *figura 7.55* nos muestra la arquitectura de red configurada para proceder con el análisis en Ubuntu 12.04 (idéntica arquitectura y topología que en *GNS3/Dynamips* bajo S.O. Windows)

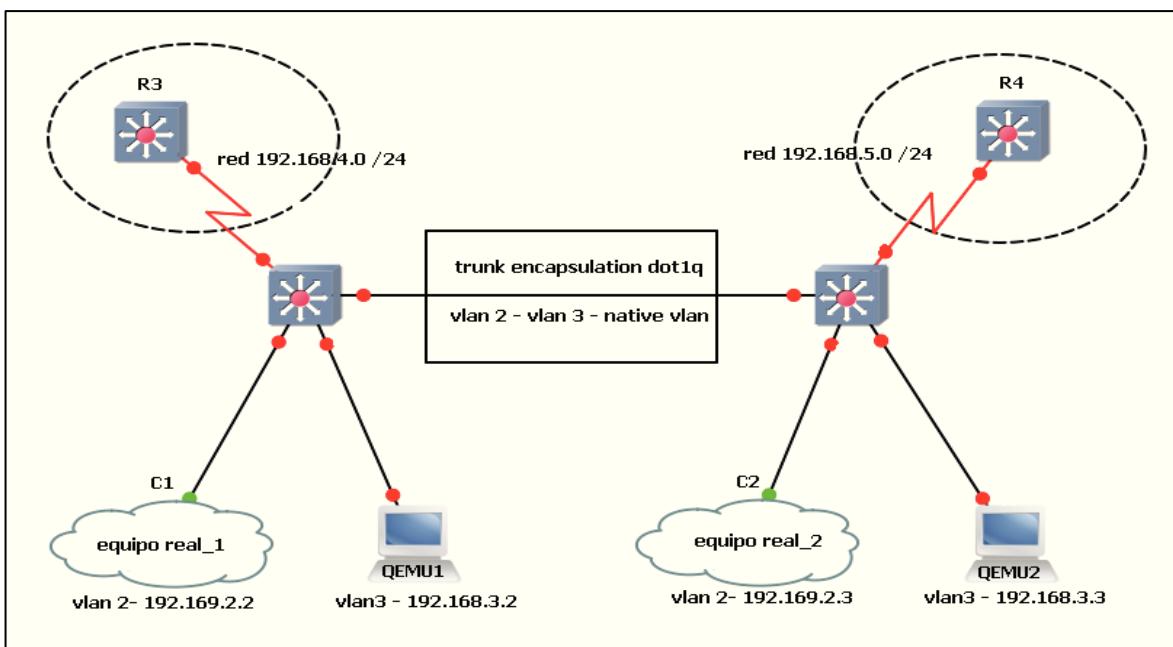


Figura 7.55. Arquitectura de red LAN Ethernet /VTP para analizar la capacidad y rendimiento de *GNS3/Dynamips* en Linux (idéntica que en Windows)

Conforme a los planteamientos anteriores, todas las connotaciones de configuración y diseño descritas en los *apartados 7.2.6 y 7.2.7* del anexo 2 son aplicables al contexto de trabajo actual. Así mismo, los análisis de capacidad y rendimiento han sido efectuados para la *configuración hardware 1*, descrita en la *tabla 4.1* del capítulo 4, con un valor óptimo de *idle PC* para los dispositivos de red emulados y en situaciones de consumo máximo y mínimo de recursos. Los resultados correspondientes al consumo de recursos se muestran a continuación, en la *tabla 7.3* de la página siguiente. En estos resultados, habiéndose establecido las consideraciones oportunas en el capítulo 5 y anexo 2, no se considera el uso de *host* emulados Linux QEMU (en líneas generales se constata que su uso eleva considerablemente el consumo de recursos en el PC emulador).

Configuración hardware 1	Carga media de CPU (<i>offmode</i>)	MemoriaRAM en uso (<i>offmode</i>)	Carga media de CPU (<i>pseudoactivo</i>)	Memoria RAM en uso en MB (<i>pseudoactivo</i>)
1 <i>switch-router</i> activo	0-1%	24 %	1 %	716 → 490
2 <i>switch-router</i> activos	0-1%	24 %	3% → 2 %	820 → 610
2 <i>switch-router</i> activos + 1 nodo auxiliar	0-1%	24 %	5 % → 3 %	830 → 660
2 <i>switch-router</i> activos + 2 nodos auxiliar	0-1%.	24 %	10 % → 8 %	870 → 720

Tabla 7.3. Consumo de recursos del sistema emulador para la *configuración hardware 1*, bajo S.O. Linux 32 bits y arquitectura de red LAN/VTP.

Configuración hardware 1	Carga media de CPU (<i>offmode</i>)	MemoriaRAM en uso en MB (<i>offmode</i>)	Carga media de CPU (<i>pseudoactivo</i>)	Memoria RAM en uso en MB (<i>pseudoactivo</i>)
1 <i>switch-router</i> activo	1% max.	318	2 %	840 → 570
2 <i>switch-router</i> activos	1% max.	318	5% → 4 %	960 → 740
2 <i>switch-router</i> activos + 1 nodo auxiliar	1% max.	318	7 % → 6 %	980 → 770
2 <i>switch-router</i> activos + 2 nodos auxiliar	1% max.	318	13 % → 11 %	990 → 780

Tabla 4.9. Consumo de recursos del sistema emulador para la *configuración hardware 1*, bajo S.O. Windows 32 bits y arquitectura de red LAN/VTP.

Tal como se describió en el capítulo 4, las expresiones **XX → XX** deben interpretarse como un descenso en el consumo de recursos en el PC emulador.

Los resultados comparativos anteriores muestran un descenso en el consumo de recursos en el sistema emulador con respecto a los resultados mostrados en la *tabla 4.9*, tabla que muestra el consumo de recursos bajo sistemas operativos Windows. Por claridad

y comodidad en la exposición y el análisis comparativo, se reproduce de nuevo la *tabla 4.10* del capítulo 4. Conforme a esta comparación, se constata que la eficiencia de *GNS3/Dynamips* en Linux es ligeramente mejor que en Windows, y la potencial escalabilidad de la arquitectura en cuanto al número de nodos emulados (abstrayéndonos de otras consideraciones de diseño) podría ser ligeramente mayor con respecto a la misma en Windows, pudiendo aumentar la cota superior “emulable” en un nodo más, llegado el caso, y para nuestro particular caso de estudio.

7.4.3. Análisis de la latencia en la *interoperación entorno real ↔ entorno emulado* en sistemas operativos Linux.

Relacionado con el consumo de recursos anterior, se realiza un breve análisis comparativo de la latencia de red en la *interoperación entorno real ↔ entorno emulado* en sistemas Linux respecto a sistemas Windows. Tal como se estableció en el apartado anterior, las connotaciones de diseño y configuración son las mismas que para entornos de trabajo Windows del capítulo 4. Siendo así, se procede al análisis comparativo para los dos experimentos de la latencia descritos en el capítulo 4, que recordemos son :

1. Latencia para diferentes longitudes de trama *Ethernet*, en transmisiones sin control de errores (protocolo ICMP) para la conexión: *equipo real ↔ interfaz SVI de nodo virtual emulado 1*.
2. Latencia para diferentes longitudes de trama *Ethernet*, en transmisiones sin control de errores (protocolo ICMP), para la conexión *extremo a extremo*: *equipo real 1 → entorno emulado (vía SVI de nodo virtual emulado1) ↔ equipo real 2*.

RTT promedio (configuración hardware 1) En milisegundos	1 switch-router activo	2 switch-router activos	3 switch-router activos	4 switch-router activos
Longitud Trama 64 bytes	30 → 28	36 → 30	42 → 35	54 → 43
Longitud Trama 512 bytes	31 → 28	37 → 30	42 → 35	55 → 44
Longitud Trama 1024 bytes	35 → 29	39 → 32	46 → 39	59 → 47
Longitud Trama 1514 bytes	37 → 30	40 → 33	48 → 41	60 → 48

Tabla 7.4. Latencia de red para el protocolo ICMP en la conexión *equipo_real 1 → interfaz SVI de nodo virtual 1*, para diferentes longitudes de trama y consumo de recursos del PC emulador 1, arquitectura de red LAN/VTP y sistemas operativos Linux.

RTT promedio <i>(configuración hardware 1)</i> En milisegundos	1 switch-router activo	2 switch-router activos	3 switch-router activos	4 switch-router activos
Longitud Trama 64 bytes	32 → 30	40 → 34	47 → 40	61 → 49
Longitud Trama 512 bytes	34 → 31	41 → 35	48 → 42	62 → 51
Longitud Trama 1024 bytes	37 → 32	44 → 37	51 → 45	67 → 54
Longitud Trama 1514 bytes	39 → 33	45 → 39	52 → 46	68 → 55

Tabla 4.11. Latencia de red para el protocolo ICMP en la conexión *equipo_real 1* → *interfaz SVI de nodo virtual 1*, para diferentes longitudes de trama y consumo de recursos del PC emulador 1, arquitectura de red LAN/VTP y sistemas operativos Windows.

Al respecto, el análisis comparativo de las *tablas 7.4* y *4.11* anteriores relacionadas con el primero de los análisis de la latencia constata un ligero descenso en los valores de la misma para la *interoperación entorno real* ↔ *entorno emulado*, descenso directamente relacionado con el del consumo de recursos en sistemas operativos Linux mostrado en el apartado anterior. Del mismo modo, relacionado con el segundo de los análisis de la latencia, se constata en las *tablas 7.5* y *4.13*, aunque en menor medida, este descenso en el valor de la latencia de red para entornos de trabajo Linux con respecto a sistemas operativos Windows.

RTT promedio <i>(configuración hardware 1)</i> En milisegundos	1 switch-router activo	2 switch-router activos	3 switch-router activos	4 switch-router activos
Longitud Trama 802.1q 68 bytes	-	2 → 1	2 → 1	4 → 3
Longitud Trama 802.1q 516 bytes	-	2 → 1	3 → 2	5 → 4
Longitud Trama 802.1q 1028 bytes	-	2 → 1	3 → 2	6 → 5
Longitud Trama 802.1q 1518 bytes	-	3 → 2	4 → 3	7 → 5

Tabla 7.5. Latencia de red para el protocolo ICMP en la conexión *equipo_real 1* → *entorno emulado* → *equipo real 2*, para diferentes longitudes de trama Ethernet, consumo de recursos del PC emulador 1, arquitectura de red LAN/VTP y sistemas operativos Linux.

De hecho, la mejora en los valores de la latencia en sistemas operativos Linux para este segundo experimento sólo es contable a partir de 3 nodos emulados activos (lógico, debido al menor valor de la latencia en este ensayo).

RTT promedio (configuración hardware 1) En milisegundos	1 switch-router activo	2 switch-router activos	3 switch-router activos	4 switch-router activos
Longitud Trama 802.1q 68 bytes	-	2 → 1-2	4 → 3	5 → 4
Longitud Trama 802.1q 516 bytes	-	2 → 1-2	4 → 3	6 → 5
Longitud Trama 802.1q 1028 bytes	-	2 → 1-2	4 → 3	7 → 6
Longitud Trama 802.1q 1518 bytes	-	3 → 2-3	5 → 4	8 → 7

Tabla 4.13. Latencia de red para el protocolo ICMP en la conexión *equipo_real 1* → *entorno emulado* → *equipo real 2*, para diferentes longitudes de trama Ethernet, consumo de recursos del PC emulador 1, arquitectura de red LAN/VTP y sistemas operativos Windows.

Los valores recogidos con formato **XX → XX**, en este caso, hacen referencia a valores de latencia en instantes de tiempo de consumo máximo y mínimo respectivamente de recursos en el PC emulador (carga de CPU y consumo de memoria RAM), tal como se describió en el capítulo 4.

Los análisis anteriores sirven para confirmar la mejor eficiencia (ligera mejor eficiencia) de *GNS3/Dynamips* en entornos de trabajo Linux respecto a sistemas Windows, tanto en el consumo de recursos del sistema emulador, como en el comportamiento de la latencia (evolución en tiempo real del tráfico de la arquitectura de red). Por claridad en la exposición de los análisis comparativos anteriores se han mostrado de nuevo las *tablas 4.11 y 4.13* del capítulo 4.

7.5. Anexo 5: Implementación de un servicio telemático cliente /servidor FTP vía interoperación entorno real (Ethernet 802.3) ↔ entorno emulado en arquitecturas de red emuladas.

El experimento siguiente aborda intensamente el objetivo considerado en el proyecto de analizar la potencial *interoperación entorno real ↔ entorno emulado* en *GNS3 /Dynamips* y complementa los análisis de los capítulos 3 y 4 al respecto. Secundariamente, se evalúan los efectos directos sobre el tiempo requerido para la prestación de un servicio *cliente/servidor* (transferencia FTP de un archivo), efectos asociados de manera indirecta al consumo de recursos *hardware* en el PC emulador, y por consiguiente, relacionados con la latencia del tráfico de red emulado. Pero en este caso, sólo vamos a constatar la viabilidad de implementar este escenario de trabajo *real ↔ emulado*, y la correcta transmisión/recepción FTP de un fichero, con objetivos principalmente académicos. Un análisis en profundidad de la latencia en este escenario sería ciertamente complejo debido a que hay que considerar la latencia asociada al protocolo TCP y la interacción con la latencia introducida por los procesos de la emulación y el interfaz *Ethernet* empleado en la *interoperación*, latencia previamente constatada.

7.5.1. Implementación de un servicio cliente/servidor FTP vía interoperación entorno real (Ethernet 802.3) ↔ entorno emulado en el análisis de la arquitectura de red WAN ppp.

Siendo la arquitectura de red emulada WAN ppp, el escenario de trabajo implementado es el reflejado en la *figura 7.56*. (los nodos emulados Auxiliar 1 y 2 no son necesarios).

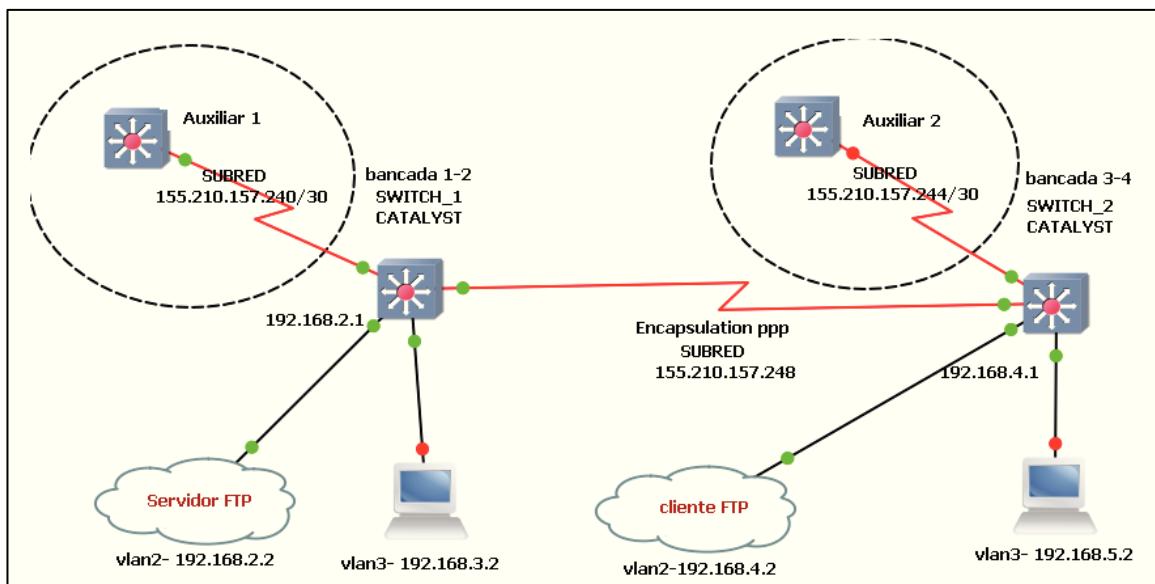


Figura 7.56. Topología en *GNS3* de la arquitectura WAN ppp para implementar el servicio cliente/servidor FTP con los roles de servidor y cliente FTP particularizados.

Se analiza así la viabilidad de la conexión y la transferencia FTP pasiva (solicitada por el cliente) y anónima [15] de un archivo desde el *servidor FTP (equipo real 1)* → *cliente FTP (equipo real 2)*. Tanto el servidor como el cliente FTP se configuran sobre equipos reales externos al emulador, adscritos a una misma VLAN (VLAN2), pero en nodos de red emulados diferentes. Esto es, nuestro archivo (*archivo.rar* de 250 KB) localizado en un equipo/servidor externo, se transfiere por el entramado virtual/emulado hasta llegar al otro equipo/cliente FTP, externo al emulador (conexión *extremo a extremo* entre equipos reales a través de una arquitectura de red emulada). FTP utiliza TCP como protocolo de transporte, aportando así control de errores en la transmisión. En nuestro caso, implementaremos un servicio cliente/servidor en *modo pasivo*, es decir, el cliente será el que inicie las conexiones de control y de datos [15]. Para configurar los equipos servidor y el cliente FTP se ha hecho uso del software *Filezilla* [16], en sus versiones cliente y servidor.

Verificada la correcta conexión FTP y la consiguiente transmisión/recepción FTP del archivo, así como la integridad del mismo (salvo para el valor de longitud de trama de 64 bytes, valor para el que no es posible establecer correctamente la conexión FTP *cliente* → *servidor*), la tabla 7.6 de la página siguiente muestra los resultados generales obtenidos en el experimento, empleando para ello la herramienta software *Wireshark*. Para la configuración del MTU/longitud de trama hacemos uso de la herramienta software *freeware TCP Optimizer 3.0.8* de la figura 7.57 [17]. Los resultados generales obtenidos estarán condicionados por la longitud de trama y el consumo de recursos del sistema emulador. Confirmada en los análisis de los capítulos 3 y 4 la dependencia directa de la latencia de red en la *interoperación entorno real* ↔ *entorno emulado* con el consumo de recursos del sistema, el análisis que nos ocupa se realiza sólo con dos nodos de red activos, los implicados indefectiblemente en la prestación del servicio FTP.

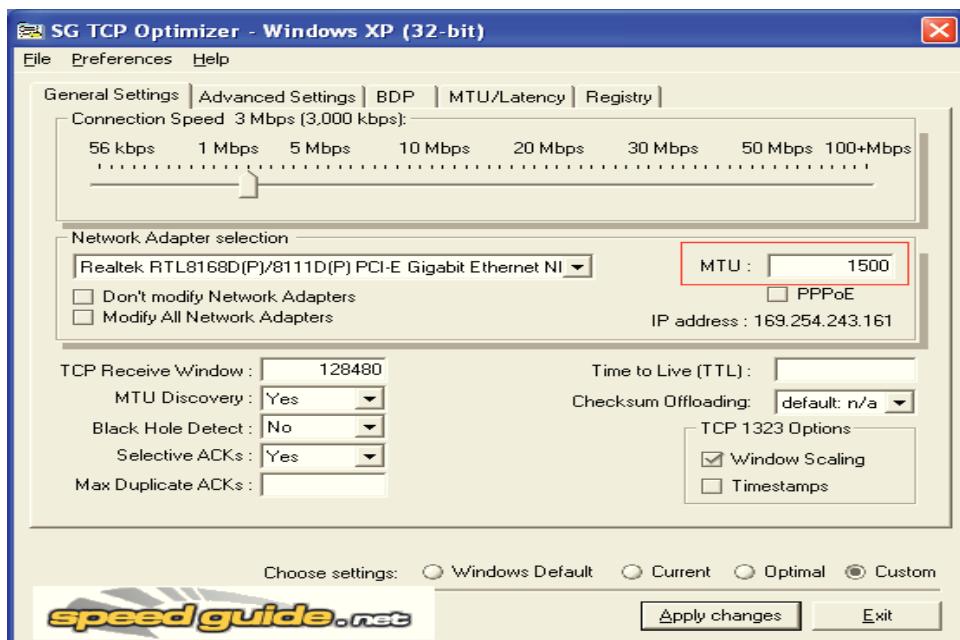


Figura 7.57. Interfaz gráfica de *TCP Optimizer 3.0.8* para la configuración del MTU.

Resultados generales de la transferencia FTP	<i>Longitud de trama Ethernet 64 bytes</i>	<i>Longitud de trama Ethernet 512 bytes</i>	<i>Longitud de trama Ethernet 1024 bytes</i>	<i>Longitud de trama Ethernet 1514 bytes</i>
Longitud trama ACK	-	60 bytes	60 bytes	60 bytes
Nº tramas totales en la transmisión (información y control)	-	986	477	361
Nº tramas de información FTP_DATA	-	560 + 1 trama de 419 bytes	257 + 1 trama de 139 bytes	171 + 1 trama de 1129 bytes
Nº tramas de información FTP_DATA retransmitidas.	-	9	10	9
Δ tiempo total en la transmisión (de información)	-	13,703 seg.	9,114 seg.	11,36 seg.

Tabla 7.6. Resultados obtenidos en la transferencia FTP *servidor* → *cliente* de archivo.rar (250KB) para distintas longitudes de trama Ethernet 802.3 en el análisis de la arquitectura de red WAN ppp.

A grandes rasgos, de los resultados de la *tabla 7.6* podemos concluir que sobre este escenario de trabajo un tamaño de trama *Ethernet* de 1514 bytes no es óptimo, siendo 1024 bytes un mejor valor (para los valores de trama del experimento). El tamaño de trama *Ethernet* de 1514 bytes parece conducir a una situación de congestión (en el enlace serie *ppp*, de capacidad 1,544 Mb) que eleva considerablemente el tiempo necesario para la correcta transmisión FTP del archivo.

7.5.2. Implementación de un servicio cliente /servidor FTP vía interoperación entorno real (*Ethernet 802.3*) ↔ entorno emulado en el análisis de la arquitectura de red LAN Ethernet/VTP.

En el contexto de la arquitectura de red LAN Ethernet/VTP, se plantea un escenario de trabajo similar al descrito anteriormente, reflejado en la *figura 7.58*. Se analiza la viabilidad de la conexión y la transferencia FTP pasiva y anónima de un archivo desde *servidor FTP* (*equipo real 1*) → *cliente FTP* (*equipo real 2*). Tanto el servidor como el cliente FTP se configuran sobre equipos reales externos al emulador, adscritos a una misma VLAN (VLAN2), pero en nodos de red emulados diferentes. En este caso, la transferencia se produce vía arquitectura de red emulada comutada, siendo así esperable un menor tiempo para la correcta transferencia FTP del archivo. Los resultados se muestran en la *tabla 7.7*, y así lo constatan.

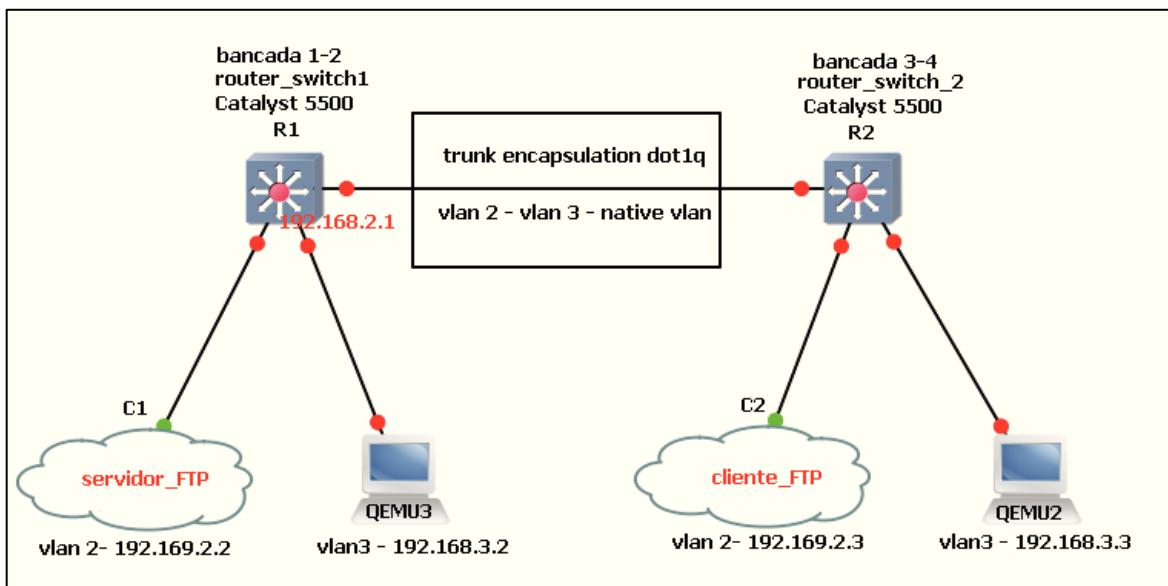


Figura 7.58. Topología en GNS3 de la arquitectura LAN/VTP para analizar el comportamiento del escenario de trabajo en servicios cliente/servidor FTP, con los roles particularizados de servidor y cliente FTP.

Resultados generales de la transferencia FTP	<i>Longitud de trama Ethernet 802.1q 68 bytes</i>	<i>Longitud de trama Ethernet 802.1q 516 bytes</i>	<i>Longitud de trama Ethernet 802.1q 1028 bytes</i>	<i>Longitud de trama Ethernet 802.1q 1518 bytes</i>
Longitud trama ACK	-	66 bytes	66 bytes	66 bytes
Nº tramas totales en la transmisión (información y control)	-	966	475	357
Nº tramas de información FTP_DATA	-	561 + 1 trama final de 305 bytes	259 + 1 trama final de 919 bytes	172 + 1 trama final de 453 bytes
Nº tramas de información FTP_DATA retransmitidas.	-	24	21	7
Δ tiempo total en la transmisión (información)	-	1,3 seg.	1,02 seg.	0,43 seg.

Tabla 7.7. Resultados obtenidos en la transferencia FTP servidor → cliente para distintas longitudes de trama Ethernet 802.1q del enlace troncal en la arquitectura LAN /VTP.

Los resultados de la *tabla 7.7* anterior muestran cómo ha descendido considerablemente el tiempo necesario para la transmisión FTP del archivo (transmisión vía commutación a nivel de enlace o nivel 2) frente al tiempo requerido en la arquitectura anterior (transmisión FTP a nivel de red o nivel 3, vía enlace serie ppp de 1,544 MB). En la arquitectura LAN Ethernet /VTP actual podemos aumentar el valor de la longitud de trama *Ethernet 802.3*, siendo el valor óptimo 1514 bytes (1518 bytes para *Ethernet 802.1q*).

El escenario planteado en este anexo es el punto álgido en el análisis actual de la *interoperación entorno real ↔ entorno emulado* en *GNS3/Dynamips*, completando los experimentos de los capítulos 3 y 4. Aquí se implementa un servicio *cliente/servidor* para la transferencia FTP en modo *anónimo* y *pasivo* de un archivo de 250 KB. Esta transferencia se realiza desde el servidor (*equipo real 1*) al cliente (*equipo real 2*), a través de las arquitecturas de red emuladas. Conclusión a este análisis es que la conexión FTP se estableció correctamente (salvo para el experimento con longitud de trama *Ethernet 802.3* de 64 bytes) y la transmisión del archivo fue igualmente correcta, conservándose la integridad del mismo. La funcionalidad de este escenario en el que *interoperan entorno real ↔ entorno emulado* es acorde a lo esperable teóricamente. Siendo así, la figura 7.59 muestra una captura de tráfico de red mediante *Wireshark* en el segmento *nodo emulado 2 ↔ equipo real 2 (cliente FTP)* de la conexión extremo a extremo: *equipo real 1 (servidor FTP) ↔ entorno emulado ↔ equipo real 2 (cliente FTP)*, para un tamaño de trama de 1514 bytes (correspondiente a un valor de MTU de 1500 bytes) en la arquitectura LAN Ethernet/VTP. Dicha captura confirma la presencia de tramas de información correspondientes al protocolo FTP en uso (tramas FTP_DATA) y con una longitud de trama *Ethernet 802.3* funcionalmente acorde al diseño planteado, además de tramas relacionadas con el protocolo TCP para el control de errores en la transmisión (ACK's cliente → servidor). Sin embargo, en cuanto al tiempo empleado en la transmisión para cada arquitectura los resultados son distintos, según el resumen de la página siguiente.

No.	Time	Source	Destination	Protocol	Length	Info
66	28.497018	192.168.2.2	192.168.2.3	FTP-DATA	1514	FTP Data: 1448 bytes
67	28.497511	192.168.2.2	192.168.2.3	FTP-DATA	1514	FTP Data: 1448 bytes
68	28.497518	192.168.2.2	192.168.2.3	FTP-DATA	1514	FTP Data: 1448 bytes
69	28.499861	192.168.2.3	192.168.2.2	TCP	66	aplx > td-postman [ACK]
70	28.500268	192.168.2.2	192.168.2.3	FTP-DATA	1514	FTP Data: 1448 bytes
71	28.500551	192.168.2.2	192.168.2.3	FTP-DATA	1514	FTP Data: 1448 bytes
72	28.500557	192.168.2.2	192.168.2.3	FTP-DATA	1514	FTP Data: 1448 bytes
73	28.500631	192.168.2.2	192.168.2.3	FTP-DATA	1514	FTP Data: 1448 bytes
74	28.501121	192.168.2.3	192.168.2.2	TCP	66	aplx > td-postman [ACK]
75	28.502567	192.168.2.2	192.168.2.3	FTP-DATA	1514	FTP Data: 1448 bytes
76	28.502572	192.168.2.2	192.168.2.3	FTP-DATA	1514	FTP Data: 1448 bytes
77	28.502577	192.168.2.2	192.168.2.3	FTP-DATA	1514	FTP Data: 1448 bytes
78	28.502581	192.168.2.2	192.168.2.3	FTP-DATA	1514	FTP Data: 1448 bytes

Figura 7.59. Trafico de red capturado en servicios FTP *cliente/servidor* (conexión *equipo real 1 → entorno emulado → equipo real 2*, de la arquitectura LAN/VTP)

Diferencias en la prestación del servicio FTP entre arquitecturas emuladas.

- Arquitectura WAN ppp → Transmisión del fichero a nivel de red (nivel 3) a través del enlace *serie ppp*, considerablemente más lenta que en el caso de la arquitectura LAN Ethernet /VTP conmutada (capacidad del enlace 1,544 MB)
- Arquitectura WAN ppp: la influencia de la latencia de red elevada en la *interoperación entorno real ↔ entorno emulado*, y en la conexión extremo a extremo, previamente constatada, influye decisiva e incondicionalmente de manera negativa.
- Arquitectura LAN Ethernet/VTP → Transmisión FTP vía conmutación a nivel de enlace o nivel 2, a través del enlace troncal configurado (ver anexo 2).
- Arquitectura LAN/VTP: la influencia de la latencia de red en la *interoperación entorno real ↔ entorno emulado* es considerablemente menor, según los resultados del capítulo 4.